QGIS Desktop 3.44 User Guide

QGIS Project

15 mar. 2026

Sumário

1 Iniciando **3**
- 1.1 Instalando o QGIS . . . 3
  - 1.1.1 Instalando a partir de binários . . . 3
  - 1.1.2 Instalando a partir da source . . . 3
  - 1.1.3 Instalando em mídia externa . . . 4
  - 1.1.4 Baixando dados de amostra . . . 4
- 1.2 Iniciando e parando o QGIS . . . 4
- 1.3 Sessão de amostra: Carregando camadas raster e vetorial . . . 5

2 Trabalhando com Arquivos de Projetos **13**
- 2.1 Introduzindo projetos do QGIS . . . 13
- 2.2 Manipulação de caminhos de arquivos quebrados . . . 15
- 2.3 Gerando saída . . . 16

3 Interface do QGIS **17**
- 3.1 Barra de Menu . . . 18
  - 3.1.1 Projeto . . . 18
  - 3.1.2 Editar . . . 19
  - 3.1.3 Ver . . . 22
  - 3.1.4 Camada . . . 25
  - 3.1.5 Configurações . . . 28
  - 3.1.6 Complementos . . . 29
  - 3.1.7 Vetor . . . 29
  - 3.1.8 Raster . . . 31
  - 3.1.9 Base de dados . . . 33
  - 3.1.10 Web . . . 33
  - 3.1.11 Malha . . . 33
  - 3.1.12 Processamento . . . 34
  - 3.1.13 Ajuda . . . 34
  - 3.1.14 QGIS . . . 34
- 3.2 Painéis e Barras de Ferramentas . . . 35
  - 3.2.1 Barra de ferramentas . . . 35
  - 3.2.2 Painéis . . . 36
- 3.3 Barra de Status . . . 36
  - 3.3.1 Barra de localização . . . 36
  - 3.3.2 Reportando ações . . . 37
  - 3.3.3 Control the map canvas . . . 37
  - 3.3.4 Mensagens . . . 38

4 Painel do Navegador **39**
- 4.1 Resources that can be opened / run from the Browser . . . 41

4.2 Browser panel top-level entries . . . 41
4.2.1 Favoritos . . . 41
4.2.2 Favoritos Espaciais . . . 42
4.2.3 Project Home . . . 42
4.2.4 Drives and file system . . . 42
4.2.5 Data type . . . 43
4.2.6 Entradas no banco de dados . . . 43
4.2.7 Mosaicos e Serviços da Web . . . 47
4.2.8 Project and supported file types . . . 48

5 Configuração QGIS **49**
5.1 Opções . . . 49
5.1.1 Configurações Gerais . . . 50
5.1.2 Configurações de sistema . . . 52
5.1.3 User Profiles Settings . . . 54
5.1.4 Configurações de SRC transformações . . . 54
5.1.5 Configurações de fonte de informação . . . 57
5.1.6 Configurações de renderização. . . . 61
5.1.7 Configurações de tela e legenda . . . 64
5.1.8 Configurações de ferramentas de mapa . . . 66
5.1.9 Elevation settings . . . 70
5.1.10 Configurações 3D. . . . 70
5.1.11 Configurações de cores . . . 71
5.1.12 Configurações de fontes . . . 72
5.1.13 Configurações de compositor . . . 73
5.1.14 Configurações de variáveis . . . 74
5.1.15 Configurações de autenticação . . . 74
5.1.16 Configurações de rede . . . 75
5.1.17 Configurações de GPS . . . 77
5.1.18 Configurações do localizador . . . 79
5.1.19 Configurações de aceleração . . . 81
5.1.20 Configurações de IDE . . . 81
5.1.21 Configurações de processamento . . . 85
5.1.22 Configurações avançadas . . . 85
5.2 Trabalhando com Perfis de Usuário . . . 86
5.2.1 The concept . . . 86
5.2.2 Setting user profile . . . 87
5.3 Propriedades do Projeto . . . 88
5.3.1 Propriedades Gerais . . . 88
5.3.2 Propriedades dos Metadados . . . 91
5.3.3 View Settings . . . 92
5.3.4 CRS Properties . . . 92
5.3.5 Transformations Properties . . . 93
5.3.6 Styles Properties . . . 93
5.3.7 Colors properties . . . 94
5.3.8 Data Sources Properties . . . 95
5.3.9 Propriedades de Relações . . . 97
5.3.10 Propriedades das Variáveis . . . 98
5.3.11 Propriedades das Macros . . . 98
5.3.12 Propriedades do Servidor QGIS . . . 99
5.3.13 Propriedades Temporais . . . 100
5.3.14 Elevation Properties . . . 100
5.3.15 Sensors Properties . . . 103
5.4 Personalização . . . 104
5.5 Atalhos do Teclado . . . 106
5.6 Executando o QGIS com configurações avançadas . . . 107
5.6.1 Command line and environment variables . . . 107
5.6.2 Implantando o QGIS em uma organização . . . 113

6 Trabalhando com Projeções **115**
6.1 Visão geral do Suporte a Projeções . . . 115
6.2 Sistemas de referência de coordenadas da camada . . . 115
6.3 Sistemas referência de coordenadas projetadas . . . 117
6.4 Seletor do Sistemas de Referência de Coordenadas . . . 119
6.5 Sistema de Referência de Coordenadas personalizado . . . 120
6.5.1 Integrate an NTv2-transformation in QGIS . . . 121
6.6 Transformações de Datum . . . 122

7 Visualizing Maps **125**
7.1 2D Map View . . . 125
7.1.1 Explorando a visualização do mapa . . . 126
7.1.2 Controlling map rendering . . . 127
7.1.3 Controle baseado no tempo na tela do mapa . . . 128
7.1.4 The Elevation Controller panel . . . 130
7.1.5 Bookmarking extents on the map . . . 131
7.1.6 Decorating the map . . . 133
7.1.7 Ferramentas de anotação . . . 140
7.1.8 Medição . . . 145
7.1.9 Definição de visualizações adicionais do mapa . . . 148
7.1.10 Exportando a visualização do mapa . . . 149
7.2 Visualização de Mapa 3D . . . 151
7.2.1 Scene Configuration . . . 153
7.2.2 Opções de navegação . . . 161
7.2.3 Criar uma animação . . . 162
7.2.4 Camadas vetoriais 3D . . . 162
7.3 Elevation Profile View . . . 163
7.3.1 The interface . . . 163
7.3.2 Creating an elevation profile . . . 165
7.3.3 Interacting with the profile Views . . . 166
7.3.4 Exporting from the elevation profile . . . 167

8 Ferramentas Gerais **169**
8.1 Conteúdo da ajuda . . . 169
8.2 Painéis . . . 169
8.2.1 Painel Camadas . . . 169
8.2.2 Painel de Estilo de Camada . . . 177
8.2.3 Layer Order Panel . . . 179
8.2.4 Painel geral . . . 179
8.2.5 Log Messages Panel . . . 180
8.2.6 Undo/Redo Panel . . . 180
8.2.7 Statistical Summary Panel . . . 180
8.2.8 Debugging/Development Tools Panel . . . 181
8.3 Incorporando camadas de projetos externos . . . 185
8.4 Interagindo com feições . . . 186
8.4.1 Selecionando características . . . 186
8.4.2 Identificando Feições . . . 189
8.5 Salvar e Compartilhar Propriedades da Camada . . . 193
8.5.1 Gerenciar Estilos Personalizados . . . 193
8.5.2 Storing Styles in a File or a Database . . . 195
8.5.3 Arquivo de definição de camada . . . 197
8.6 Documentando seus dados . . . 197
8.6.1 Metadados . . . 197
8.6.2 Notas de camadas . . . 198
8.7 Armazenando valores em Variáveis . . . 199
8.8 Elementos comuns . . . 201
8.8.1 Seletor de Cor . . . 201
8.8.2 Symbol Widget . . . 205

8.8.3 Seletor de arquivos remoto ou incorporado . . . 205
8.8.4 Visibility Scale Selector . . . 206
8.8.5 Seletor de Extensão Espacial . . . 206
8.8.6 Seletor de Fonte . . . 207
8.8.7 Seletor de Unidade . . . 208
8.8.8 Formatação de números . . . 209
8.8.9 Modos de Mistura . . . 210
8.8.10 Data defined override setup . . . 212
8.8.11 Autenticação . . . 215

9 Subir de nível com expressões **219**
9.1 Expressões . . . 219
9.1.1 O Construtor de string de expressão . . . 219
9.1.2 Editor de Funções . . . 224
9.2 Lista de funções . . . 227
9.2.1 Funções de Agregação . . . 227
9.2.2 Funções de Array . . . 237
9.2.3 Funções de Cor . . . 249
9.2.4 Funções Condicionais . . . 258
9.2.5 Funções de Conversão . . . 261
9.2.6 CRS Functions . . . 267
9.2.7 Funções Personalizadas . . . 267
9.2.8 Funções de Data e Hora . . . 267
9.2.9 Campos e Valores . . . 278
9.2.10 Funções de Arquivos e Caminhos . . . 278
9.2.11 Funções de Formulário . . . 281
9.2.12 Funções de Combinação Fuzzy . . . 281
9.2.13 Funções Gerais . . . 283
9.2.14 Funções de Geometria . . . 285
9.2.15 Funções de Layout . . . 361
9.2.16 Camadas do mapa . . . 362
9.2.17 Funções de mapeamento . . . 365
9.2.18 Funções Matemáticas . . . 369
9.2.19 Funções de Malhas . . . 377
9.2.20 Operadores . . . 379
9.2.21 Funções de Processamento . . . 388
9.2.22 Funções de Rasters . . . 388
9.2.23 Funções de Registro e Atributos . . . 390
9.2.24 Relações . . . 399
9.2.25 Sensors Functions . . . 399
9.2.26 Funções de String . . . 399
9.2.27 Expressões do usuário . . . 410
9.2.28 Variáveis . . . 410
9.2.29 Funções Recentes . . . 414

10 Biblioteca de estilo **415**
10.1 Gerenciador de Estilo . . . 415
10.1.1 The Style Manager dialog . . . 415
10.1.2 Setting a Color Ramp . . . 421
10.1.3 Creating a Legend Patch Shape . . . 423
10.2 O Seletor de Símbolos . . . 425
10.2.1 The symbol layer tree . . . 426
10.2.2 Configurando um símbolo . . . 427
10.3 Configurando um rótulo . . . 445
10.3.1 Formatando o texto do rótulo . . . 447
10.3.2 Configurando a interação com rótulos . . . 456
10.4 Criando símbolos 3D . . . 466
10.4.1 Camadas de Ponto . . . 467

10.4.2 Camadas de linha . . . 468
10.4.3 Camadas de Ppolígonos . . . 469
10.4.4 Shading the texture . . . 471
10.4.5 Application example . . . 471

11 Gerenciando fonte de dados **473**
11.1 Abrir Dados . . . 473
11.1.1 The Browser Panel . . . 475
11.1.2 The DB Manager . . . 478
11.1.3 Provider-based loading tools . . . 479
11.1.4 The Layer Metadata Search Panel . . . 500
11.1.5 Formatos personalizados QGIS . . . 501
11.1.6 QLR - QGIS Layer Definition File . . . 501
11.1.7 Conectando-se a serviços da web . . . 502
11.2 Criando Camadas . . . 510
11.2.1 Criando novas camadas vetoriais . . . 511
11.2.2 Criando novas camadas a partir de uma camada existente . . . 519
11.2.3 Criando novos arquivos DXF . . . 524
11.2.4 Creating new layers from the clipboard . . . 525
11.2.5 Creating SQL Query Layers . . . 526
11.2.6 Criando camadas virtuais . . . 529
11.3 Georreferenciador . . . 531
11.3.1 Georeferencing raster layer . . . 532
11.3.2 Georeferencing vector layer . . . 538
11.3.3 Show and adapt layer properties . . . 539
11.3.4 Configurar o georreferenciador . . . 539
11.4 Explorando Formatos de Dados e Campos . . . 540
11.4.1 Dados raster . . . 540
11.4.2 Dado Vetorial . . . 540
11.4.3 Layers crossing 180° longitude . . . 548

12 Trabalhando com Dados Vetoriais **551**
12.1 Janela de Propriedades de Vetor . . . 551
12.1.1 Propriedades da informação . . . 552
12.1.2 Propriedades de origem . . . 553
12.1.3 Propriedades da simbologia . . . 556
12.1.4 Propriedades das etiquetas . . . 579
12.1.5 Propriedades dos diagramas . . . 591
12.1.6 Propriedades das Máscaras . . . 596
12.1.7 Propriedades da vista 3D . . . 597
12.1.8 Propriedades dos campos . . . 599
12.1.9 Propriedades do formulário de atributos . . . 600
12.1.10 Propriedades de associações . . . 612
12.1.11 Propriedades de armazenamento auxiliar . . . 612
12.1.12 Propriedades de Ações . . . 620
12.1.13 Display Properties . . . 626
12.1.14 Propriedades de Renderização . . . 629
12.1.15 Propriedades Temporais . . . 631
12.1.16 Propriedades das Variáveis . . . 632
12.1.17 Elevation Properties . . . 632
12.1.18 Propriedades dos Metadados . . . 634
12.1.19 Propriedades de Dependências . . . 634
12.1.20 Propriedades da legenda . . . 634
12.1.21 Propriedades do Servidor QGIS . . . 635
12.1.22 Propriedades de Digitalização . . . 637
12.2 Trabalhando com a tabela de atributos . . . 639
12.2.1 Prefácio: Tabelas espaciais e não espaciais . . . 639
12.2.2 Apresentando a Interface da Tabela de Atributos . . . 640

12.2.3 Interagindo com feições em uma tabela de atributos . . . 645
12.2.4 Editando valores de atributo . . . 649
12.2.5 Exploring features attributes through the Identify Tool . . . 653
12.3 Editando . . . 655
12.3.1 Setting the snapping tolerance and search radius . . . 656
12.3.2 Snapping and Digitizing Options . . . 656
12.3.3 Edição Topológica . . . 659
12.3.4 Digitalizar uma camada existente . . . 661
12.3.5 Digitalização Avançada . . . 673
12.3.6 Digitalização de forma . . . 682
12.3.7 O painel Digitalização avançada . . . 685
12.3.8 The Processing in-place layer modifier . . . 693
12.4 Conectando e Editando Dados Através de Camadas . . . 695
12.4.1 Joining features between two layers . . . 695
12.4.2 Setting relations between multiple layers . . . 697
12.4.3 Storing and fetching an external resource . . . 711

13 Trabalhando com Dados Raster **715**
13.1 Caixa de Diálogo de Propriedades do Raster . . . 715
13.1.1 Propriedades da informação . . . 716
13.1.2 Propriedades de origem . . . 716
13.1.3 Propriedades da simbologia . . . 717
13.1.4 Propriedades de Transparência . . . 727
13.1.5 Labels Properties . . . 728
13.1.6 Propriedades do Histograma . . . 730
13.1.7 Propriedades de Renderização . . . 731
13.1.8 Propriedades Temporais . . . 732
13.1.9 Propriedades das Pirâmides . . . 733
13.1.10 Elevation Properties . . . 734
13.1.11 Propriedades dos Metadados . . . 735
13.1.12 Propriedades da legenda . . . 735
13.1.13 Display Properties . . . 736
13.1.14 Propriedades do Servidor QGIS . . . 737
13.1.15 Attribute Tables . . . 738
13.1.16 Identify raster cells . . . 743
13.2 Análise raster . . . 743
13.2.1 Calculadora raster . . . 743

14 Working with Mesh Data **747**
14.1 O que é uma malha? . . . 747
14.2 Formatos suportados . . . 749
14.3 Mesh Dataset Properties . . . 750
14.3.1 Propriedades da informação . . . 751
14.3.2 Propriedades de origem . . . 752
14.3.3 Propriedades da simbologia . . . 753
14.3.4 Propriedades das etiquetas . . . 758
14.3.5 Propriedades da vista 3D . . . 760
14.3.6 Propriedades de Renderização . . . 762
14.3.7 Propriedades Temporais . . . 763
14.3.8 Elevation Properties . . . 763
14.3.9 Propriedades dos Metadados . . . 765
14.4 Exploring mesh datasets . . . 765
14.5 Editando uma camada de malha . . . 766
14.5.1 Visão geral das ferramentas de digitalização de malhas . . . 767
14.5.2 Explorando a lógica de atribuição de valor Z . . . 767
14.5.3 Selecionando elementos de malha . . . 769
14.5.4 Modificando elementos de malha . . . 770
14.5.5 Reindexação de malhas . . . 775

14.6 Calculadora de Malha . . . 776

15 Trabalhando com Blocos Vetoriais **779**
15.1 O que são Blocos Vetoriais? . . . 779
15.2 Formatos Suportados . . . 780
15.3 Vector Tiles Dataset Properties . . . 781
15.3.1 Propriedades da informação . . . 781
15.3.2 Propriedades de origem . . . 781
15.3.3 Symbology and Labels Properties . . . 782
15.3.4 Propriedades de Renderização . . . 783
15.3.5 Propriedades dos Metadados . . . 784
15.4 QGIS Server Properties . . . 784

16 Working with Point Clouds **785**
16.1 Introduction to Point Clouds . . . 785
16.2 Point Clouds Properties . . . 786
16.2.1 Propriedades da informação . . . 786
16.2.2 Propriedades de origem . . . 787
16.2.3 Propriedades da simbologia . . . 790
16.2.4 Propriedades da vista 3D . . . 798
16.2.5 Propriedades de Renderização . . . 800
16.2.6 Elevation Properties . . . 800
16.2.7 Propriedades dos Metadados . . . 802
16.2.8 Statistics Properties . . . 802
16.3 Virtual point cloud . . . 803

17 Working with 3D Tiles **805**
17.1 What are 3D Tiles? . . . 805
17.1.1 State of 3D Tiles Support . . . 806
17.2 3D Tiles Properties . . . 807
17.2.1 Propriedades da informação . . . 807
17.2.2 Propriedades de origem . . . 807
17.2.3 Propriedades da simbologia . . . 808
17.2.4 Propriedades da vista 3D . . . 809
17.2.5 Propriedades de Renderização . . . 809
17.2.6 Elevation Properties . . . 810
17.2.7 Propriedades dos Metadados . . . 810

18 Laying out the maps **811**
18.1 Visão Geral do Layout de Impressão . . . 811
18.1.1 Sessão de Amostra para iniciantes . . . 811
18.1.2 The Layout Manager . . . 812
18.1.3 Menus, tools and panels of the print layout . . . 813
18.2 Itens de layout . . . 826
18.2.1 Opções comuns de itens de esquema . . . 826
18.2.2 Ítem do Mapa . . . 832
18.2.3 O item de mapa 3D . . . 843
18.2.4 A rotulagem de item . . . 844
18.2.5 O Item Legenda . . . 849
18.2.6 Item da Barra de Escala . . . 857
18.2.7 Os itens da tabela . . . 861
18.2.8 Os itens de marcador, imagem e seta do norte . . . 870
18.2.9 The Elevation Profile Item . . . 875
18.2.10 O quadro com item em HTML . . . 876
18.2.11 Itens do shape . . . 879
18.3 Criando um arquivo de Saída . . . 882
18.3.1 Definições para exportação . . . 882
18.3.2 Exportar como Imagem . . . 883
18.3.3 Exportar como SVG . . . 884

18.3.4 Exportar como PDF . . . 885
18.3.5 Gerar um Atlas . . . 887
18.4 Criando um Relatório . . . 893
18.4.1 O que é? . . . 893
18.4.2 Iniciando . . . 893
18.4.3 Espaço de Trabalho do Compositor de Relatório . . . 894
18.4.4 Definições para exportação . . . 909

19 Working with OGC / ISO protocols **911**
19.1 Cliente WMS/WMTS . . . 912
19.1.1 Visão Global do Suporte WMS . . . 912
19.1.2 Visão Global do Suporte WMTS . . . 912
19.1.3 Selecionando os Servidores WMS/WMTS . . . 913
19.1.4 Carregando as camadas WMS/WMTS . . . 915
19.1.5 Mosaicos . . . 917
19.1.6 Usando a Ferramenta Identificar . . . 917
19.1.7 Viewing Properties . . . 918
19.1.8 Mostrar gráfico de legendas WMS no índice e layout . . . 922
19.2 WCS Cliente . . . 922
19.3 WFS e WFS-T Cliente . . . 924
19.4 SensorThings . . . 927
19.4.1 Setting connection . . . 927
19.4.2 Loading SensorThings data . . . 929
19.4.3 Working with a vector layer from SensorThings . . . 930
19.5 STAC (SpatioTemporal Asset Catalogs) . . . 930
19.5.1 Setting connection . . . 931
19.5.2 Browsing STAC Catalogs . . . 932
19.5.3 Filtering and Searching STAC Items . . . 933
19.6 ArcGIS REST Server . . . 935

20 Trabalhando com dados GPS **937**
20.1 Introducing GNSS/GPS Data . . . 937
20.1.1 O que é GPS? . . . 937
20.1.2 Transferring or loading GPS data . . . 938
20.2 Rastreio GPS em tempo real . . . 940
20.2.1 GPS toolbar . . . 940
20.2.2 GPS Information Panel . . . 942
20.2.3 Conectar um GPS com Bluetooth para rasteamento ao vivo . . . 944
20.2.4 Examples of GPS devices connection in QGIS . . . 944

21 Sistema de Autenticação **947**
21.1 Visão Geral do Sistema de Autenticação . . . 947
21.1.1 Banco de Dados de Autenticação . . . 948
21.1.2 Custom authentication databases . . . 948
21.1.3 Senha mestra . . . 949
21.1.4 Configurações de Autenticação . . . 950
21.1.5 Métodos de autenticação . . . 953
21.1.6 Utilitários de configuração de senha mestra e autenticação . . . 960
21.1.7 Usando configurações de autenticação . . . 961
21.1.8 Vinculações do Python . . . 961
21.2 Fluxos de trabalho de autenticação de usuário . . . 962
21.2.1 Autenticação HTTP(S) . . . 962
21.2.2 Autenticação de Base de Dados . . . 963
21.2.3 Autenticação de PKI . . . 964
21.2.4 Lidando com camadas ruins . . . 971
21.2.5 Alterando o ID de configuração de autenticação . . . 972
21.2.6 Suporte ao servidor QGIS . . . 972
21.2.7 Exceções do servidor SSL . . . 973

22 Integração com SIG GRASS **977**
22.1 Demo dataset . . . 977
22.2 Carregando camadas raster e vetorial GRASS . . . 977
22.3 Importing data into a GRASS LOCATION via drag and drop . . . 978
22.4 Managing GRASS data in QGIS Browser . . . 978
22.5 Opções do GRASS . . . 978
22.6 Iniciando o complemento GRASS . . . 978
22.7 Abrindo o mapset GRASS . . . 979
22.8 Localização e MAPSET GRASS . . . 979
22.9 Importando dados para uma localização GRASS . . . 979
22.9.1 Criando uma nova Localização GRASS . . . 980
22.9.2 Adicionando um novo MAPSET . . . 981
22.10 O modelo de dados vetorial do GRASS . . . 982
22.11 Criando uma nova camada vetorial GRASS . . . 982
22.12 Digitalizando e editando uma camada vetorial GRASS . . . 983
22.13 A ferramenta região GRASS . . . 985
22.14 Ferramentas GRASS . . . 986
22.14.1 Trabalhando com módulos GRASS . . . 986
22.14.2 Exemplos de módulos GRASS . . . 989
22.14.3 Personalizando a caixa de ferramentas GRASS . . . 991

23 QGIS estrutura de processamento **993**
23.1 Introducción . . . 993
23.2 Configuring the Processing Framework . . . 996
23.2.1 General . . . 996
23.2.2 Menus . . . 998
23.2.3 Models and Scripts . . . 998
23.2.4 Providers . . . 998
23.3 A Barra de Ferramentas . . . 999
23.3.1 O diálogo do algoritmo . . . 1001
23.3.2 Objetos de dados gerados por algoritmos . . . 1009
23.4 Gerenciador do histórico . . . 1010
23.4.1 O histórico do processamento . . . 1010
23.4.2 O log do processamento . . . 1011
23.5 The model designer . . . 1011
23.5.1 The model designer interface . . . 1012
23.5.2 Criando um modelo . . . 1015
23.5.3 Salvando e carregando os modelos . . . 1024
23.5.4 Editando um modelo . . . 1024
23.6 A interface de processamento em lote . . . 1026
23.6.1 Introducción . . . 1026
23.6.2 Os parâmetros da tabela . . . 1027
23.6.3 Preenchendo os parâmetros da tabela . . . 1028
23.6.4 Executando o processo em lote . . . 1029
23.7 Usando os algoritmos do processamento a partir do Terminal Python. . . . 1029
23.7.1 Chamando os algoritmos a partir do Terminal Python . . . 1030
23.7.2 Criando scripts e correndo-os a partir da caixa de ferramentas . . . 1034
23.7.3 Pré- e pós-execução de encaixe da script . . . 1037
23.8 Usando o processamento da linha de comando . . . 1037
23.9 Escrevendo novos algoritmos de Processamento como scripts Python . . . 1039
23.9.1 Extending QgsProcessingAlgorithm . . . 1039
23.9.2 The @alg decorator . . . 1043
23.9.3 Handing algorithm output . . . 1045
23.9.4 Comunicação com o usuário . . . 1045
23.9.5 Documentando os seus scripts . . . 1046
23.9.6 Bandeiras . . . 1046
23.9.7 Melhores práticas para algoritmos de script escrito . . . 1046
23.10 Tipos de entrada e saída para Algoritmos de Processamento . . . 1046

23.10.1 Tipos de entrada . . . 1046
23.10.2 Tipos de saída . . . 1049
23.11 Configurando as aplicações externas . . . 1050
23.11.1 Uma nota para usuários Windows . . . 1050
23.11.2 Uma nota para os formatos dos arquivos . . . 1050
23.11.3 Uma nota para as seleções da camada vetorial . . . 1051
23.11.4 Using third-party Providers . . . 1051

24 Fornece algoritmos e processamento **1061**
24.1 Provedor de algorítmo QGIS . . . 1061
24.1.1 3D Tiles . . . 1061
24.1.2 Cartografia . . . 1063
24.1.3 Check Geometry . . . 1084
24.1.4 Base de dados . . . 1130
24.1.5 Ferramentas de arquivo . . . 1137
24.1.6 Fix Geometry . . . 1140
24.1.7 GPS . . . 1166
24.1.8 Interpolação . . . 1169
24.1.9 Ferramentas de camada . . . 1182
24.1.10 Malha . . . 1186
24.1.11 Metadata tools . . . 1199
24.1.12 Ferramentas do modelador . . . 1204
24.1.13 Análise de rede . . . 1214
24.1.14 Plots . . . 1228
24.1.15 Point Cloud Conversion . . . 1242
24.1.16 Point Cloud Data Management . . . 1249
24.1.17 Point Cloud Extraction . . . 1262
24.1.18 Análise de dados Raster . . . 1267
24.1.19 Criação de Matriz . . . 1333
24.1.20 Raster terrain analysis . . . 1350
24.1.21 Ferramentas de raster . . . 1364
24.1.22 Análise vetorial . . . 1375
24.1.23 Vector coverage . . . 1403
24.1.24 Criação de vetor . . . 1406
24.1.25 Vector general . . . 1438
24.1.26 Geometria vetorial . . . 1479
24.1.27 Vector overlay . . . 1605
24.1.28 Seleção vetorial . . . 1625
24.1.29 Tabela de vetores . . . 1643
24.1.30 Vector Tiles . . . 1661
24.2 Provedor de algoritmo geoespacial GDAL . . . 1665
24.2.1 Análise de dados Raster . . . 1665
24.2.2 Conversão de raster . . . 1696
24.2.3 Extrair raster . . . 1705
24.2.4 Miscelânea de Raster . . . 1713
24.2.5 Projeções raster . . . 1733
24.2.6 Conversão vetorial . . . 1738
24.2.7 Geoprocessamento vetorial . . . 1744
24.2.8 Diversos vetoriais . . . 1753

25 Complementos **1767**
25.1 Complementos QGIS . . . 1767
25.1.1 Core and External plugins . . . 1767
25.1.2 Diálogo de Complementos . . . 1768
25.2 Usando os plug-ins do núcleo QGIS . . . 1771
25.2.1 Plugin do gerenciador de banco de dados . . . 1771
25.2.2 Plugin do Verificador de Geometria . . . 1776
25.2.3 Cliente do Catálogo Metabusca . . . 1780

25.2.4 Plugin de edição fora de rede . . . 1787
25.2.5 Complementos Verificador de Topologia . . . 1789
25.3 Terminal Python QGIS . . . 1791
25.3.1 O console interativo . . . 1792
25.3.2 O Editor de Código . . . 1794

26 Apêndices **1797**
26.1 Apêndice C: Formatos de arquivo QGIS . . . 1797
26.1.1 QGS/QGZ - O Formato de Arquivo de Projeto QGIS . . . 1797
26.1.2 QLR - O arquivo de definição de camada QGIS . . . 1799
26.1.3 QML - O Formato de Arquivo de Estilo QGIS . . . 1800
26.2 Apêndice D: sintaxe de script QGIS R . . . 1801
26.2.1 Entradas . . . 1802
26.2.2 Saídas . . . 1802
26.2.3 Sumário de Sintaxe para scripts QGIS R . . . 1802
26.2.4 Exemplos . . . 1804
26.3 Apêndice E: Conexões de Rede do Aplicativo QGIS . . . 1806

27 Referências Bibliográficas e Web **1809**

CAPÍTULO 1

Iniciando

Este capítulo fornece uma visão geral rápida da instalação do QGIS, do download de dados de amostra do QGIS e da execução de uma primeira sessão simples de visualização de dados raster e vetoriais.

1.1 Instalando o QGIS

O projeto QGIS fornece maneiras diferentes de instalar o QGIS, dependendo da sua plataforma.

1.1.1 Instalando a partir de binários

Instaladores padrão estão disponíveis para MS Windows e Mac OS. Pacotes binários (rpm e deb) ou repositórios de software são fornecidos para muitos tipos de GNU/Linux.

Para mais informações e instruções para o seu sistema operacional, verifique https://qgis.org/download/.

1.1.2 Instalando a partir da source

If you need to build QGIS from source, please refer to the installation instructions. They are distributed with the QGIS source code in a file called `INSTALL`. You can also find them online at https://github.com/qgis/QGIS/blob/release-3_44/INSTALL.md.

Se você deseja compilar uma versão específica e não a versão em desenvolvimento, substitua `master` pelo branch do release (geralmente no formulário `release-X_Y`) no link mencionado acima (as instruções de instalação podem ser diferentes).

1.1.3 Instalando em mídia externa

É possível instalar o QGIS (com todos os complementos e configurações) em um flashdrive, e. g. pendrive. Isto pode ser feito definindo uma opção *–profiles-path* que substitui o caminho padrão *user profile* e força **QSettings** a usar este diretório, também. Veja a seção *Configurações de sistema* para informações adicionais..

1.1.4 Baixando dados de amostra

This user guide contains examples based on the QGIS sample dataset (also called the `Alaska dataset`). Download the sample data from https://github.com/qgis/QGIS-Sample-Data/archive/master.zip and unzip the archive on any convenient location on your system.

O conjunto de dados do Alasca inclui todos os dados GIS usados para os exemplos e capturas de tela deste guia do usuário; Ele também inclui um pequeno banco de dados GRASS. A projeção para os conjuntos de dados de amostra QGIS é a Alaska Albers Equal Area com unidades em pés. O código EPSG é 2964.

```
PROJCS["Albers Equal Area",
GEOGCS["NAD27",
DATUM["North_American_Datum_1927",
SPHEROID["Clarke 1866",6378206.4,294.978698213898,
AUTHORITY["EPSG","7008"]],
TOWGS84[-3,142,183,0,0,0,0],
AUTHORITY["EPSG","6267"]],
PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],
UNIT["degree",0.0174532925199433,
AUTHORITY["EPSG","9108"]],
AUTHORITY["EPSG","4267"]],
PROJECTION["Albers_Conic_Equal_Area"],
PARAMETER["standard_parallel_1",55],
PARAMETER["standard_parallel_2",65],
PARAMETER["latitude_of_center",50],
PARAMETER["longitude_of_center",-154],
PARAMETER["false_easting",0],
PARAMETER["false_northing",0],
UNIT["us_survey_feet",0.3048006096012192]]
```

Se você pretende usar o QGIS como interface gráfica para GRASS, você pode encontrar uma seleção de locais de amostra (por exemplo, Spearfish ou Dakota do Sul) no site oficial da GRASS GIS, https://grass.osgeo.org/download/data/.

1.2 Iniciando e parando o QGIS

O QGIS pode ser iniciado como qualquer outro aplicativo no seu computador. Isso significa que você pode iniciar o QGIS fazendo:

- using the Applications menu, the Start menu, or the Dock
- clicando duas vezes no ícone na pasta Aplicativos ou no atalho da área de trabalho
- clicando duas vezes em um arquivo de projeto existente do QGIS (com `.qgz` ou `.qgs`). Observe que isso também abrirá o projeto.
- digitando `qgis` em um prompt de comando (assumindo que o QGIS foi adicionado ao seu CAMINHO ou você está na pasta de instalação)

Para parar o QGIS, use:

- a opção do menu *Projeto* ► *Sair do QGIS* ou use o atalho `Ctrl+Q`

- *QGIS* ► *Sair do QGIS*, ou use o atalho `Cmd+Q`
- ou use a cruz vermelha no canto superior direito da interface principal do aplicativo.

1.3 Sessão de amostra: Carregando camadas raster e vetorial

Agora que você tem *QGIS installed* e um *sample dataset* disponíveis, demonstraremos uma primeira sessão de amostra. Neste exemplo, visualizaremos uma camada raster e uma camada vetorial. Nós vamos usar:

- a camada raster `landcover` (`qgis_sample_data/raster/landcover.img`)
- e a camada vetorial `lakes` (`qgis_sample_data/gml/lakes.gml`)

Em que `qgis_sample_data` representa o caminho para o conjunto de dados descompactado.

1. Inicie o QGIS como mostrado em *Iniciando e parando o QGIS*.
2. The data we will be working with are in `Albers Equal Area`, so let's set the project's CRS accordingly:
    1. Click the Select projection button in the bottom right of QGIS interface. The project properties dialog opens with the *CRS* tab active.
    2. Type `2964` in the *Filter* text area.
    3. Select the row with `NAD27 / Alaska Albers` CRS name.

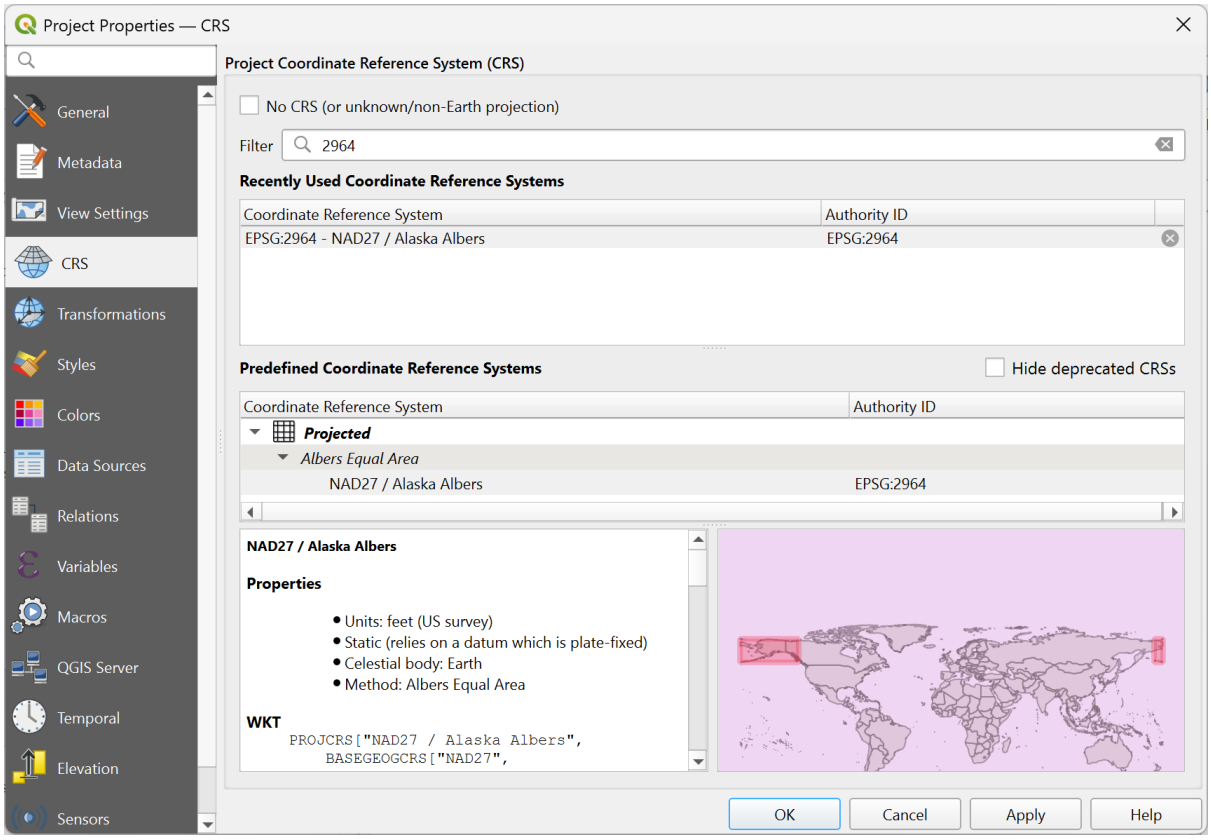


Figura1.1: Selecione o Sistema de Referência de Coordenadas de dados

4. Pressione *OK*

> **Nota**
>
> You can ignore/close for now the “ballpark transform” message that could display.

3. Load the files in QGIS:

   1. Clique no ícone Abrir Gerenciador de Fonte de Dados. O Gerenciador de Fonte de Dados deve abrir em modo Navegador.
   2. Navegue até a pasta `qgis_sample_data/raster/`
   3. Select the ERDAS IMG file *landcover.img* and double-click it. The landcover layer is added in the background while the Data Source Manager window remains open.

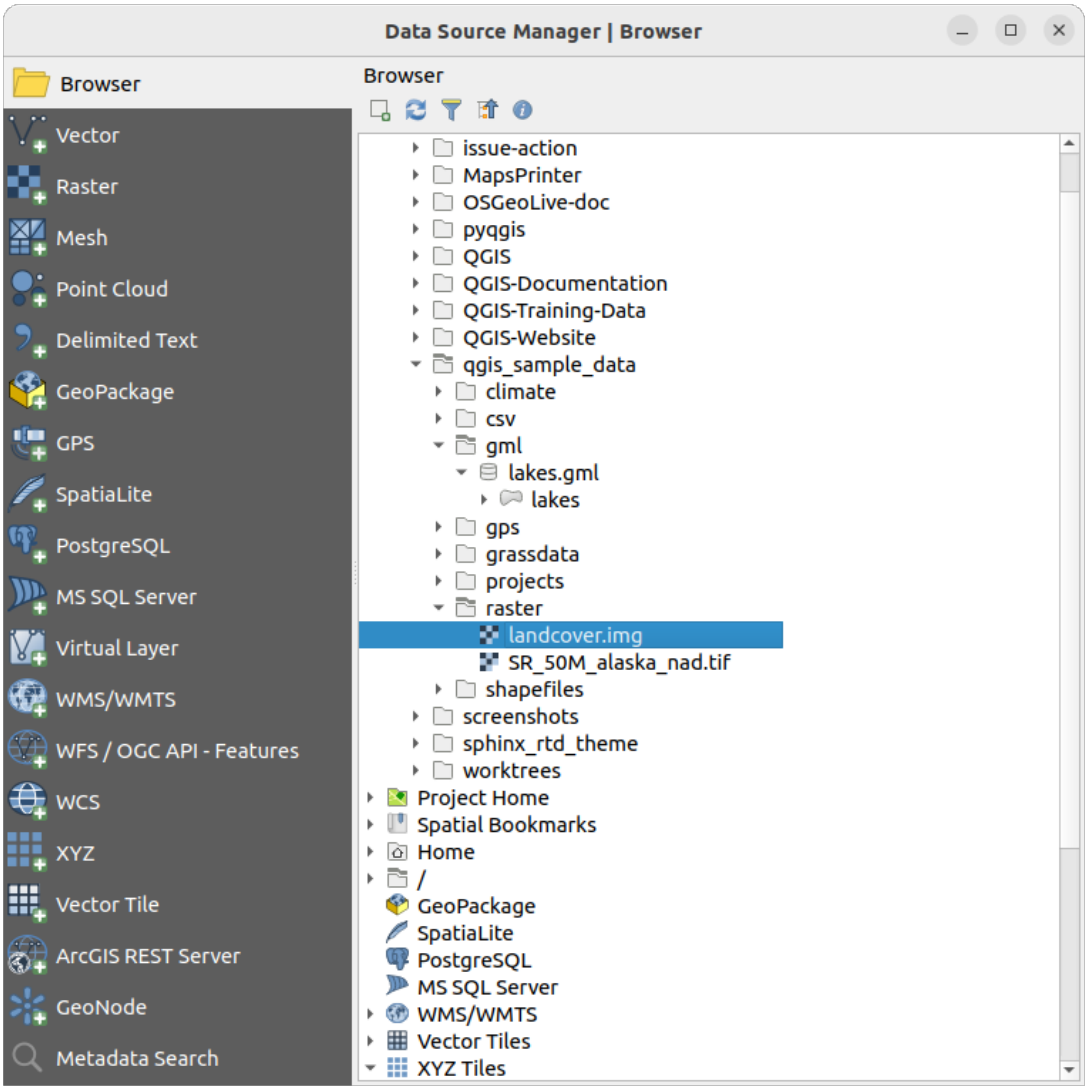


Figura1.2: Adicionando dados a um novo projeto no QGIS

4. To load the lakes data, browse to the folder `qgis_sample_data/gml/`, and drag and drop the *lakes.gml* file over QGIS main dialog. (Or just double-click as mentioned above.)
5. The *Select Items to Add* dialog opens, scanning the file. This is due to `.gml` file format being able to store more than one layer at a time.

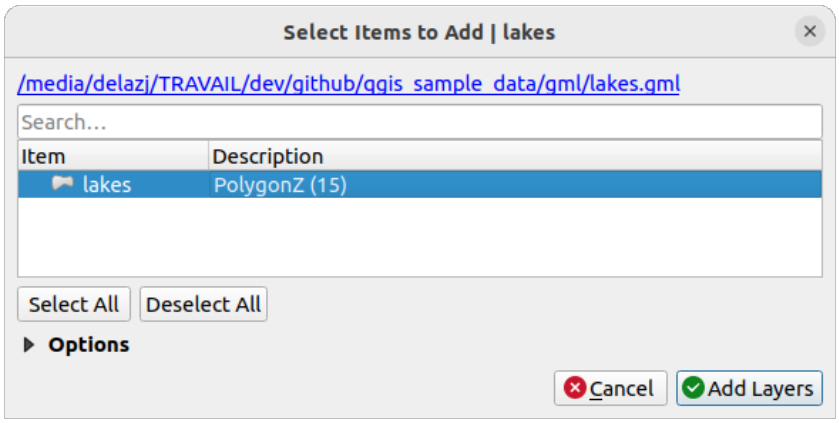


Figura1.3: Select layers within a file

6. In our case there is a single *lakes* layer. Select it and press *Add Layers*.
7. The layer is added to the *Layers* panel

4. Feche a janela Gerenciador de Fonte de Dados

In the *Layers* panel, you can notice that the *lakes* layer displays Layer has no coordinate reference system set next to it. Let's adjust that.

1. Click the icon. The *Coordinate Reference System Selector* dialog opens.
2. As done earlier, find and select the *NAD27 / Alaska Albers* CRS entry.
3. Clique *OK*.

Agora você tem as duas camadas disponíveis no seu projeto em algumas cores aleatórias. Vamos fazer algumas personalizações na camada dos lagos.

1. Selecione o Zoom In na barra de ferramentas *Navegação*
2. Zoom para uma área com alguns lagos
3. Clique duas vezes na camada `lakes` na legenda do mapa para abrir a caixa de diálogo *Propriedades*
4. Para mudar a cor dos lagos:
    1. Clique na guia *Simbologia*.
    2. Selecione azul como cor de preenchimento.

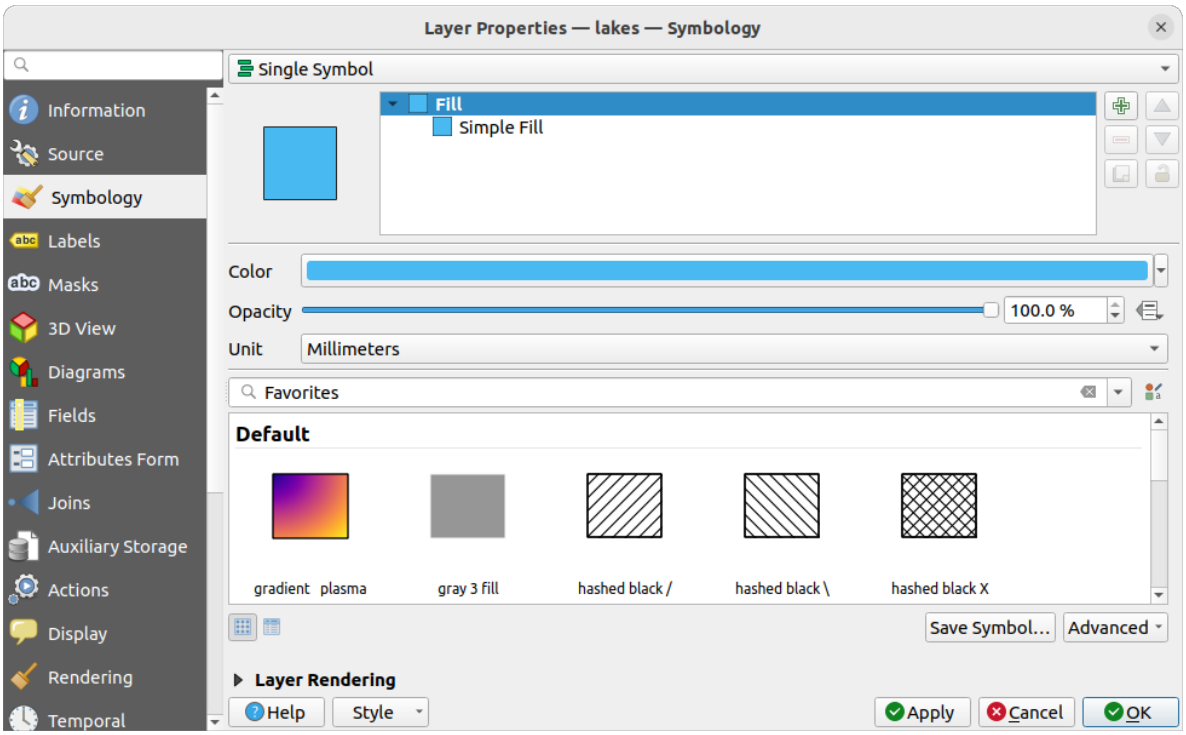


Figura1.4: Selecionando a cor dos lagos

3. Pressione *OK*. Os lagos agora são exibidos em azul na tela do mapa.

5. Para exibir o nome dos lagos:

   1. Reabra a camada `lakes` na caixa de diálogo *Propriedades*
   2. Click on the *Labels* tab
   3. Selecione *Rótulos únicos* no menu drop-down para ativar a marcação.
   4. Na lista *Rotular com*, escolha o campo `NOMES`.

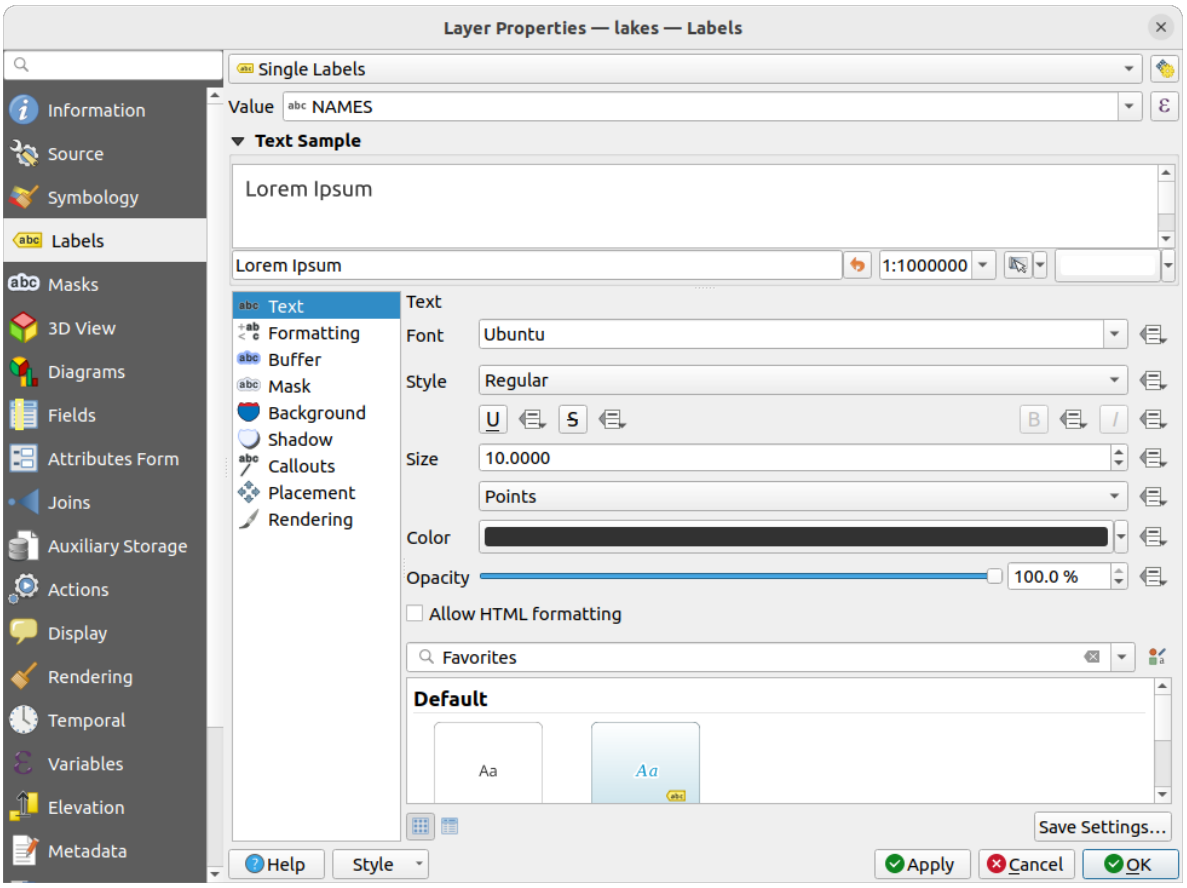


Figura1.5: Mostrar nomes dos Lagos

5. Pressione *Aplicar*. Os nomes agora serão carregados além dos limites.

6. Você pode melhorar a legibilidade dos rótulos adicionando um buffer branco ao redor deles:

   1. Clique na guia *Buffer* na lista à esquerda
   2. Check *Draw text buffer*
   3. Escolha `3` como tamanho do buffer
   4. Clique em *Aplicar*
   5. Verifique se o resultado está bom e atualize o valor, se necessário.
   6. Por fim, clique em *OK* para fechar a caixa de diálogo *Propriedades da Camada* e aplicar as alterações.

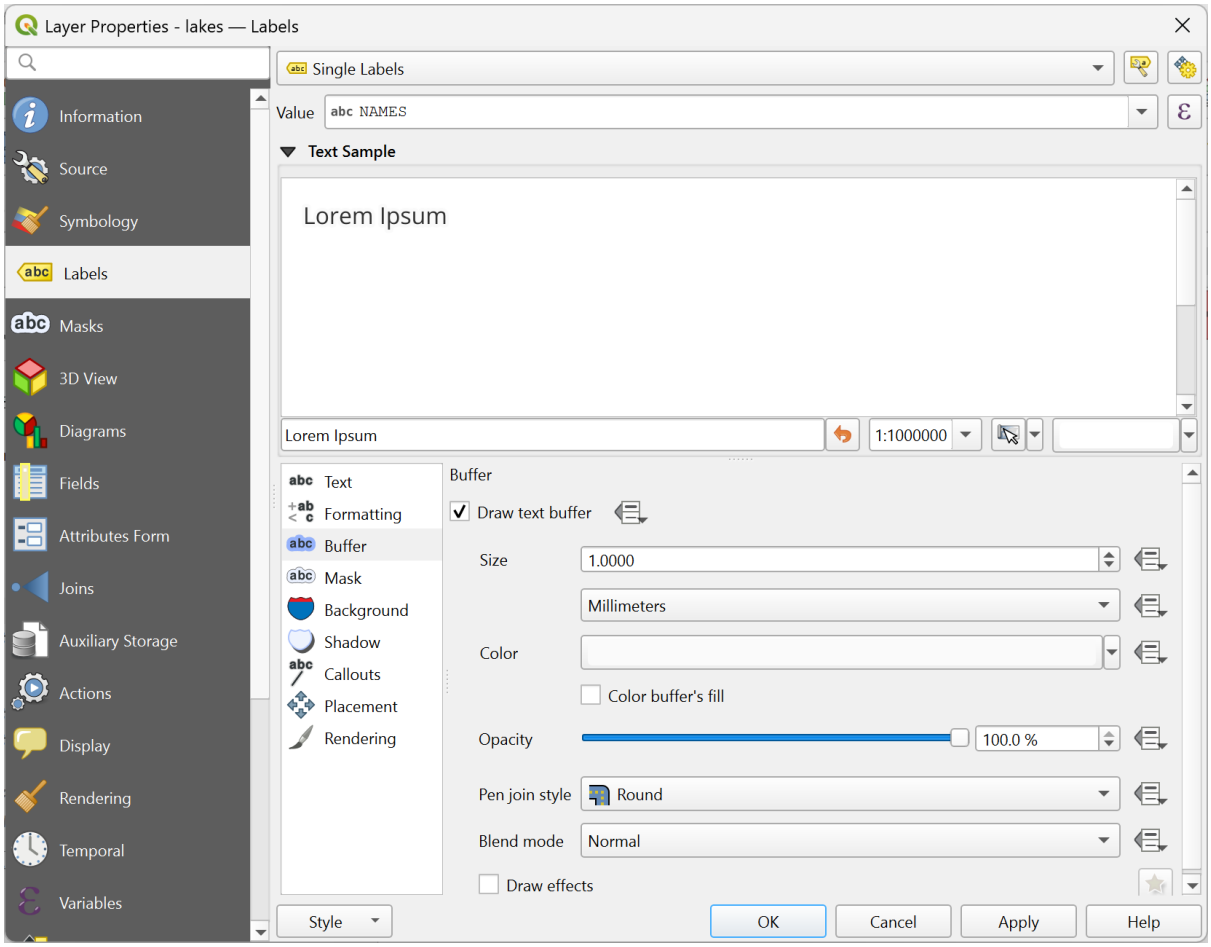


Figura1.6: Showing Buffers around Labels

Vamos agora adicionar algumas decorações para moldar o mapa e exportá-lo para fora do QGIS:

1. Selecione o menu *Exibir* ► *Decorações* ► *Barra de Escala*
2. Na caixa de diálogo aberta, selecione a opção *Ativar Barra de Escala*
3. Personalize as opções da caixa de diálogo como desejar
4. Pressione *Aplicar*
5. Da mesma forma, no menu de decorações, adicione mais itens (seta norte, direitos autorais…) à tela do mapa com propriedades personalizadas.
6. Clique em *Projeto* ► *Importar/Exportar* ► *Exportar Mapa para Imagem…*
7. Pressione *Salvar* na caixa de diálogo aberta
8. Selecione um local de arquivo, um formato e confirme pressionando *Salvar* novamente.
9. Pressione *Projeto* ► *Salvar…* para armazenar suas alterações como um arquivo de projeto `.qgz`.

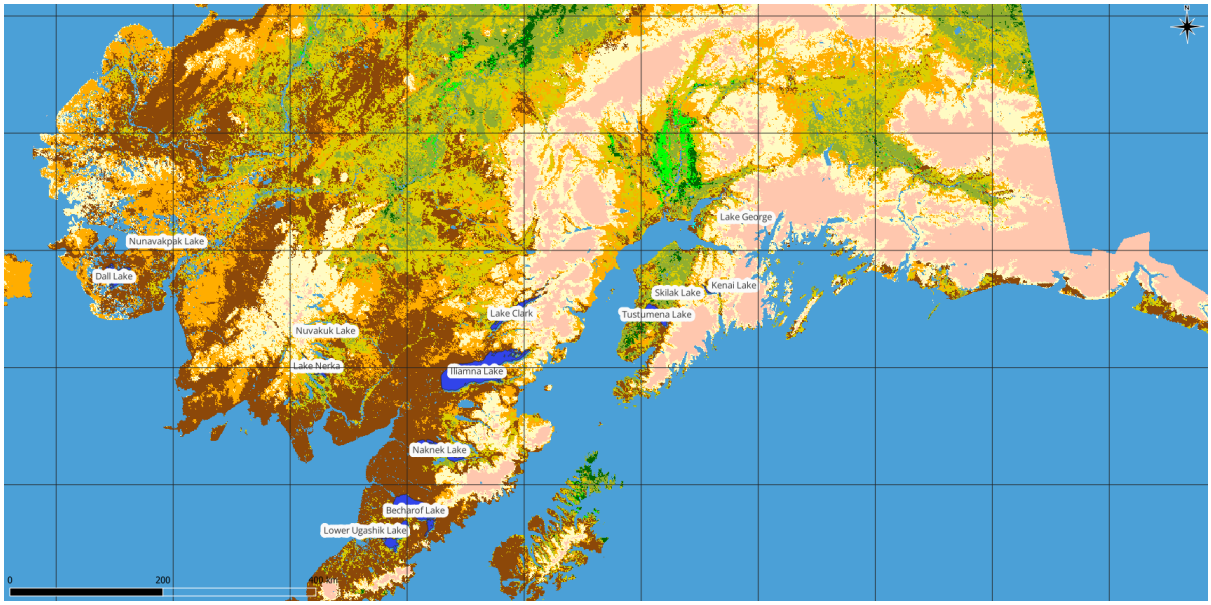


Figura1.7: Showing Exported Map with Decorations

É isso! Você pode ver como é fácil visualizar camadas raster e vetoriais no QGIS, configurá-las e gerar seu mapa em um formato de imagem que você pode usar em outros softwares. Vamos seguir para aprender mais sobre a funcionalidade, recursos e configurações disponíveis e como usá-los.

Nota

Para continuar aprendendo o QGIS através de exercícios passo a passo, siga as instruções em Training manual.

CAPÍTULO 2

Trabalhando com Arquivos de Projetos

2.1 Introduzindo projetos do QGIS

Sua sessão atual do QGIS pode ser chamada de projeto. O QGIS trabalha em um projeto por vez. Uma configuração pode ser específica do projeto ou padrão de toda a aplicação para novos projetos (consulte a seção:ref:*gui_options*). O QGIS pode salvar a sessão do seu espaço de trabalho em: ref: *arquivo de projeto QGIS <qgisprojectfile>* usando as opções de menu: menuselection: *Projeto* -> | fileSave | *Salvar* ou: menuselection:` Projeto -> | *fileSaveAs* | *:menuselection:`Salvar como ….*

Nota

Se o projeto tiver sido modificado, o símbolo * aparecerá na barra de título e QGIS perguntará, por padrão, se você gostaria de salvar as mudanças. Este comportamento é controlado pela caixa de checagem |[caixa de checagem] : guilabel:*Pedido para salvar as alterações do projeto e da fonte de dados quando necessário* em *Configurações* ► *Opções* ► *Geral*.

You can load existing projects through *Project* ► *Open…*, *Project* ► *New from template* or *Project* ► *Open Recent* ►.

From the *Browser panel* you can *Open Project*, *Extract Symbols…* – opens the style manager to export symbols to XML, add them to the default style, or export as PNG/SVG. You can also access the *File Properties…* dialog. The project file can be expanded to see its layers. The context menu of a layer offers the same actions as elsewhere in the browser.

At startup, a list of *Project Templates* and *Recent Projects* are displayed, including screenshots, names and file paths (for up to ten projects). The *Recent Projects* list is handy to access recently used projects. Double-click an entry to open the project or project template. Right-click an entry to *Pin to List*, *Open Directory…* or *Remove from List*. If you see your project on the *Recent Projects* list but can't find it in your file manager use the *Open Directory…* option to help you locate projects that may be missing, moved or renamed. You can also go to *Clear List* if you want to remove all projects from the *Recent Projects* list. If you have pinned projects, the *Clear List* action will be followed by message box asking whether the pinned projects should also be removed. You can also add a layer to create a new project automatically. The lists will then disappear, giving way to the map canvas.

Se você quiser limpar sua sessão e iniciar uma nova, vá para *Projeto* ► **|fileNovo|** *Novo*. Isso irá possibilitar salvar o projeto existente caso mudanças tenham sido feitas desde que este tenha sido aberto ou salvo.

Quando você abre um novo projeto, a barra do título mostrará "Projeto sem título" até que você o salve.

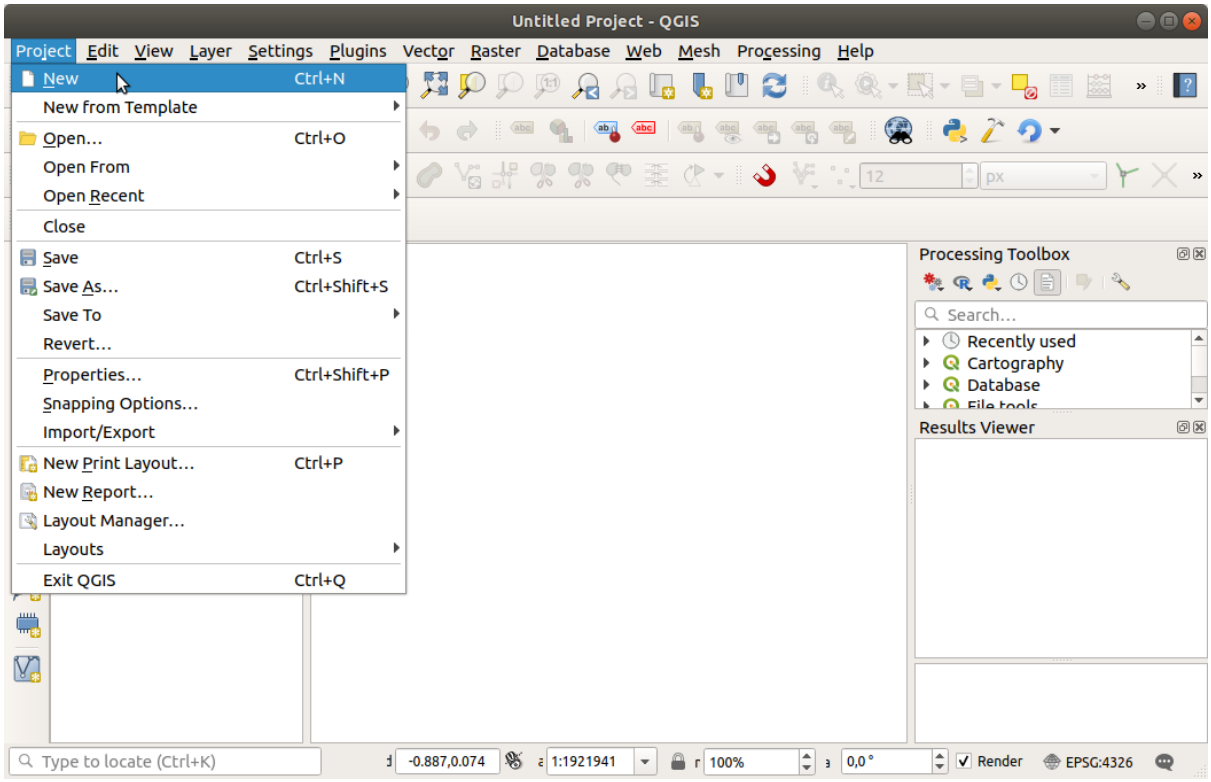


Figura2.1: Iniciando um novo projeto no QGIS

A informação salva em um arquivo de projeto inclui:

- Camadas adicionadas
- Quais camadas podem ser consultadas
- Propriedades das camadas, incluindo simbologia e estilos
- Notas de camadas
- 2D and 3D map views
- Projection for each map view
- Last viewed extent for each map
- Leiautes de impressão
- Elementos de leiaute de impressão com definições
- Definições de leiaute de impressão de atlas
- Definições de digitalização
- Relações de tabelas
- Macros de projetos
- Estilos pré-definidos de projeto
- Definições de plugins
- Definições do QGIS Server a partir das definições de tabela OWS nas propriedades do projeto
- Consultas armazenadas no gestor de banco de dados

O arquivo do projeto é salvo no formato XML (see *QGS/QGZ - O Formato de Arquivo de Projeto QGIS*). Isto significa que é possível editar o arquivo fora do QGIS caso saiba o que está fazendo. O formato do arquivo de projeto foi

atualizado diversas vezes. Arquivos de projetos de versões anteriores do QGIS podem não mais funcionar corretamente.

Nota

Por definição, o QGIS irá advertir sobre diferenças nas versões. Este comportamento é controlado no *Geral* tab of *Configurações* ► *Opções* (*Verificar a versão do QGIS ao iniciar*).

Sempre que salvar um arquivo de projeto `.qgs` no QGIS, um backup do arquivo é criado no mesmo diretório do arquivo de projeto, com a extensão `.qgs~`.

The extension for QGIS projects is `.qgs` but when saving from QGIS, the default is to save using a compressed format with the `.qgz` extension. The `.qgs` file is embedded in the `.qgz` file (a zip archive), together with its associated SQLite database (`.qgd`) for *auxiliary data*. You can get to these files by unzipping the `.qgz` file.

Nota

O mecanismo *Propriedades de armazenamento auxiliar* torna um projeto zipado particularmente útil, desde que incorpore dados auxiliares

Projects can also be saved/loaded to/from a PostgreSQL, GeoPackage or Oracle database using the following Project menu items:

- *Projeto* ► *Abrir a partir de*
- *Projeto* ► *Guardar para*

Both menu items have a sub-menu with a list of extra project storage implementations (PostgreSQL, GeoPackage and Oracle). Clicking the action will open a dialog to pick a GeoPackage connection and project, a PostgreSQL connection, schema and project or Oracle connection, owner and project.

Projects stored in GeoPackage, PostgreSQL or Oracle can also be loaded through the QGIS browser panel, either by double-clicking them or by dragging them to the map canvas.

2.2 Manipulação de caminhos de arquivos quebrados

Ao abrir um projeto, o QGIS pode não alcançar algumas fontes de dados devido à indisponibilidade de serviço/base de dados, ou a um arquivo renomeado ou movido. O QGIS abre então uma janela :guilabel:`Manusear camadas indisponíveis', referenciando as camadas infundadas. Você pode:

- Double-click in the *Datasource* field, adjust the path of each layer and click *Apply changes*;
- Selecione uma linha, pressione *Navegar* para indicar a localização correta e clique *Aplicar mudanças*;
- Pressione *Auto-pesquisa* para navegar pelas pastas e tentar consertar automaticamente todos os caminhos quebrados ou selecionados. Esteja ciente de que a navegação pode levar algum tempo. Depois clique em *Aplicar mudanças*.
- Ignore the message and open your project with the broken path(s) by clicking *Keep Unavailable Layers*. Your layer is then displayed in the *Layers* panel, but without any data until you fix the path using the

icon next to it in the *Layers* panel, or *Repair Data Source...* in the layer contextual menu.

  Com a ferramenta *Reparar fonte de dados...*, uma vez que um caminho de camadas tenha sido fixado, o QGIS escaneia todos os outros caminhos quebrados e tenta fixar automaticamente aqueles que têm o mesmo caminho de arquivo quebrado.

- :guilabel:`Remover camadas indisponíveis' do projeto.

Launching QGIS from command line using the *--skipbadlayers* option can help you skip the *Handle Unavailable Layers* dialog at startup.

2.3 Gerando saída

Há diversas maneiras de gerar saída de uma sessão QGIS. Já discutimos salvar como um arquivo de projeto em *Introduzindo projetos do QGIS*. Outras maneiras de produzir arquivos de saída são:

- Creating images: *Project* ► *Import/Export* ► *Export Map to Image…* outputs the map canvas rendering to an image format (PNG, JPG, TIFF…) at custom scale, resolution and size. Including georeference information in the exported image is possible, simply enable *Append georeference information (embedded or via world file)*. See *Exportando a visualização do mapa* for more details.
- Exportar para arquivos PDF: *Projeto* ► *Importar/Exportar* ► *Exportar Mapa como PDF…* resulta numa tela de mapa renderizado para PDF em escala e resolução personalizados e com algumas definições avançadas (simplificação, georreferenciamento, …). Veja *Exportando a visualização do mapa* para mais detalhes.
- Exportar para arquivos DXF: *Projeto* ► *Importar/Exportar* ► *Exportar Projeto como DXF…* abre uma caixa de diálogo onde pode-se definir o 'Modo de simbologia', a ' Escala da Simbologia' e as camadas vetoriais que se deseja exportar para DXF. Através do 'Modo de simbologia', símbolos da simbologia orifinal do QGIS podem ser exportados com alta fidelidade (veja seção *Criando novos arquivos DXF*).
- Designing maps: *Project* ► *New Print Layout…* opens a dialog where you can layout and print the current map canvas (see section *Laying out the maps*).

CAPÍTULO 3

Interface do QGIS

A interface gráfica do usuário do QGIS (GUI) é mostrada na figura abaixo (os números de 1 a 5 em círculos amarelos indicam elementos importantes da GUI do QGIS e são discutidos abaixo).

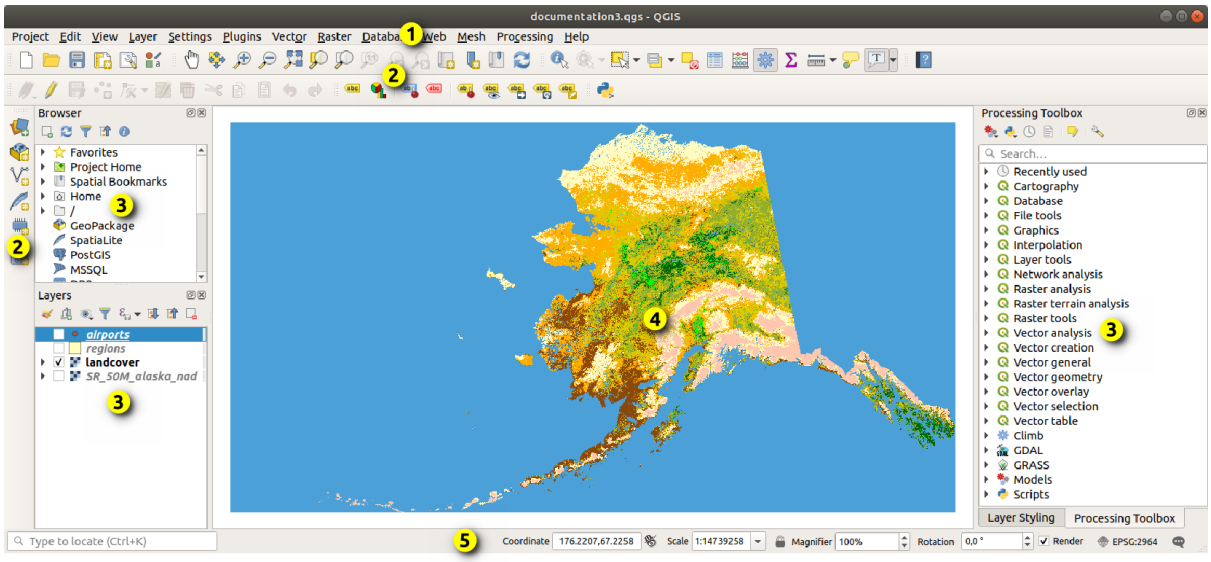


Figura3.1: GUI QGIS com dados de amostra do Alasca

Nota

Suas decorações de janela (barra de título, etc) podem parecer diferentes dependendo do seu sistema operacional e gerenciador de janelas.

A principal GUI QGIS (Fig.3.1) consiste em cinco componentes / tipos de componentes:

1. *Menu Bar*
2. *Toolbars*
3. *Panels*

4. *Map View*
5. *Status Bar*

Role para baixo para informações mais detalhadas destes.

3.1 Barra de Menu

A barra de menus fornece acesso às funções QGIS usando menus hierárquicos padrão. Os Menus, suas opções, ícones associados e atalhos de teclado são descritos abaixo. Os atalhos de teclado podem ser reconfigurados (*Configurações ► Atalhos de Teclado*).

A maioria das opções de menu possui uma ferramenta correspondente e vice-versa. No entanto, os menus não estão organizados exatamente como as barras de ferramentas. Os locais das opções de menu nas barras de ferramentas são indicados abaixo na tabela. Os complementos podem adicionar novas opções aos menus. Para mais informações sobre ferramentas e barras de ferramentas, veja *Barra de ferramentas*.

Nota

O QGIS é um aplicativo multiplataforma. As ferramentas geralmente estão disponíveis em todas as plataformas, mas podem ser colocadas em menus diferentes, dependendo dos sistemas operacionais. As listas abaixo mostram os locais mais comuns, incluindo variações conhecidas.

3.1.1 Projeto

O menu *Projeto* fornece pontos de acesso e saída para *project files*. Fornece ferramentas para:

- Criar um *Novo* arquivo de projeto do zero ou utilizar outro arquivo de projeto como modelo (veja *Opções de arquivos de projeto* para configuração de modelo)
- *Open…* a project from a file, a GeoPackage, PostgreSQL or Oracle database
- guilabel:*Fechar* um projeto ou revertê-lo para seu último estado salvo
- *Save* a project in `.qgs` or `.qgz` file format, either as a file or within a GeoPackage, PostgreSQL or Oracle database
- Exporta a tela do mapa para diferentes formatos ou usa um *print layout* para saída mais complexa
- Definir as propriedades do projeto e as opções de snapping para edição de geometria.

Tabela3.1: The Project menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Novo	`Ctrl+N`	*Projeto*	*Introduzindo projetos do QGIS*
Novo a partir do modelo ►			*Introduzindo projetos do QGIS*
Abrir…	`Ctrl+O`	*Projeto*	*Introduzindo projetos do QGIS*
Abrir de ►			*Introduzindo projetos do QGIS*
► GeoPackage…			*Introduzindo projetos do QGIS*
► PostgreSQL…			*Introduzindo projetos do QGIS*
► Oracle…			*Introduzindo projetos do QGIS*
Abrir Recente ►	`Alt+J + R`		*Introduzindo projetos do QGIS*
► Clear List			*Introduzindo projetos do QGIS*

continua na próxima página

Tabela 3.1 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Fechar			*Introduzindo projetos do QGIS*
Salvar	`Ctrl+S`	*Projeto*	*Introduzindo projetos do QGIS*
Salvar Como…	`Ctrl+Shift+S`	*Projeto*	*Introduzindo projetos do QGIS*
Salvar para ►			*Introduzindo projetos do QGIS*
► *Modelos…*			*Introduzindo projetos do QGIS*
► *GeoPackage…*			*Introduzindo projetos do QGIS*
► *PostgreSQL…*			*Introduzindo projetos do QGIS*
Reverter…			
|propriedadesDoProjeto| *Propriedades…*	`Ctrl+Shift+P`		*Propriedades do Projeto*
Snapping Options…			*Snapping and Digitizing Options*
Importar/Exportar ►			
► *Exportar Mapa para Imagem…*			*Exportando a visualização do mapa*
► *Exportar Mapa como PDF…*			*Exportando a visualização do mapa*
► *Exportar Projeto para DXF…*			*Criando novos arquivos DXF*
► *Importar Camadas a partir de DWG/DXF…*			*Importando um arquivo DXF ou DWG*
Novo layout de impressão…	`Ctrl+P`	*Projeto*	*Laying out the maps*
New Report…			*Criando um Relatório*
Gerenciador de Layout…		*Projeto*	*Laying out the maps*
Layouts ►			*Laying out the maps*
Models ►			*The model designer*
Sair do QGIS	`Ctrl+Q`		

Sob X no macOS, o comando *Sair do QGIS* corresponde a *QGIS* ► *Sair do QGIS* (`Cmd+Q`).

3.1.2 Editar

The *Edit* menu provides most of the native tools needed to edit layer attributes or geometry. To enable *Edit* menu options you need to switch to editing mode by clicking on Toggle editing (see *Editando* for details).

Tabela3.2: The Edit menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Desfazer	`Ctrl+Z`	*Digitalização*	*Retroceder e Retomar*
Refazer	`Ctrl+Shift+Z`	*Digitalização*	*Retroceder e Retomar*
Cortar Feições	`Ctrl+X`	*Digitalização*	*Cortando, Copiando e Colando Elementos*

continua na próxima página

Tabela 3.2 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Copiar Feições	`Ctrl+C`	*Digitalização*	*Cortando, Copiando e Colando Elementos*
Colar Feições	`Ctrl+V`	*Digitalização*	*Cortando, Copiando e Colando Elementos*
Colar Feições como ►			*Trabalhando com a tabela de atributos*
► *Nova camada vetorial…*			*Trabalhando com a tabela de atributos*
► *Temporary Scratch Layer…*	`Ctrl+Alt+V`		*Trabalhando com a tabela de atributos*
Deletar Selecionados		*Digitalização*	*Apagando os Elementos Selecionados*
Selecionar ►			*Selecionando características*
► *Selecione a(s) feição(ões)*		*Seleção*	*Selecionando características*
► *Selecionar Feições por Polígono*		*Seleção*	*Selecionando características*
► *Selecionar Feições a Mão LIvre*		*Seleção*	*Selecionando características*
► *Selecione feições por raio*		*Seleção*	*Selecionando características*
► *Selecione Feições por Valor……*	`F3`	*Seleção*	*Selecionando características*
► *Selecione Feições por Expressão…*	`Ctrl+F3`	*Seleção*	*Selecionando características*
► *Desmarcar feições de todas as camadas*	`Ctrl+Alt+A`	*Seleção*	*Selecionando características*
► *Desmarcar feições da camada ativa atual*	`Ctrl+Shift+A`	*Seleção*	*Selecionando características*
► *Reselecionar feições*			*Selecionando características*
► *Selecionar Todas as Feições*	`Ctrl+A`	*Seleção*	*Selecionando características*
► *Inverter seleção de feições*		*Seleção*	*Selecionando características*
Adicionar Registro	`Ctrl+.`	*Digitalização*	
Adicionar Feição de Ponto	`Ctrl+.`	*Digitalização*	*Adicionando Elementos*
Add Line Feature	`Ctrl+.`	*Digitalização*	*Adicionando Elementos*

continua na próxima página

Tabela 3.2 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Add Polygon Feature	`Ctrl+.`	*Digitalização*	*Adicionando Elementos*
Adicionar Nota ►			*Ferramentas de anotação*
► *Form Annotation*		*Annotations*	*Ferramentas de anotação*
► *HTML Annotation*		*Annotations*	*Ferramentas de anotação*
Editar Atributos ►			
► *Modify Attributes of Selected Features*		*Digitalização*	*Editando valores de atributo*
► *Merge Attributes of Selected Features*		*Advanced Digitizing*	*Juntar os atributos dos elementos selecionados*
:menuselecao:`Editar Geometria -->`			
► *Move Feature(s)*		*Advanced Digitizing*	*Mover feições*
► *Copy and Move Feature(s)*		*Advanced Digitizing*	*Mover feições*
► *Rotate Feature(s)*		*Advanced Digitizing*	*Rodar Elemento(s)*
► *Scale Feature(s)*		*Advanced Digitizing*	*Scale Feature*
► *Simplify Feature*		*Advanced Digitizing*	*Simplificar elemento*
► *Add Ring*		*Advanced Digitizing*	*Adicionar Anel*
► *Adicionar Parte*		*Advanced Digitizing*	*Adicionar Parte*
► *Fill Ring*		*Advanced Digitizing*	*Preenchimento Anel*
► *Delete Ring*		*Advanced Digitizing*	*Apagar Anel*
► *Delete Part*		*Advanced Digitizing*	*Apagar Parte*
► *Reshape Features*		*Advanced Digitizing*	*Refazer elementos*
► *Offset Curve*		*Advanced Digitizing*	*Curvas de Afastamento*
► *Split Features*		*Advanced Digitizing*	*Dividir Elementos*
► *Split Parts*		*Advanced Digitizing*	*Dividindo partes*

continua na próxima página

Tabela 3.2 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
► *Merge Selected Features*		*Advanced Digitizing*	*Juntar elementos selecionados*
► *Vertex Tool (All Layers)*		*Digitalização*	*Ferramenta Vertex*
► *Vertex Tool (Current Layer)*		*Digitalização*	*Ferramenta Vertex*
► *Reverse Line*		*Advanced Digitizing*	*Linha Invertida*
► *Trim/extend Feature*		*Advanced Digitizing*	*Recortar / estender o recurso*
Rotate Point Symbols		*Advanced Digitizing*	*Rodar Símbolos de Pontos*
Offset Point Symbols		*Advanced Digitizing*	*Offset Point Symbols*

As ferramentas que dependem do tipo de geometria da camada selecionada, ex., ponto, polilinha ou polígono, são ativadas de acordo:

Tabela3.3: The “Move feature” geometry based icons

Opção de menu	Ponto	Linha	Polígono
Mover Feição(ões)			
Copiar e Mover Elemento(s)			

3.1.3 Ver

The map is rendered in map views. You can interact with these views using the *View* tools. For example, you can:

- Criar novas vistas de mapa 2D ou 3D ao lado da tela principal do mapa
- *Zoom or pan* para qualquer lugar
- Query the attributes or geometry of the displayed features
- Aprimore o mapa com modos de visualização, anotações ou decorações
- Acesse qualquer painel ou barra de ferramentas

O menu também permite reorganizar a própria interface do QGIS usando ações como:

- *Alternar Modo de Ecrã Completo*: preenche a totalidade do ecrã e oculta a barra de título
- *Toggle Panel Visibility*: shows or hides enabled *panels* - useful when digitizing features (for maximum canvas visibility) as well as for (projected/recorded) presentations using QGIS main canvas
- *Toggle Map Only*: hides panels, toolbars, menus and status bar and only shows the map canvas. Combined with the full screen option, it makes your screen display only the map.

Tabela3.4: The View menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Nova Visualização de Mapa	`Ctrl+M`	*Navegação de mapa*	*2D Map View*
3D Map Views ►			*Visualização de Mapa 3D*
► *New 3D Map View*	`Ctrl+Alt+M`	*Navegação de mapa*	*Visualização de Mapa 3D*
► *Manage 3D Map Views*			*Visualização de Mapa 3D*
Pan Map		*Navegação de mapa*	*Explorando a visualização do mapa*
Pan Map to Selection		*Navegação de mapa*	*Explorando a visualização do mapa*
Zoom In	`Ctrl+Alt++`	*Navegação de mapa*	*Explorando a visualização do mapa*
Reduzir Zoom	`Ctrl+Alt+-`	*Navegação de mapa*	*Explorando a visualização do mapa*
Data Filtering ►			*2D Map View*
► *Temporal Controller*		*Navegação de mapa*	*O painel de controle temporal*
► *Elevation Controller*			*The Elevation Controller panel*
Identify Features	`Ctrl+Shift+I`	*Atributos*	*Identificando Feições*
Medir ►		*Atributos*	*Medição*
► *Measure Line*	`Ctrl+Shift+M`	*Atributos*	*Medição*
► *Measure Area*	`Ctrl+Shift+J`	*Atributos*	*Medição*
► *Measure Bearing*		*Atributos*	*Medição*
► *Measure Angle*		*Atributos*	*Medição*
Statistical Summary		*Atributos*	*Statistical Summary Panel*
Elevation Profile			*Elevation Profile View*
Zoom Full	`Ctrl+Shift+F`	*Navegação de mapa*	*Explorando a visualização do mapa*
Zoom Para Seleção	`Ctrl+J`	*Navegação de mapa*	*Explorando a visualização do mapa*

continua na próxima página

Tabela 3.4 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Aproximar para camada(s)		*Navegação de mapa*	*Explorando a visualização do mapa*
Aproximar para Resolução Nativa (100%)		*Navegação de mapa*	*Explorando a visualização do mapa*
Zoom Last		*Navegação de mapa*	*Explorando a visualização do mapa*
Zoom Next		*Navegação de mapa*	*Explorando a visualização do mapa*
Decorações ►	`Alt+V+D`		*Decorating the map*
► *Grid…*			*Malha*
► *Barra de escala…*			*Barra de Escala*
► *Imagem…*			*Image Decoration*
► *North Arrow…*			*Seta Norte*
► *Title Label…*			*Title Label*
► *Rótulo de direitos autorais…*			*Copyright Label*
► *Layout Extents…*			*Extensões de Layout*
Modo de Pré-visualização ►			
► *Normal*			
► *Simular Monocromático*			
► *Simulate Achromatopsia Color Blindness (Grayscale)*			
► *Simulate Protanopia Color Blindness (No Red)*			
► *Simulate Deuteranopia Color Blindness (No Green)*			
► *Simulate Tritanopia Color Blindness (No Blue)*			
Mostrar dicas de mapas		*Atributos*	*Display Properties*
New Spatial Bookmark…	`Ctrl+B`	*Navegação de mapa*	*Bookmarking extents on the map*
Show Spatial Bookmarks	`Ctrl+Shift+B`	*Navegação de mapa*	*Bookmarking extents on the map*
Show Spatial Bookmark Manager			*Bookmarking extents on the map*
Atualizar	`F5`	*Navegação de mapa*	
Visibilidade da camada ►			*Painel Camadas*

continua na próxima página

Tabela 3.4 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
► *Mostrar todas as camadas*	`Ctrl+Shift+U`		*Painel Camadas*
► *Ocultar todas as camadas*	`Ctrl+Shift+H`		*Painel Camadas*
► *Mostrar camadas selecionadas*			*Painel Camadas*
► *Ocultar camadas selecionadas*			*Painel Camadas*
► *Alternar camadas selecionadas*			*Painel Camadas*
► *Toggle Selected Layers Independently*			*Painel Camadas*
► *Ocultar camadas deselecionadas*			*Painel Camadas*
Painéis ►			*Painéis e Barras de Ferramentas*
Barra de Ferramentas ►			*Painéis e Barras de Ferramentas*
Alternar Modo de Tela Cheia	`F11`		
Toggle Panel Visibility	`Ctrl+Tab`		
Toggle Map Only	`Ctrl+Shift+Tab`		

Under Linux KDE, *Panels* ►, *Toolbars* ► and *Toggle Full Screen Mode* are in the *Settings* menu.

3.1.4 Camada

O menu *Layer* fornece um grande conjunto de ferramentas para novas fontes de dados *create* , *add* as fontes a um projeto ou *save modifications* a elas. Usando as mesmas fontes de dados, você também pode:

- *Duplicate* a layer to generate a copy where you can modify the name, style (symbology, labels, …), joins, … The copy uses the same data source as the original.
- *Copy* and *Paste* layers or groups from one project to another as a new instance whose properties can be modified independently. As for *Duplicate*, the layers are still based on the same data source.
- ou *Incorporar camadas e grupos* … de outro projeto, como cópias somente leitura que você não pode modificar (veja *Incorporando camadas de projetos externos*)

The *Layer* menu also contains tools to configure, copy or paste layer properties (style, scale, CRS…).

Tabela3.5: The Layer menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Data Source Manager	`Ctrl+L`	*Data Source Manager*	*Opening Data*
Criar Camada ►			*Criando novas camadas vetoriais*
► *New GeoPackage Layer…*	`Ctrl+Shift+N`	*Data Source Manager*	*Criando uma nova camada GeoPackage*
► *New Shapefile Layer…*		*Data Source Manager*	*Criando uma nova camada Shapefile*
► *New SpatiaLite Layer…*		*Data Source Manager*	*Creating a new SpatiaLite layer*

continua na próxima página

Tabela 3.5 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
► *New Temporary Scratch Layer…*		*Data Source Manager*	*Creating a new Temporary Scratch Layer*
► *New Mesh Layer…*		*Data Source Manager*	*Criando uma nova camada de malha*
► *New GPX Layer…*		*Data Source Manager*	*Criando uma nova camada GPX*
► *Nova Camada Virtual…*		*Data Source Manager*	*Criando camadas virtuais*
Adicionar Camada ►			*Abrir Dados*
► *Adicionar camada vetorial…*	`Ctrl+Shift+V`	*Gerenciar camadas*	*Carregando uma camada de um arquivo*
► *Adicionar Camada Raster…*	`Ctrl+Shift+R`	*Gerenciar camadas*	*Carregando uma camada de um arquivo*
► *Add Mesh Layer…*		*Gerenciar camadas*	*Carregando uma camada de malha*
► *Adicionar Camada de Texto Delimitado…*	`Ctrl+Shift+T`	*Gerenciar camadas*	*Importando um arquivo de texto delimitado*
► *Add PostgreSQL Layer…*	`Ctrl+Shift+D`	*Gerenciar camadas*	*Database related tools*
► *Add SpatiaLite Layer…*	`Ctrl+Shift+L`	*Gerenciar camadas*	*Camadas SpatiaLite*
► *Add MS SQL Server Layer…*		*Gerenciar camadas*	*Database related tools*
► *Adicionar camada espacial Oracle…*		*Gerenciar camadas*	*Database related tools*
► *Add SAP HANA Spatial Layer…*		*Gerenciar camadas*	*Database related tools*
► *Adicionar/Editar camada virtual…*		*Gerenciar camadas*	*Criando camadas virtuais*
► *Add WMS/WMTS Layer…*	`Ctrl+Shift+W`	*Gerenciar camadas*	*Carregando as camadas WMS/WMTS*
► *Adicionar camada XYZ…*			*Using XYZ Tile services*
► *Add WCS Layer…*		*Gerenciar camadas*	*WCS Cliente*
► *Add WFS / OGC API - Features Layer…*		*Gerenciar camadas*	*WFS e WFS-T Cliente*
► *Add ArcGIS REST Server Layer…*		*Gerenciar camadas*	*Using ArcGIS REST Servers*
► *Add Vector Tile Layer…*			*Using Vector Tiles services*
► *Add Point Cloud Layer…*			*Working with Point Clouds*
► *Add GPX Layer…*			*Introducing GNSS/GPS Data*

continua na próxima página

Tabela 3.5 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Incorporar Camadas e Grupos…			*Incorporando camadas de projetos externos*
Add from Layer Definition File…			*Arquivo de definição de camada*
Georeferencer…			*Georreferenciador*
Copiar Estilo			*Salvar e Compartilhar Propriedades da Camada*
Colar estilo			*Salvar e Compartilhar Propriedades da Camada*
Copy Layer			
Paste Layer/Group			
Abrir Tabela de Atributos	`F6`	*Atributos*	*Trabalhando com a tabela de atributos*
Filtrar tabela de atributos ►			*Trabalhando com a tabela de atributos*
► *Tabela de Atributos Abertos (Características Selecionadas)*	`Shift+F6`	*Atributos*	*Trabalhando com a tabela de atributos*
► *Tabela de Atributos Abertos (Características Visíveis)*	`Ctrl+F6`	*Atributos*	*Trabalhando com a tabela de atributos*
► *Tabela de Atributos Abertos (Editados e Novas Características)*		*Atributos*	*Trabalhando com a tabela de atributos*
Alternar Edição		*Digitalização*	*Digitalizar uma camada existente*
Salvar Edições de Camada		*Digitalização*	*Salvando as Camadas Editadas*
Edições Atuais ►		*Digitalização*	*Salvando as Camadas Editadas*
► *Salvar para camada(s) selecionada(s)*		*Digitalização*	*Salvando as Camadas Editadas*
► *Rollback for Selected Layer(s)*		*Digitalização*	*Salvando as Camadas Editadas*
►*Cancelar para a(s) camada(s) selecionada(s)*		*Digitalização*	*Salvando as Camadas Editadas*
► *Salvar para todas as camadas*		*Digitalização*	*Salvando as Camadas Editadas*
► *Rollback for all Layers*		*Digitalização*	*Salvando as Camadas Editadas*
► *Cancelar para todas as camadas*		*Digitalização*	*Salvando as Camadas Editadas*
Salvar Como…			*Criando novas camadas a partir de uma camada existente*
Save As Layer Definition File…			*Arquivo de definição de camada*
Remove Layer/Group	`Ctrl+D`		
Duplicar Camada(s)			

continua na próxima página

Tabela 3.5 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Definir Escala de Visibilidade da(s) Camada(s)			*Visibility Scale Selector*
Set CRS of Layer(s)	`Ctrl+Shift+C`		*Sistemas de referência de coordenadas da camada*
Set Project CRS from Layer			*Sistemas referência de coordenadas projetadas*
Propriedades da Camada…			• *vector* • *raster* • *mesh* • *point cloud* • *vector tiles*
Filtrar…	`Ctrl+F`		*Criador de consultas*
Labeling			*Propriedades das etiquetas*
Mostrar em Visão Geral			*Painel geral*
Show All in Overview			*Painel geral*
Hide All from Overview			*Painel geral*

3.1.5 Configurações

Tabela3.6: The Settings menu items

Opção de menu	Referencia
Perfis de Usuário ►	*Trabalhando com Perfis de Usuário*
► *padrão*	*Trabalhando com Perfis de Usuário*
► *Open Active Profile Folder*	*Trabalhando com Perfis de Usuário*
► *Novo Perfil…*	*Trabalhando com Perfis de Usuário*
Gerenciar estilo…	*Gerenciador de Estilo*
Customizar Projeções…	*Sistema de Referência de Coordenadas personalizado*
Keyboard Shortcuts…	*Atalhos do Teclado*
Interface Customization…	*Personalização*
Opções…	*Opções*

Under Linux KDE, you'll find more tools in the *Settings* menu such as *Panels* ►, *Toolbars* ► and *Toggle Full Screen Mode*.

3.1.6 Complementos

Tabela3.7: The Plugins menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Manage and Install Plugins…			*Diálogo de Complementos*
Python Console	`Ctrl+Alt+P`	*Complementos*	*Terminal Python QGIS*

Ao iniciar o QGIS pela primeira vez, nem todos os complementos principais são carregados.

3.1.7 Vetor

The *Vector* menu provides tools for analyzing, managing and manipulating vector datasets. By default, it is filled with a set of predefined *Processing* algorithms, grouped by sub-menus. This provides shortcuts for many common vector-based GIS tasks from core providers. If not all these sub-menus are available, enable the *Processing* plugin in *Plugins* ► *Manage and Install Plugins…*.

Nota

The list of tools and their menu can be modified/extended with any Processing algorithms (read *Configuring the Processing Framework*) or by some third-party *plugins*.

This is what the *Vector* menu looks like if all the core plugins are enabled.

Tabela3.8: The Vector menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Check Geometries…			*Plugin do Verificador de Geometria*
Topology Checker		*Vetor*	*Complementos Verificador de Topologia*
Ferramentas de geoprocessamento ►	`Alt+O` + `G`		
► *Buffer…*			*Buffer*
► *Clip…*			*Clip*
► *Convex Hull…*			*Casco convexo.*
► *Difference…*			*Diferença*
► *Dissolve…*			*Dissolver*
► *Intersection…*			*Interseção*
► *Symmetrical Difference…*			*Diferença simétrica*
► *Union…*			*União*
► *Eliminate Selected Polygons…*			*Eliminar polígonos selecionados*
Geometrias ►	`Alt+O` + `E`		
► *Centroids…*			*Centroides*

continua na próxima página

Tabela 3.8 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
► *Collect Geometries…*			*Coletar geometrias*
► *Extract Vertices…*			*Extrair vértices*
► *Multipart to Singleparts…*			*Multipartes para partes únicas.*
► *Polygons to Lines…*			*Polígonos para linhas*
► *Simplify…*			*Simplificar*
► *Check Validity…*			*Verificar a validade*
► *Delaunay Triangulation…*			*Triangulação de Delaunay*
► *Densify by Count…*			*Densificar por contagem.*
► *Add Geometry Attributes…*			*Adicionar atributos da geometria*
► *Lines to Polygons…*			*Linhas para polígonos*
► *Voronoi Polygons…*			*Polígonos de Voronoi*
Ferramentas de análise ►	Alt+O + A		
► *Line Intersection…*			*Interseções de linha*
► *Mean Coordinate(s)…*			*Coordenada(s) média(s)*
► *Basic Statistics for Fields…*			*Estatísticas básicas para campos*
► *Count Points in Polygon…*			*Contar pontos em polígono*
► *Distance Matrix…*			*Matriz de distância*
► *List Unique Values…*			*Listar valores únicos*
► *Nearest Neighbour Analysis…*			*Análise do vizinho mais próximo*
► *Sum Line Lengths…*			*Soma dos comprimentos das linhas*
Data Management Tools ►	Alt+O + D		
► *Merge Vector Layers…*			*Mesclar camadas vetoriais*
► *Reproject Layer…*			*Reprojetar camada*
► *Create Spatial Index…*			*Criar índice espacial*
► *Join Attributes by Location…*			*Unir atributos por localização*
► *Split Vector Layer…*			*Dividir camada vetorial*
Ferramentas de pesquisa ►	Alt+O + R		
► *Create Grid…*			*Create grid*
► *Extract Layer Extent…*			*Extrair extensão da camada*
► *Random Points in Extent…*			*Random points in extent*
► *Random Points in Polygons…*			*Pontos aleatórios em polígonos*
► *Random Points on Lines…*			*Pontos aleatórios nas linhas*
► *Select by Location…*			*Selecionar pela localização*
► *Select Within Distance…*			*Selecionar dentro da distância*

continua na próxima página

Tabela 3.8 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
► *Random Points in Layer Bounds…*			*Random points in layer bounds*
► *Random Points Inside Polygons…*			*Pontos aleatórios dentro de polígonos*
► *Random Selection…*			*Seleção aleatória*
► *Random Selection Within Subsets…*			*Seleção aleatória dentro de subconjuntos*
► *Regular Points…*			*Pontos regulares*

3.1.8 Raster

The *Raster* menu provides tools for analyzing, managing and manipulating raster datasets. By default, it is filled with a set of predefined *Processing* algorithms, grouped by sub-menus. This provides shortcuts for many common raster-based GIS tasks from core providers. If not all these sub-menus are available, enable the *Processing* plugin in *Plugins* ► *Manage and Install Plugins…*.

Nota

The list of tools and their menu can be modified/extended with any Processing algorithms (read *Configuring the Processing Framework*) or by some third-party *plugins*.

This is what the *Raster* menu looks like if all the core plugins are enabled.

Tabela3.9: The Raster menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Raster calculator…			*Calculadora raster*
Align Rasters…			*Align rasters*
Analysis ►			
► *Aspect…*			*Aspecto*
► *Fill nodata…*			*Fill NoData*
► *Grid (Moving Average)…*			*Grid (média móvel)*
► *Grid (Data Metrics)…*			*Grid (Data metrics)*
► *Grid (Inverse Distance to a Power)…*			*Grid (Inverse distance to a power)*
► *Grid (Nearest Neighbor)…*			*Grid (IDW with nearest neighbor searching)*
► *Near Black…*			*Quase preto*
► *Hillshade…*			*Sombreamento*

continua na próxima página

Tabela 3.9 – continuação da página anterior

Opção de menu	Atalho	Barra de Ferramentas	Referencia
► *Proximity (Raster Distance)…*			*Proximidade (distância raster)*
► *Roughness…*			*Rugosidade*
► *Sieve…*			*Sieve*
► *Slope…*			*Inclinação*
► *Topographic Position Index (TPI)…*			*Topographic Position Index (TPI)*
► *Terrain Ruggedness Index (TRI)…*			*Terrain Ruggedness Index (TRI)*
Projeções ►			
► *Assign Projection…*			*Atribuir projeção*
► *Extract Projection…*			*Extrair projeção*
► *Warp (Reproject)…*			*Warp (reproject)*
Miscelânea ►			
► *Build Virtual Raster…*			*Construir raster virtual*
► *Raster Information…*			*Informação de raster*
► *Merge…*			*Mesclar*
► *Build Overviews (Pyramids)…*			*Construir Visão geral (Pirâmides)*
► *Tile Index…*			*Índice deslizante*
Extração ►			
► *Clip Raster by Extent…*			*Recortar raster extensão*
► *Clip Raster by Mask Layer…*			*Recortar raster por camada de máscara*
► *Contour…*			*contorno*
Conversão ►			
► *PCT to RGB…*			*PCT para RGB*
► *Polygonize (Raster to Vector)…*			*Poligonizar (raster para vetor)*
► *Rasterize (Vector to Raster)…*			*Rasterizar (vetor a rasterizar)*
► *RGB to PCT…*			*RGB para PCT*
► *Translate (Convert Format)…*			*Traduzir (converter formato)*

3.1.9 Base de dados

The *Database* menu provides tools for Database interaction. If no database-related plugins are enabled, there will be no *Database* menu. This is what the *Database* menu looks like if all the core plugins are enabled.

Tabela3.10: The Database menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Offline editing…	`Alt+D` + `O`		*Plugin de edição fora de rede*
► *Convert to Offline Project…*		*Database*	*Plugin de edição fora de rede*
► *Synchronize*		*Database*	*Plugin de edição fora de rede*
Query History…		*Database*	*Creating SQL Query Layers*
DB Manager…		*Database*	*Plugin do gerenciador de banco de dados*

Ao iniciar o QGIS pela primeira vez, nem todos os complementos principais são carregados.

3.1.10 Web

This is what the *Web* menu looks like if all the core plugins are enabled. If no web plugins are enabled, there will be no *Web* menu.

Tabela3.11: The Web menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
MetaSearch ►	`Alt+W` + `M`		*Cliente do Catálogo Metabusca*
► *Metasearch*		*Web*	*Cliente do Catálogo Metabusca*
► *Ajuda*			*Cliente do Catálogo Metabusca*

Ao iniciar o QGIS pela primeira vez, nem todos os complementos principais são carregados.

3.1.11 Malha

The *Mesh* menu provides tools needed to manipulate *mesh layers*. Third-party plugins can add items to this menu.

Tabela3.12: The Mesh menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Calculadora da Malha…			*Calculadora de Malha*
Reindex Faces and Vertices			*Reindexação de malhas*

3.1.12 Processamento

Tabela3.13: The Processing menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Toolbox	`Ctrl+Alt+T`		*A Barra de Ferramentas*
Model Designer…	`Ctrl+Alt+G`		*The model designer*
History…	`Ctrl+Alt+H`		*Gerenciador do histórico*
Visualizador de Resultados	`Ctrl+Alt+R`		*Configurando as aplicações externas*
Edit Features In-Place			*The Processing in-place layer modifier*

Ao iniciar o QGIS pela primeira vez, nem todos os complementos principais são carregados.

3.1.13 Ajuda

Tabela3.14: The Help menu items

Opção de menu	Atalho	Barra de Ferramentas	Referencia
Help Contents	`F1`	*Ajuda*	
API Documentation			
Complementos ►			
Report an Issue			
Need commercial support?			
QGIS Home Page	`Ctrl+H`		
Check QGIS Version			
About			
QGIS Membros Mantenedores			

3.1.14 QGIS

This menu is only available under macOS and contains some OS related commands.

Tabela3.15: The QGIS menu items

Opção de menu	Atalho	Referencia
Sobre o QGIS		
Preferências	`Cmd+,`	*Opções*
Ocultar QGIS	`Cmd+H`	
Mostrar Todas		
Hide Others	`Shift+Cmd+H`	
Sair do QGIS	`Cmd+Q`	

Preferences corresponds to *Settings* ► *Options*, *About QGIS* corresponds to *Help* ► *About* and *Quit QGIS* corresponds to *Project* ► *Exit QGIS* for other platforms.

3.2 Painéis e Barras de Ferramentas

From the *View* menu (or *Settings*), you can switch QGIS widgets (*Panels* ►) and toolbars (*Toolbars* ►) on and off. To (de)activate any of them, right-click the menu bar or toolbar and choose the item you want. Panels and toolbars can be moved and placed wherever you like within the QGIS interface. The list can also be extended with the activation of *Core or external plugins*.

3.2.1 Barra de ferramentas

The toolbars provide access to most of the functions in the menus, plus additional tools for interacting with the map. Each toolbar item has pop-up help available. Hover your mouse over the item and a short description of the tool's purpose will be displayed.

As barras de ferramentas disponíveis são:

Tabela3.16: QGIS Toolbars

Toolbar name	Referência principal para ferramentas
Advanced Digitizing	*Digitalização Avançada*
Annotations	*Ferramentas de anotação*
Atributos	*Trabalhando com a tabela de atributos*, *Ferramentas Gerais*
Data Source Manager	*Gerenciando fonte de dados*
Database	*Plugin do gerenciador de banco de dados*
Digitalização	*Digitalizar uma camada existente*
GPS	*Rastreio GPS em tempo real*
GRASS	*Integração com SIG GRASS*
Ajuda	
Rótulo	*A barra de ferramentas Rótulo*
Gerenciar camadas	*Abrir Dados*
Navegação de mapa	*Explorando a visualização do mapa*
Mesh Digitizing	*Editando uma camada de malha*
Complementos	*Complementos*
Projeto	*Trabalhando com Arquivos de Projetos*, *Laying out the maps*, *Biblioteca de estilo*
Processing Algorithms	*Configuring the Processing Framework*
Raster	*Complementos*
Seleção	*Selecionando características*
Shape digitizing	*Digitalização de forma*
Snapping	*Setting the snapping tolerance and search radius*
Vetor	*Complementos*
Web	*Complementos*, *Cliente do Catálogo Metabusca*

Nota

Third-party plugins can extend the default toolbar with their own tools or provide their own toolbar.

Dica

Restaurar barra de ferramentas

Se você ocultou acidentalmente uma barra de ferramentas, é possível recuperá-la usando: *Exibir* ► *Barras de Ferramentas* ► (ou *Configurações* ► *Barras de Ferramentas* ►). Se, por algum motivo, uma barra de ferramentas (ou qualquer outro widget) desaparecer totalmente da interface, você encontrará dicas para recuperá-la em *restoring initial GUI*.

3.2.2 Painéis

O QGIS fornece muitos painéis. Painéis são widgets especiais com os quais você pode interagir (selecionar opções, marcar caixas, preencher valores…) para executar tarefas mais complexas.

Abaixo está uma lista dos painéis padrão fornecidos pelo QGIS:

Tabela3.17: QGIS Panels

Panel name	Atalho	Referencia
Advanced Digitizing	`Ctrl+4`	*O painel Digitalização avançada*
Browser	`Ctrl+2`	*The Browser Panel*
Browser (2)		*The Browser Panel*
Debugging/Development Tools	`F12`	*Debugging/Development Tools Panel*
Elevation Profile		*Elevation Profile View*
Geometry Validation		*Propriedades de Digitalização*
GPS Information	`Ctrl+0`	*Rastreio GPS em tempo real*
GRASS Tools		*Integração com SIG GRASS*
Layer Order	`Ctrl+9`	*Layer Order Panel*
Layer Styling	`Ctrl+3`	*Painel de Estilo de Camada*
Layers	`Ctrl+1`	*Painel Camadas*
Log Messages		*Log Messages Panel*
Overview	`Ctrl+8`	*Painel geral*
Processing Toolbox		*A Barra de Ferramentas*
Results Viewer		*A Barra de Ferramentas*
Snapping and Digitizing Options		*Setting the snapping tolerance and search radius*
Spatial Bookmark Manager	`Ctrl+7`	*Bookmarking extents on the map*
Statistics	`Ctrl+6`	*Statistical Summary Panel*
Temporal Controller		*O painel de controle temporal*
Tile Scale		*Mosaicos*
Undo/Redo	`Ctrl+5`	*Undo/Redo Panel*
Vertex Editor		*O painel do editor de vértices*

3.3 Barra de Status

A barra de status fornece informações gerais sobre a visualização do mapa e ações processadas ou disponíveis, e oferece ferramentas para gerenciar a visualização do mapa.

3.3.1 Barra de localização

No lado esquerdo da barra de status, a barra de localização, um elemento de pesquisa rápida, ajuda você a encontrar e executar qualquer recurso ou opção no QGIS:

1. Clique no elemento de texto para ativar a barra de pesquisa do localizador ou pressione `Ctrl+K`.
2. Type a text associated with the item you are looking for (name, tag, keyword, …). By default, results are returned for the enabled locator filters, but you can limit the search to a certain scope by prefixing your text with the *locator filters* prefix, ie. typing `l cad` will return only the layers whose name contains `cad`.

   O filtro também pode ser selecionado com um clique duplo no menu que aparece ao acessar o elemento localizador.
3. Clicar em um resultado para executar a ação correspondente, dependendo do tipo de item.

Dica

Limit the lookup to particular field(s) of the active layer

By default, a search with the "active layer features" filter (`f`) runs through the whole attribute table of the layer. You can limit the search to a particular field using the `@` prefix. E.g., `f @name sal` or `@name sal` returns only the features whose "name" attribute contains 'sal'. Text autocompletion is active when writing and the suggestion can be applied using `Tab` key.

A more advanced control on the queried fields is possible from the layer *Fields* tab. Read *Propriedades dos campos* for details.

Searching is handled using threads, so that results always become available as quickly as possible, even if slow search filters are installed. They also appear as soon as they are encountered by a filter, which means that e.g. a file search filter will show results one by one as the file tree is scanned. This ensures that the UI is always responsive, even if a very slow search filter is present (e.g. one which uses an online service).

Nota

The Nominatim locator tool may behave differently (no autocompletion search, delay of fetching results, …) with respect to the OpenStreetMap Nominatim usage policy.

Dica

Acesso rápido às configurações do localizador

Click on the icon inside the locator widget on the status bar to display the list of filters you can use and a *Configure* entry that opens the *Locator* tab of the *Settings* ► *Options…* menu.

3.3.2 Reportando ações

Na área ao lado da barra de localização, um resumo das ações que você executou será mostrado quando necessário (como selecionar feições em uma camada, remover camada, distância e direção da panorâmica) ou uma longa descrição da ferramenta sobre a que você está passando (não disponível para todas as ferramentas).

No caso de operações demoradas, como coleta de estatísticas em camadas matricial, execução de algoritmos de processamento ou renderização de várias camadas na visualização do mapa, uma barra de progresso é exibida na barra de status.

3.3.3 Control the map canvas

A opção *Coordenada* mostra a posição atual do mouse, seguindo-o enquanto se move pela visualização do mapa. Você pode definir as unidades (e precisão) na guia *Projeto* ► *Propriedades…* ► *Geral*. Clique no pequeno botão à esquerda da caixa de texto para alternar entre a opção Coordenada e a opção *Extensão* que mostra as coordenadas dos cantos inferior-esquerdo e superior direito da visualização do mapa em unidades de mapa.

Next to the coordinate display you will find the *Scale* display. It shows the scale of the map view. There is a scale selector, which allows you to choose between *predefined and custom scales*.

On the right side of the scale display, press the button to lock the scale to use the magnifier to zoom in or out. The magnifier allows you to zoom in to a map without altering the map scale, making it easier to tweak the positions of labels and symbols accurately. The magnification level is expressed as a percentage. If the *Magnifier* has a level of 100%, then the current map is not magnified, i.e. is rendered at accurate scale relative to the monitor's resolution

(DPI). A default magnification value can be defined within *Settings* ► *Options* ► *Rendering* ► *Rendering Behavior*, which is very useful for high-resolution screens to enlarge small symbols. In addition, a setting in *Settings* ► *Options* ► *Canvas & Legend* ► *DPI* controls whether QGIS respects each monitor's physical DPI or uses the overall system logical DPI.

À direita da ferramenta de ampliação, você pode definir uma rotação atual no sentido horário para a visualização do mapa em graus.

On the right side of the status bar, the *Render* checkbox can be used to temporarily suspend the map view rendering (see section *Controlling map rendering*).

To the right of the *Render* function, you find the *EPSG:code* button showing the current project CRS. Clicking on this opens the *Project Properties* dialog and lets you reproject the map view or adjust any other project property.

Dica

Calculando a escala correta do seu Mapa da tela/visualização

When you start QGIS, the default CRS is `WGS 84 (EPSG 4326)` and units are degrees. This means that QGIS will interpret any coordinate in your layer as specified in degrees. To get correct scale values, you can either manually change this setting in the *General* tab under *Project* ► *Properties...* (e.g. to meters), or you can use the EPSG:code icon seen above. In the latter case, the units are set to what the project projection specifies (e.g., `+units=us-ft`).

Note that CRS choice on startup can be set in *Settings* ► *Options* ► *CRS Handling*.

3.3.4 Mensagens

The Messages button next to it opens the *Log Messages Panel* which has information on underlying processes (QGIS startup, plugins loading, processing tools…)

CAPÍTULO 4

Painel do Navegador

- *Resources that can be opened / run from the Browser*
- *Browser panel top-level entries*
  - *Favoritos*
  - *Favoritos Espaciais*
  - *Project Home*
  - *Drives and file system*
  - *Data type*
  - *Entradas no banco de dados*
  - *Mosaicos e Serviços da Web*
  - *Project and supported file types*

O painel do Navegador QGIS é uma ótima ferramenta para navegar, pesquisar, inspecionar, copiar e carregar recursos QGIS. Apenas os recursos que o QGIS sabe como lidar são mostrados no navegador.

Using the Browser panel you can locate, inspect and add data, as described in *The Browser Panel*. In addition, the Browser panel supports drag and drop of many QGIS resources, such as project files, Python scripts, Processing scripts and Processing models.

Scripts Python, scripts de processamento e modelos de processamento também podem ser abertos para edição em um editor externo e no modelador gráfico.

You can drag and drop layers from the *Layers* panel to the *Browser* panel, for instance into a GeoPackage or a PostgreSQL database.

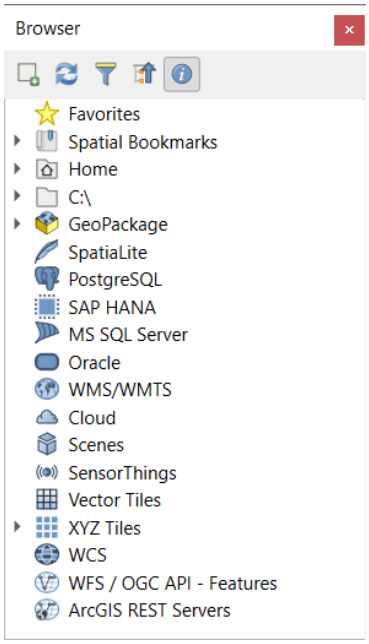


Figura4.1: Painel do Navegador

The browser panel (Fig.4.1) is organised as an expandable hierarchy with some fixed top-level entries that organise the resources handled by the browser. Node entries are expanded by clicking on to the left of the entry name. A branch is collapsed by clicking on . The Collapse All button collapses all top-level entries.

In *Settings* ► *Interface Customization* it is possible to disable resources. If you, for instance, would not like to show Python scripts in the browser, you can uncheck the *Browser* ► *py* entry, and if you want to get rid of your home folder in the browser, you can uncheck the *Browser* ► *special:Home* entry.

A filter (Filter Browser) can be used for searching based on entry names (both leaf entries and node entries in the hierarchy). Using the Options pull-down menu next to the filter text field, you can

- toggle *Case Sensitive* search
- set the *Filter pattern syntax* to one of
  - *Normal*
  - *Wildcard(s)*
  - *Regular Expressions*

The *Properties widget*, showing useful information about some entries / resources, can be enabled / disabled using the Enable/disable properties widget button. When enabled, it opens at the bottom of the browser panel, as shown in Fig.4.2.

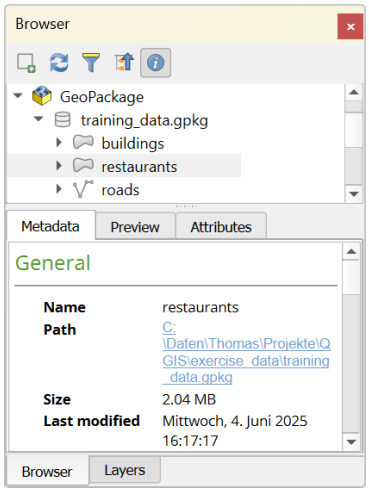


Figura4.2: The properties widget

A second browser panel can be opened by activating the *Browser (2)* panel in *View* ► *Panels*. Having two browser panels can be useful when copying layers between resources that are locationed deep down in different branches of the browser hierarchy.

4.1 Resources that can be opened / run from the Browser

A lot can be accomplished in the Browser panel

- Add vector, raster and mesh layers to your map by double-clicking, dragging onto the map canvas or clicking the Add Selected Layers button (after selecting layers)
- Run Python scripts (including Processing algorithms) by double-clicking or dragging onto the map canvas
- Run models by double-clicking or dragging onto the map canvas
- *Extract Symbols…* from QGIS Project files using the context menu
- Open files with their default applications (*Open <file type> Externally…* in the context menu). Examples: HTML files, spreadsheets, images, PDFs, text files, …
- Copy entries
- Rename and delete (multiple) layers (context menu: *Manage* ►)
- Abra uma janela do explorador de arquivos e selecione diretamente o arquivo *Mostrar em arquivos*.

Resource specific actions are listed for the different resource groups sorted under the top-level entries listed below.

4.2 Browser panel top-level entries

4.2.1 Favoritos

Often used file system locations can be tagged as favorites. The ones you have tagged will appear here.

In addition to the operations described under *Home*, the context menu allows you to *Rename Favorite…* and *Remove Favorite*.

4.2.2 Favoritos Espaciais

This is where you will find your *spatial bookmarks*, divided into *Project Bookmarks* and *User Bookmarks*.

Tabela4.1: available actions for spatial bookmarks

Nível	Contextual menu
Spatial Bookmarks	*New Spatial Bookmark…*
	Show the Spatial Bookmark Manager
	Import Spatial Bookmarks…
	Export Spatial Bookmarks…
Bookmark group (*Project Bookmarks* and *User Bookmarks*)	*New Spatial Bookmark…*
	Import Spatial Bookmarks…
	Export Spatial Bookmarks…
Individual bookmark	*Zoom to Bookmark*
	Edit Spatial Bookmark…
	Delete Spatial Bookmark

4.2.3 Project Home

Available once the project file has been saved, the *Project home* entry is a folder containing data and other contents (scripts, models, text, …) that may be used within the current project. Displayed in the *Browser* panel, it allows you to quickly access data and other files of the project.

It defaults to the project file folder but can be changed through the *Project* ► *Properties…* ► *General* ► *Project home* option, or by right-clicking on the *Project Home* item of the Browser panel and selecting *Set project home…*. Customizing that folder is especially useful in contexts where QGIS projects are not stored in the root folder of an organisational 'project', along with datasets.

4.2.4 Drives and file system

The next items of the *Browser* panel depend on the OS in use and concern the top level entries of its file system.

São principalmente:

- The *Home* folder, pointing to the current user home folder
- on Unix-based machines, the root `/` folder
- the connected drives, either local or network. Depending on the OS, they are directly listed (eg, `C:\`, `D:\`) or through the `/Volumes` entry.

From the contextual menu of each of these folders or drives, you can:

- atualizar conteúdo
- create a *New* ► subitem that is a *Directory*, *GeoPackage* or ESRI *Shapefile* format dataset
- hide the directory (*Hide from Browser*)
- *Set color*: customize the folder icon color, aiding in rapid browser navigation of complex folder structures
- enable *Scanning*:
  - *Monitor for changes*: allows to manually control whether a particular directory should be monitored and automatically updated. This setting applies to the selected directory and all subdirectories. This means that you can manually opt-in to monitoring of network drives if you know there's no issue, or manually opt-out of monitoring of large directories which you don't want monitored for other reasons. By default, remote or network drives are not automatically monitored.
  - *Fast scan this directory*

- abra o diretório em seu gerenciador de arquivos (*Abrir Diretório…*)
- abrir o diretório em uma janela terminal (*Abrir em Terminal…*)
- inspect the *Properties…* or the parent *Directory Properties…*

4.2.5 Data type

Tabela4.2: Available actions for plain layer files

Level and Context menu	Types of layers Raster (*Reference*)	Vector (*Reference*)	Point clouds (*Reference*)	Mesh (*Reference*)
File				
Delete File <dataset name>	☑	☑		
Mostrar em arquivos	☑	☑	☑	☑
Manage	☑	☑	☑	☑
File Properties…	☑	☑	☑	☑
Open <file type> Externally…	☑			
Layer				
Adicionar camada ao projeto	☑	☑	☑	☑
Export Layer	☑	☑		
Layer Properties…	☑	☑	☑	☑
Open with Data Source Manager…	☑	☑		
Execute SQL…		☑		

4.2.6 Entradas no banco de dados

Depending on your OS and installed drivers, you might have access to different database types to use in QGIS. Below are listed the different entries of contextual menu at each level of the dataset tree.

Tabela4.3: Available actions for database entries

Level and Context menu	Tipo de banco de dados GeoPackage ([1])	SpatiaLite	PostgreSQL	SAP HANA	MS SQL Server	Oracle
Top menu						
Create a *New Connection…* to an existing database	☑	☑	☑	☑	☑	☑
Create Database…	☑	☑				
Create Database and Layer…	☑					
Save Connections… details to a file			☑	☑	☑	
Load Connections…			☑	☑	☑	

continua na próxima página

Tabela 4.3 – continuação da página anterior

Connection / Database						
Refresh a connection			☑	☑	☑	☑
Edit Connection… settings			☑	☑	☑	☑
Duplicate Connection			☑	☑	☑	☑
Remove Connection…	☑	☑	☑	☑	☑	☑
Import Vector Layer…	☑	☑	☑	☑	☑	
New Field Domain ► New Range Domain	☑					
New Field Domain ► New Coded Values Domain	☑					
New Field Domain ► New Glob Domain	☑					
Delete <database_name>	☑	☑				
Compact Database (VACUUM)	☑					
Create a *New Schema…*			☑	☑	☑	
Create a *New Table…*	☑	☑	☑	☑		
Execute SQL… query	☑	☑	☑	☑		
Schema						
Refresh a schema			☑	☑	☑	
Schema Operations ► Rename Schema…			☑	☑	☑	
Schema Operations ► Delete Schema…			☑	☑	☑	
Create a *New Table…*			☑	☑		
Execute SQL… query			☑	☑		
Table / Layer						
Table Operations ► Rename Table…			☑	☑	☑	
Table Operations ► Truncate Table…			☑		☑	
Execute SQL… query	☑	☑	☑			
Export Layer ► To file…	☑	☑	☑	☑	☑	
Manage ► Rename Layer <layer_name>…	☑	☑			☑	
Manage ► Delete Layer <layer_name>…	☑	☑	☑	☑	☑	
Manage ► Delete Selected Layers	☑	☑	☑	☑	☑	
Manage ► Add Layer to Project	☑	☑	☑	☑	☑	
Manage ► Add Selected Layers to Project	☑	☑	☑	☑	☑	
Manage ► Move to Another Schema			☑		☑	

continua na próxima página

Tabela 4.3 – continuação da página anterior

Open *Layer Properties…* dialog	✓	✓	✓	✓	✓
Open *File Properties…* dialog	✓				
Open with Data Source Manager	✓	✓			
Fields					
Add New Field…	✓	✓	✓	✓	
Field					
Set Field Domain	✓				
Rename Field ([2])	✓		✓		✓
Set Alias…	✓				
Set Comment…	✓		✓		
Delete Field…	✓	✓	✓	✓	

Nota

When loading vector layers into QGIS, fields with *Field Domains* (such as those defined in a GeoPackage or ESRI File Geodatabase) are automatically detected. These domains are database-level constraints, meaning they are enforced by the database itself and apply across different applications, not just QGIS.

Importing vector layers into databases using Browser panel

QGIS allows you to import vector layers directly into a database from the *Browser* panel. This provides a simple way to transfer datasets into supported database formats without using DB Manager (see *Plugin do gerenciador de banco de dados*).

You can start the import in two ways:

- Drag and drop a single vector layer from the Browser panel or from the Project into a database connection or schema item.
- Right-click on a database connection or schema item and choose *Import Vector Layer…* from the context menu.

Nota

When multiple layers are dragged into a database, they are imported immediately without further options. All imports run in the background, so the QGIS interface remains responsive during the process.

[1] The different entries may also be available for some GDAL supported vector file formats such as ESRI File Geodatabase, FlatGeobuf, GeoParquet, NetCDF, … when compatible.
[2] For PostgreSQL connections: only supported with ALTER TABLE privilege.

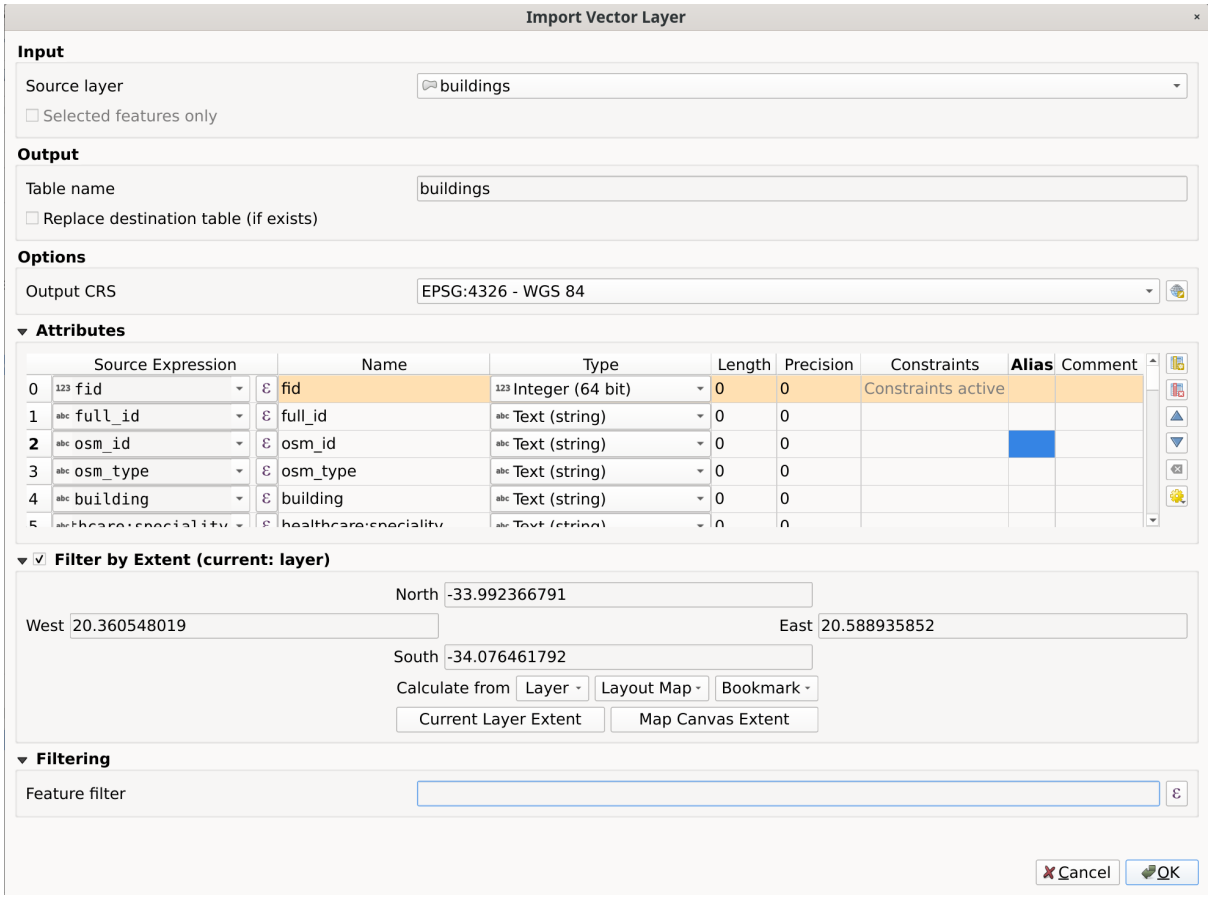


Figura4.3: Import Vector Layer dialog

To add a layer using *Import Vector Layer…* follow these steps:

1. Add *Source layer*: the layer to import, chosen from the current QGIS project. To import only selected features of the source layer use the *Selected features only* option.
2. Set the *Table name*: the name of the destination table in the database. By default, it is taken from the source layer name but can be modified. You can also use the *Replace destination table (if exists)* to overwrite an existing table.
3. Define the *Output CRS*: the coordinate reference system of the destination table. By default, it is the CRS of the source layer, but it can be changed using the *CRS selector*.
4. Configure the *Attributes*:
   - *Source expression*: expression used to populate the field. By default, this corresponds to the original source field.
   - *Name*: name of the field in the destination table.
   - *Type*: field data type (integer, text, real, etc.).
   - *Length* and *Precision*: length and precision of the field, where applicable.
   - *Constraints*: rules to enforce on the field values (e.g. NOT NULL, UNIQUE, DEFAULT). Support depends on the destination database.
   - *Alias*: an alternate display name for the field.
   - *Comment*: a description or note attached to the field.

   You can use buttons on the right side to:

- Add new field to add a new field to the destination table.
- Delete selected field to remove a field from the destination table.
- Move selected field up or Move selected field down to reorder fields.
- Reset all fields to clear all fields and start from scratch.
- Convert All Fields To Lowercase or Convert All Fields To Uppercase, to change the case of all field names.

5. Optionally, apply a spatial filter with *Filter by Extent*. This restricts the import to features within a defined bounding box.
6. Optionally, apply an attribute filter with *Feature filter*. This allows you to use an expression to select which features are imported.

4.2.7 Mosaicos e Serviços da Web

Tabela4.4: Available actions for web services (1)

Level and Context menu	Tipo de serviços				
	WMS / WMTS (*Reference*)	XYZ Tiles (*Reference*)	WCS (*Reference*)	WFS / OGC API - Features (*Reference*)	ArcGIS REST Servers (*Reference*)
Top menu					
Create a *New Connection…*	☑	☑	☑	☑	☑
Save Connections… details to a file	☑	☑	☑	☑	☑
Load Connections…	☑	☑	☑	☑	☑
Connection					
Refresh connection	☑	☑	☑	☑	☑
Edit Connection… settings	☑	☑	☑	☑	☑
Duplicate Connection	☑	☑	☑	☑	☑
Remove Connection…	☑	☑	☑	☑	☑
View Service Info in Web browser					☑
Table / Layer					
Export Layer ► To File…	☑	☑	☑	☑	☑
Adicionar camada ao projeto	☑	☑	☑	☑	☑
Add Filtered Layer to Project…					☑
Open *Layer properties…* dialog	☑	☑	☑	☑	☑
View Service Info in Web browser					☑

Tabela4.5: Available actions for web services (2)

Level and Context menu	Tipo de serviços				
	Vector Tiles (*Reference*)	Scene (*Reference*)	Cloud (*Reference*)	Sensor-Things (*Reference*)	STAC (*Reference*)
Top menu					
Create a *New Connection…*		☑	☑	☑	☑
Create a *New Generic Connection…*	☑				
Create a *New ArcGIS Vector Tile Service Connection…*	☑				
Save Connections… details to a file	☑	☑	☑	☑	☑
Load Connections…	☑	☑	☑	☑	☑
Connection					
Edit Connection… settings	☑	☑	☑	☑	☑
Duplicate Connection	☑	☑	☑	☑	
Delete Connection…	☑	☑	☑	☑	☑
Refresh					☑
Details…					☑
Table / Layer					
Export Layer ► To File…				☑	
Adicionar camada ao projeto	☑	☑		☑	
Open *Layer properties…* dialog	☑	☑		☑	
Refresh					☑
Download assets…					☑
Details…					☑

4.2.8 Project and supported file types

Tabela4.6: Available actions for various supported files :class: longtable

File type	Context menu
QGIS Project file (.qgz, .qgs)	*Open Project*
	Extract Symbols… (opens Style Manager to export symbols to XML, add to default style, or export as PNG/SVG)
	File Properties…
QGIS Layer Definition (.qlr)	*Export Layer ► To File…*
	Add Layer to Project
	Layer Properties…
Processing Model (.model3)	*Run Model…*
	Edit Model…
Print Composer Template (.qpt)	*New Layout from Template*
Python script (.py)	*Run Script…*
	Open in External Editor

CAPÍTULO 5

Configuração QGIS

O QGIS é altamente configurável. Através do menu *Configurações*, fornece ferramentas diferentes para:

- *Gerenciador de Estilos…*: cria e gerencia *símbolos, estilos e rampas de cores*.
- *Custom Projections…*: crie sua própria:ref:*coordinate reference systems <sec_custom_projections>*.
- *Keyboard Shortcuts…*: defina suas próprias configurações *keyboard shortcuts*. Para além disso, eles podem ser substituídos durante cada sessão do QGIS por *project properties* (accessible under *Project* menu).
- *Interface Customization…*: configure o:ref:*application interface <sec_customization>*, hiding dialogs ou ferramentas que talvez você não precise.
- *Options…*: configuração global *options* Para aplicar em diferentes áreas do software. Essas preferencias são salvas nas configurações *User profile* e aplicadas por padrão quando você abre um novo projeto com esse perfil.

5.1 Opções

Algumas opções básicas para o QGIS podem ser selcionadas usando a caixa de diálogo *Options*. Selecione a opção menu *Settings* ► *Options*. Você pode modificar as opções de acordo com suas necessidades. Algumas das alterações talvez requeiram reiniciar o QGIS antes delas terem efeito.

As abas onde você pode personalizar suas opções estão descritas abaixo.

Nota

Plugins podem embutir suas características dentro da caixa de diálogo das opções

Enquanto as configurações principais são apresentadas abaixo, note que essa lista pode ser estendida pelo *installed plugins* Implementando suas próprias opções na caixa de diálogo de opções padrões. Isso evita que cada plugin tenha sua própria caixa de diálogo de configuração com itens extra de menu somente para eles…

5.1.1 Configurações Gerais

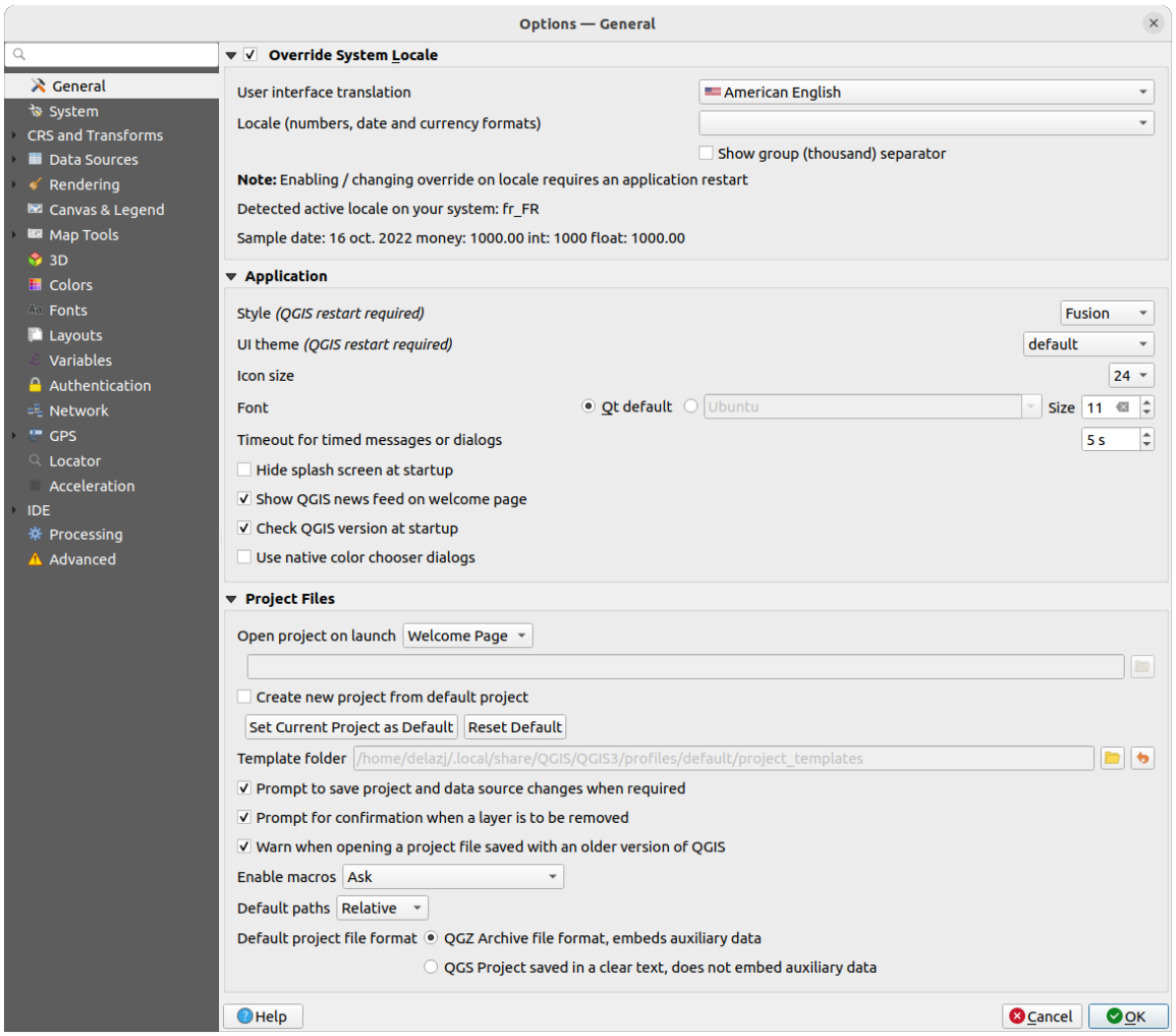


Figura5.1: Configurações Gerais

Sobrepõe o System Locale

Por padrão, o QGIS depende da configuração do seu sistema operacional para configurar a linguagem e manipular valores numéricos. Ativar esse grupo permite que você customize o comportamento.

- Selecione do *User interface translation* a linguagem que será aplicada no GUI.
- Selecione em *Locale (number, date and currency formats)* o sistema no qual a data e valores numéricos devem ser inseridos e renderizados.
- *Mostrar separador de grupo (mil)*

Um sumário das configurações selecionadas e como elas devem ser interpretadas é mostrado na parte inferior do quadro

Aplicação

- Selecione o *Style (QGIS restart required)* leia-se, a aparência e posição nas caixas de diálogo. Valores possíveis dependem do seu sistema operacional.
- Defina a *UI theme (QGIS restart required)* , pode ser 'padrão', 'Mapeamento noturno' ou 'Mistura de cinza'

- Defina a *Icon size*
- Defina a *Font* e seu *Size*. A fonte pode ser *Qt padrão* ou a definida por usuário.
- Mude a *Timeout for timed messages or dialogs*
- *Hide splash screen at startup*
- *Show QGIS news feed on welcome page*: Exibe um feed de curadoria de noticiais sobre o QGIS na página de boas vindas, dando a você uma via direta para saber notícias sobre o projeto (data de reuniões com usuários/desenvolvedores e sumários, pesquisas da comunidade, anúncios de lançamentos, dicas variadas…)
- *Check QGIS version at startup* para te manter informado se uma nova versão foi lançada
- *Use native color chooser dialogs* (ver *Seletor de Cor*)

Arquivos de projeto

- *Abrir Projeto ao iniciar*
  - 'Página de boas vindas' (padrão): pode mostrar o feed "Noticias", Modelos do projeto e os projetos mais recentes (com miniaturas) do *user profile*. Por padrão, nenhum projeto é aberto.
  - 'Novo': abre um novo projeto, com base no modelo padrão
  - 'Mais recente': reabre o último projeto salvo
  - e 'Específico': abre um projeto em particular. Use o button … para definir o projeto para usar por padrão.
- *Criar um novo projeto como projeto padrão*. Tem a possibilidade de carregar em *Usar projeto atual com padrão* ou em *Resetar padrão*. Pode navegar através dos seus arquivos e definir um diretório onde encontra os modelos de projeto definidos pelo usuário. Isto será adicionado a *Projeto* ► *Novo do modelo*. Primeiro ative *Criar novo projeto como projeto padrão* e a seguir salve o projeto dentro da pasta de modelos de projeto.
- *Prompt para salvar as mudanças de fonte do projeto e data quando requerido* para evitar perder mudanças que você fez.
- *Solicitar confirmação quando uma camada for ser removida*
- *Avisa quando abrir um projeto salvo com uma versão antiga do QGIS*. Você pode sempre abri projetos criados com uma versão mais velha do QGIS, mas uma vez que o projeto é salvo, tentar abrir com uma versão antiga talvez não dê certo, pois os recursos não estarão disponíveis naquela versão.
- *Enable project's embedded Python code* . This option handles execution of macros that are written to perform an action on project events, as well as custom Python functions to be used as expressions. You can choose between 'Never', 'Ask', 'For this session only' and 'Always (not recommended)'.
- *Default paths*: Define se os caminhos para os arquivos e camadas usados em um novo projeto são salvos como 'Absoluto' ou 'Relativo' para um arquivo de projeto. Essa configuração sobre sobrescrito no nível do projeto.
- :guilabel:`Formato de arquivo de projeto padrão'.
  - *Formato de arquivo QGZ, encorpra dados auxilíares* (see *data auxiliar*)
  - *Projeto QGS salvo em um texto limpo*: a data auxiliar é guardada em um separado `Arquivo .qgd` junto com o projeto do arquivo.

5.1.2 Configurações de sistema

SVG paths

Adiciona ou Remove *Caminho(s) para pesquisar os sÍmbolos da Escala Gráfica Vetorial (EGV)*. Esses arquivos EGV são então disponibilizados para simbolizar ou rotular os atributos ou decorar a composição do seu mapa.

Também lê incorporado_arquivo_seletor para ter modos diferentes para referenciar os arquivos EGV em caminho QGIS.

Diretórios dos complementos

Adicionar ou remover :guilabel:`Caminho(s) para procurar por bibliotecas de plugins C++ adicionais'.

Caminhos da documentação

Add or Remove *Documentation Path(s)* to use for QGIS help. By default, a link to the official online User Manual corresponding to the version being used is added (i.e., `https://docs.qgis.org/§qgis_short_version/$qgis_locale/docs/user_manual/`). You can however add other links and prioritize them from top to bottom: each time you click on a *Help* button in a dialog, the topmost link is checked and if no corresponding page is found, the next one is tried, and so on.

> **Nota**
>
> A documentação é versionada e traduzida somente para o Termo de Lançamentos Longo (TLL) do QGIS, o que significa se você está executando um lançamento regular (por exemplo, QGIS 3.0) o botão de ajuda será por padrão aperto para a próxima página do manual TLL (3.2 e 3.4). Se nenhuma documentação TLL estiver disponível, então o documento *testando*, com os recursos das versões novas e em desenvolvimento é usada.

Configurações

Te ajuda *Reiniciar a interface do usuário para as configurações padrão (Reinicio requerido)* Se você fizer qualquer *Customização*.

Ambiente

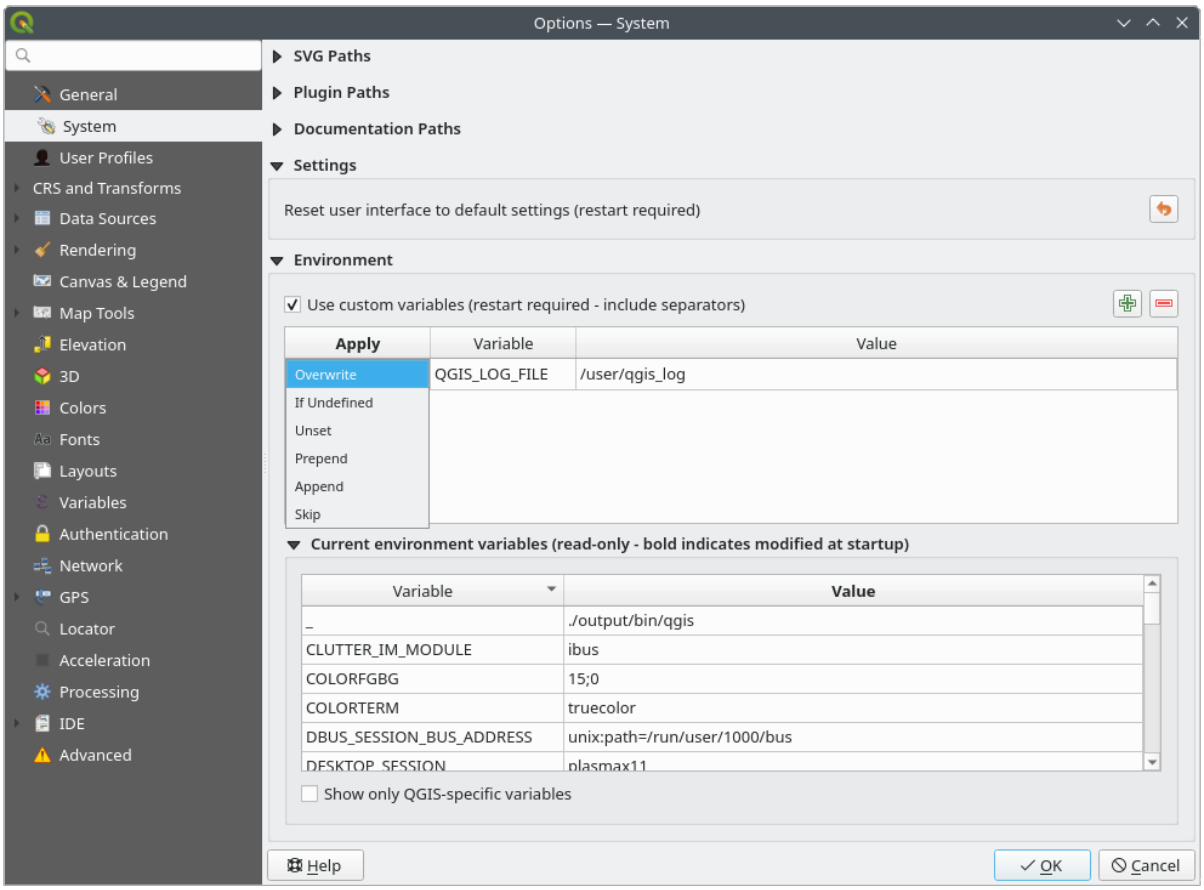


Figura5.2: Variáveis de ambiente do sistema

System environment variables can be viewed, and many configured, in the **Environment** group. This is useful for platforms, such as Mac, where a GUI application does not necessarily inherit the user's shell environment. It's also useful for setting and viewing environment variables for the external tool sets controlled by the Processing toolbox (e.g., GRASS), and for turning on debugging output for specific sections of the source code.

Marque *Use variáveis customizadas (reiniciar requerido - incluindo separadores)* e você pode Adicionar e Remova variáveis de ambiente. Para cada novo item, você pode configurar a :guilabel:nome *Variável*, é *Valor* e o :guilabel:Método *Aplicar* para usar, dentro dos quais:

- *Sobrescrever*: substitui qualquer valor pré-existente de variável
- *If undefined*: use this value for the variable if not already defined at a higher level (e.g., OS or application levels)
- *Desfazer*: remove a variável do ambiente (o parâmetro *Valor* não é usado)
- *Anteceder*: antecede o valor para o valor pré-existente da variável.
- *Acrescentar*: acrescenta o valor para o valor pré-existente da variável.
- *Passe*: o item é mantido na lista para futura referencia mas não é usado

Variáveis de ambiente já definidas são mostradas em *Variáveis de ambiente atuais*, e é possível filtra-lás ativando *Mostra somente variáveis específicas do QGIS*.

5.1.3 User Profiles Settings

> **Nota**
>
> For more information on how to manage user profiles, please read the dedicated section at *Trabalhando com Perfis de Usuário*.

5.1.4 Configurações de SRC transformações

> **Nota**
>
> Para mais informações de como o QGIS lida com projeção de camada, por favor leia a seção dedicada no rótulo_projeções.

Manuseio do SRC

Na aba *Manuseio CRS* você pode configurar qual SRC será usado para um novo projeto ou camada.

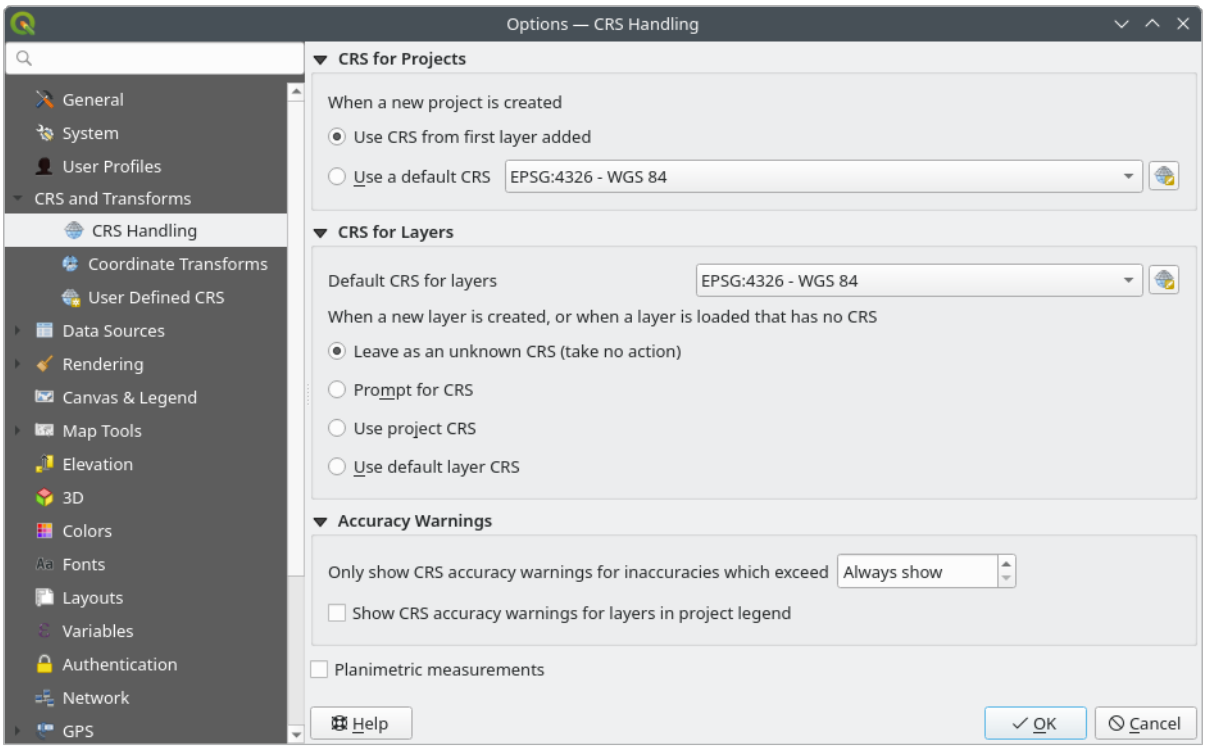


Figura5.3: Configurações SRC

SRC para projetos

Existe uma opção para definir automaticamente o SRC do novo projeto:

- *Usa SRC para a primeira camada adicionada*: o CRS do projeto será configurado para o SRC da primeira lista carregada nele.
- *Usa um SRC padrão*: um SRC pré-selecionado é aplicado por padrão para qualquer novo projeto e deixado imutável quando adicionando camadas para o projeto.

A escolha será salva para uso em sessão subsequentes do QGIS. O sistema de referência de coordenada do projeto pode ser sobrescrito por meio da aba *Project* ► *Properties…* ► *SRC*

SRC para camadas

Default SRC for layers: selecione um SRC padrão para usar quando você cria uma camada.

Você também pode definir a ação que desejar quando uma nova camada é criada, ou quando a camada sem um SRC é carregada

- *Deixe como SRC desconhecido (Não tome ação)*
- *Prompt para SRC*
- *Usar SRC do projeto*
- *Usar camada SRC padrão*

Avisos de acurácia

Only show CRS accuracy warnings for inaccuracies which exceed a given distance: occurs when you are explicitly creating or modifying a dataset and select a CRS based on a datum ensemble with lower accuracy. The default is to `Always show` the warning if any inaccuracy.

Show CRS accuracy warning for layers in project legend: If checked, any layer with a CRS with accuracy issues (i.e., a dynamic crs with no coordinate epoch available, or a CRS based on a datum ensemble with inherent inaccuracy exceeding the user-set limit) will have the warning icon in the *Layers* panel reflecting that it is a low-accuracy layer.

Isso é desenhado para uso na engenharia, BIM, gestão de ativos e outros campos onde inexatidões de níveis de metro/sub-metro são potencialmente muito perigosas ou caras!

Planimetric measurements: sets the default for the *planimetric measurements* property for newly created projects.

Transformações de Coordenadas

A aba *Transformações de Coordenadas* te ajuda a configurar trannsformações de coordenadas e operações para aplicar quando carregando uma camada para um projeto ou reprojetando uma camada.

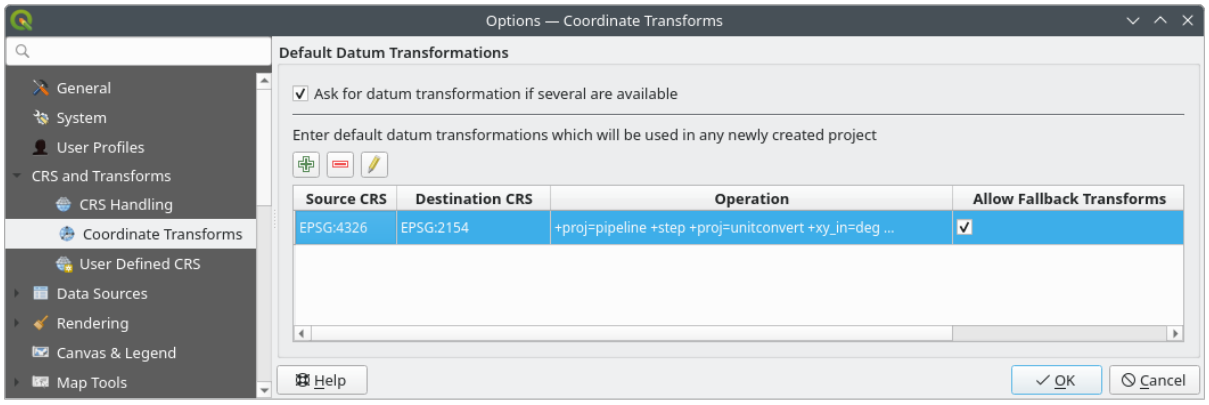


Figura5.4: Configurações de transformações

Datum padrão de transformações

Aqui você pode controlar se camadas de reprojeção para outro SRC deveria ser:

- processado automaticamente usando as configurações de transformações padrão do QGIS;

- E/ou mais controlado por você com preferências de customização como:
  - *Peça por transformação de datum se vários estão disponíveis*
  - Uma lista pré-definida de transformações de datam para aplicar por padrão. Veja Datum_transformação para mais detalhes.

Você pode Adicionar, Remover ou Editar transformações, que será usado em qualquer novo projeto criado.

SRC definido pelo usuário

A aba *SRC definido pelo usuário* te ajuda a definir um SRC customizado, cujo o qual deve estar de conformo com o WKT (texto conhecido) ou Proj em formado de string.

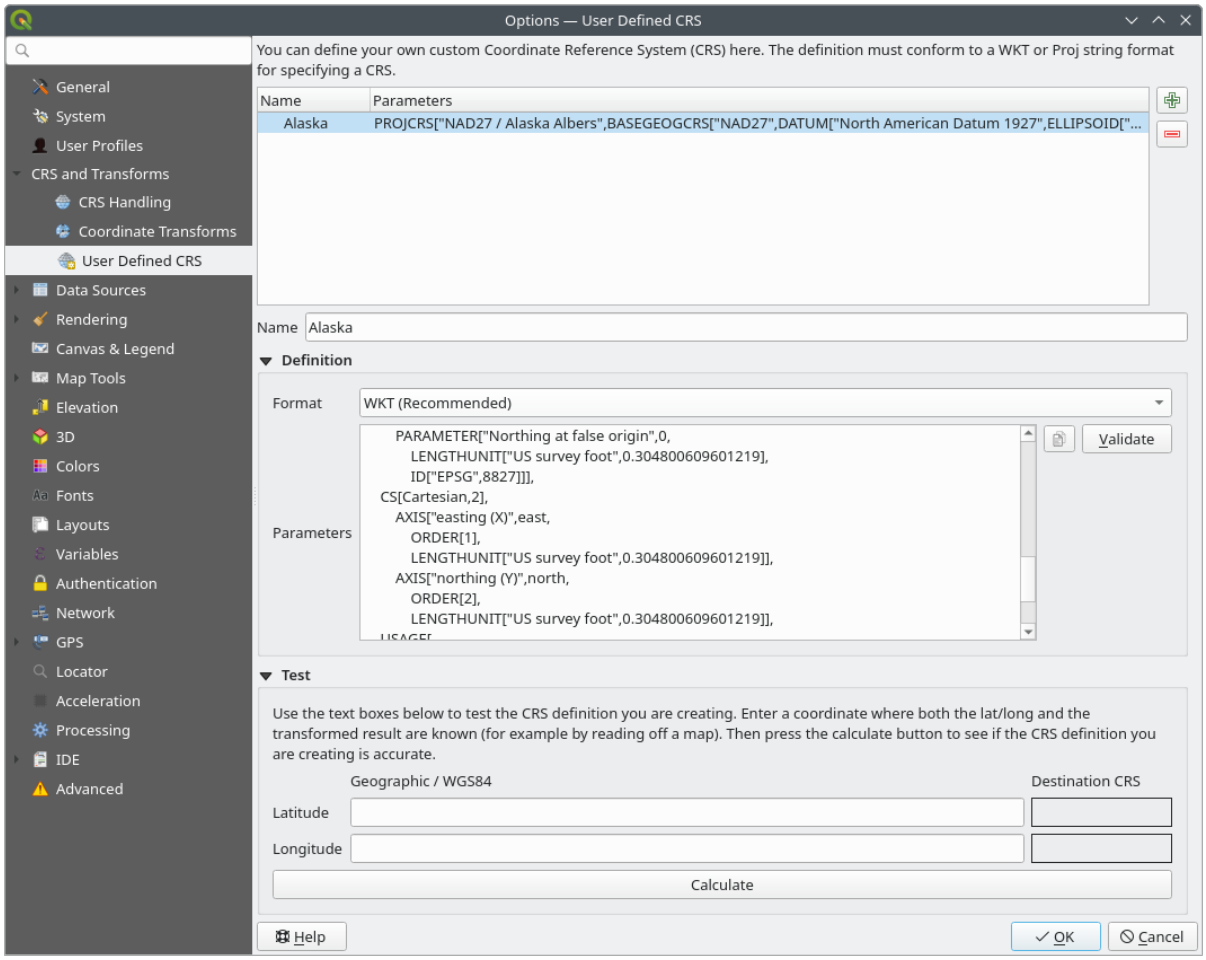


Figura5.5: SRC definido pelo usuário

Set a *Nome* e use Adicionar novo SRC. Se você quer deletar qualquer um já existente você poed usara Remover SRC.

Definição

- ***Formato***
  - WKT (Recomendado)
  - Linha proj (Legado - não recomendado)
- ***Parametros***

- Copia parâmetros de uma SRC existente.
- *Validar* testa se sua expressão está correta.

Teste

Aqui você consegue testar a definição para a SRC por latitude e longitude. Use uma coordenada conhecida para controlar se sua definição está precisa.

5.1.5 Configurações de fonte de informação

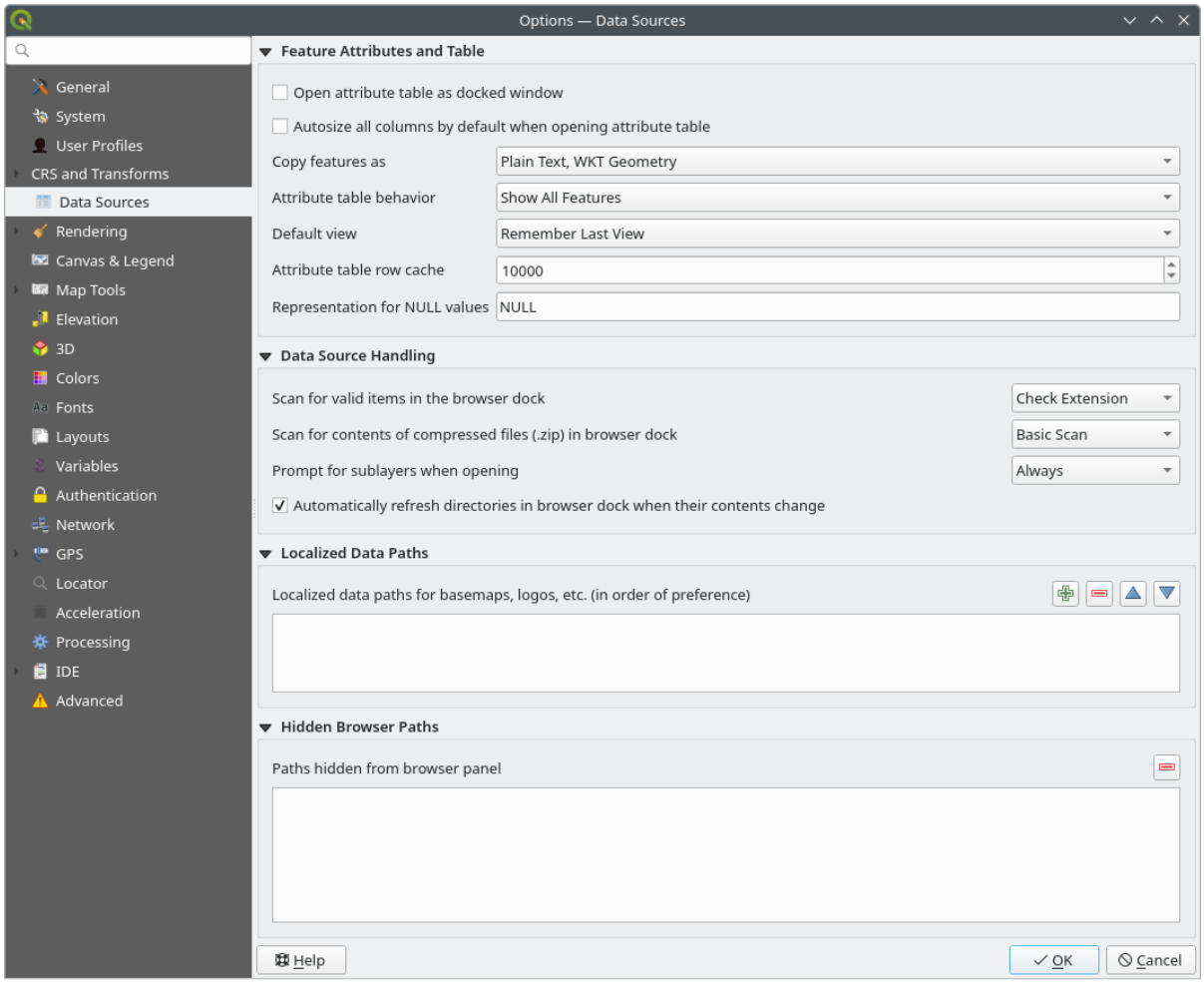


Figura5.6: Configurações de fonte de informação

Atributos dos elementos e tabela

- *Abrir tabela de atributos como uma janela encaixada*
- *Autosize all columns by default when opening attribute table*
- *Copy features as*: defines the format to use for data when pasting features in other applications.
  - *Plain text, no geometry*: attributes are copied as text but the geometry is skipped
  - *Plain text, WKT geometry*: attributes are copied as text and the geometry is returned in WKT
  - *Plain text, WKB geometry*: attributes are copied as text and the geometry is returned in WKB (hex format)
  - *GeoJSON*: attributes and geometry are formatted as GeoJSON data

- *Atributo tabela de comportamento* : configura o filtro na abertura da tabela de atributos. Há três possibilidades: 'Mostrar todos os atributos', 'Show atributos selecionados' e 'Mostrar atributos visíveis no mapa'.
- *Visualização padrão*: Definir o modo de visualização da tabela de atributo a cada abertura. Pode ser 'Lembrar ultima visualização', 'Ver tabela' ou 'Visualização de formulário'.
- *Cache de linha da tabela de atributos* . Essa linha de cache torna possível salvar as últimas linhas de N atributos carregados, assim trabalhar com a linha de atributos será mais rápido. O cache será deletado assim que fechar a tabela de atributos.
- *Representação para valores NULL*. Aqui, pode definir um valor para os campos de dados que contêm valores NULL.

Dica

Melhora a abertura da tabela de atributos de big data

Quando trabalhando com camadas com grandes quantidades de gravações, abrir a tabela de atributos pode ser vagaroso pois a caixa de diálogo chama todas as linhas na camada. Configurando *Comportamento de tabela de atributos* oara **Mostre atributos visíveis no mapa** fará o QGIS pedir somente os atributos do campo do mapa atual quando abrindo a tabela, permitindo um carregamento mais rápido.

Note que a informação nessa instância da tabela de atributos será sempre amarrada com a extensão da tela com a cuja a qual foi aberta, o que significa que selecionando **Mostrar todos os atributos** dentro de tal tabela não mostrara novos atributos. Entretanto, você pode atualizar o conjunto de atributos mostrado mudando a extensão da tela e selecionando a opção **Mostre Atributos Visíveis no Mapa** na tabela de atributos.

Manipulação das fontes de dados

- *Varredura para procurar itens válidos na doca do buscador* . você pode escolher entre 'Checar extensão' e 'Checar conteúdos de arquivo'.
- *Varredura para conteúdos de arquivos compressos (.zip) na doca do buscador* defina o quão detalhado é a informação do widget na parte inferior do painel do buscador quando filtrnado arquivos do tipo. 'Não', 'Varredura básica' e 'Varredura total' são opções possíveis.
- *Prompt para subcamadas quando abrindo-as*. Alguns rasters suportam subcamadas — eles são chamados de sub conjuntos de informações em GDAL. Um exemplo são arquivos netCDF — Se há muitas variaveis netCDF, o GDAL vê toda variável como sub conjunto de informações. A opção permite você controlar como lidar com sub-camadas quando um arquivo com subcamadas é aberto. Você tem as seguintes escolhas:
  - 'Sempre': Perguntar sempre (se existem subcamadas)
  - 'Se necessário': Perguntar se a camada não tem bandas, mas tem subcamadas
  - 'Nunca': Nunca pede, não irá carregar nada
  - 'Carregar tudo': Nunca pede, mas carrega todas as subcamadas
- *Recarregue automáticamente diretórios na doca do buscador quando o conteúdo deles mudar `: permite que você manualmente exclua diretórios de monitoramento no painel :guilabel:`Buscador´* por padrão (exemplo, para evitar potencial atraso devido a latência da internet).

Caminhos de dados localizados

É possível utilizar caminhos localizados para qualquer tipo de arquivo baseado na fonte da informação. Eles são uma lista de caminhos nos quais são usados para abstrair a localização da fonte da informação. Por exemplo, se `C:\my_maps` é listado nos caminhos localizados, uma camada contendo `C:\my_maps\my_country\ortho.tif` como fonte de dados é salvada no projeto usando `localized:my_country\ortho.tif`.

Os caminhos são listados por ordem de preferência. Em outras palavras, o QGIS irá olhar primeiro para o arquivo no primeiro caminho, e então para aquele no segundo caminho, e assim sucessivamente.

Caminhos de buscador escondidos

Esse widget lista todas as pastas que você escolhe esconder do *Browser painel*. Removendo a pasta da lista ira fazer disponível no painel *Buscador*.

Configurações GDAL

GDAL is a data exchange library for geospatial data that supports a large number of vector and raster formats. It provides drivers to read and (often) write data in these formats. The *GDAL* tab exposes the drivers for raster and vector formats with their capabilities.

GDAL drivers raster e vetor

As abas *Raster Drivers* e *Vector Drivers* permitem você definir qual driver GDAL é disponível para ler e ou escrever arquivos, assim como em alguns casos mais de um driver GDAL é disponibilizado.

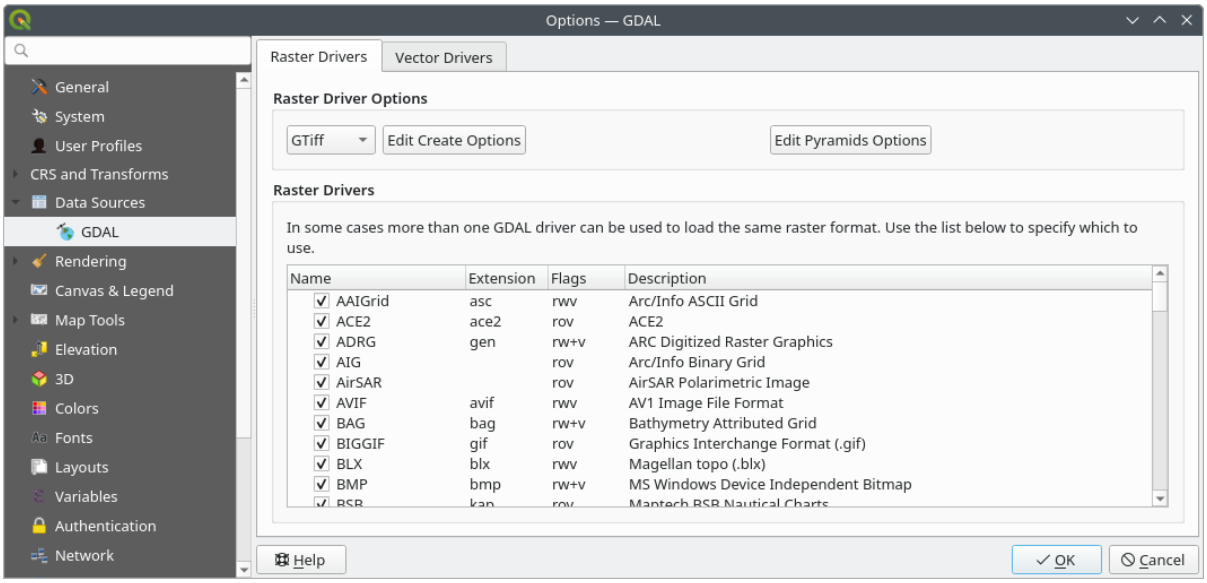


Figura5.7: Configurações GDAL - drivers Raster.

> **Dica**
>
> Clique duas vezes em um driver de Raster que permite acesso de leitura e escrita (`rw+(v)`) abra a caixa de diálogo *Opções de editar criação* para customização.

Opções de driver Raster

Esse pedaço provê maneiras de customizar o comportamento dos drivers raster que suportam acesso de leitura e escrita:

- *Edit create options*: allows you to edit or add different profiles of file transformation, i.e., a set of predefined combinations of parameters (type and level of compression, BIGTIFF support, blocks size, overview, colorimetry, alpha…) to use when outputting raster files. The parameters depend on the driver.

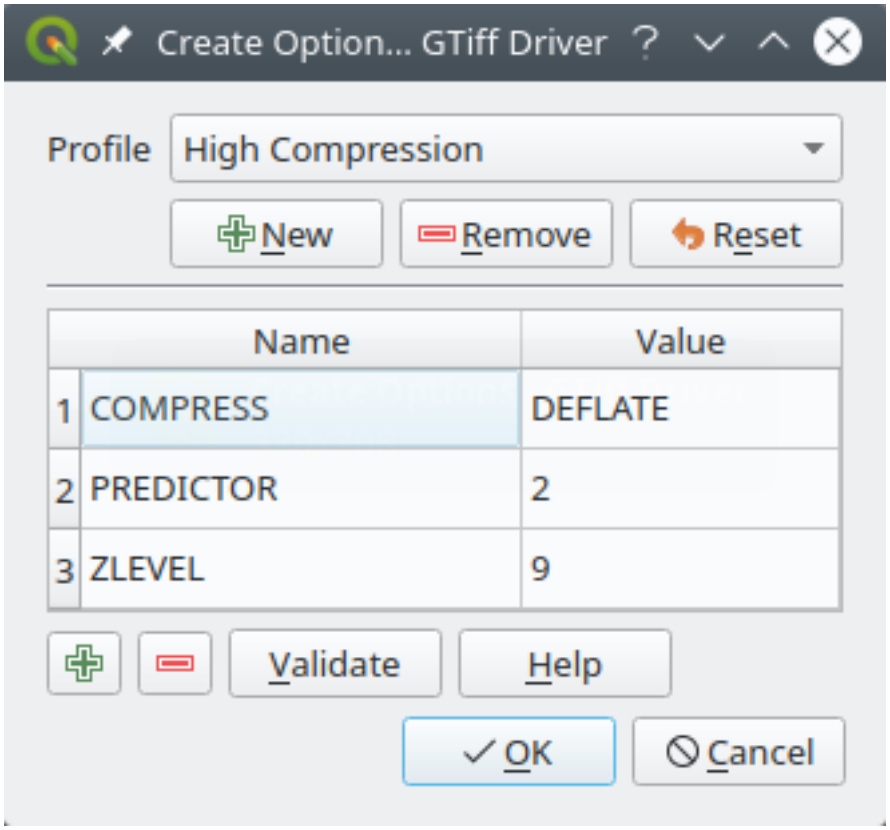


Figura5.8: Opções de criação de sample (Para GeoTiff).

A parte superior da caixa de diálogo lista os perfis atuais e permite você adicionar novos ou remover algum deles. Você pode também reiniciar o perfil para seus parâmetros padrões se você tem que mudá-los. Alguns drivers (por exemplo, GeoTiff) tem algumas amostrar de perfis que você consegue trabalhar.

Na parte inferior da caixa de diálogo:

- O botão deixa você adicionar linhas para preenche-las o nome e valor do parâmetro.
- O botão deleta o parâmetro selecionado
- Click no botão *Validate* para verificar que as opções de criação colocadas para o formato dado são válidas.
- Use the *Help* button to find the parameters to use, or refer to the GDAL raster drivers documentation.

- *Opções de edição de pirâmides*

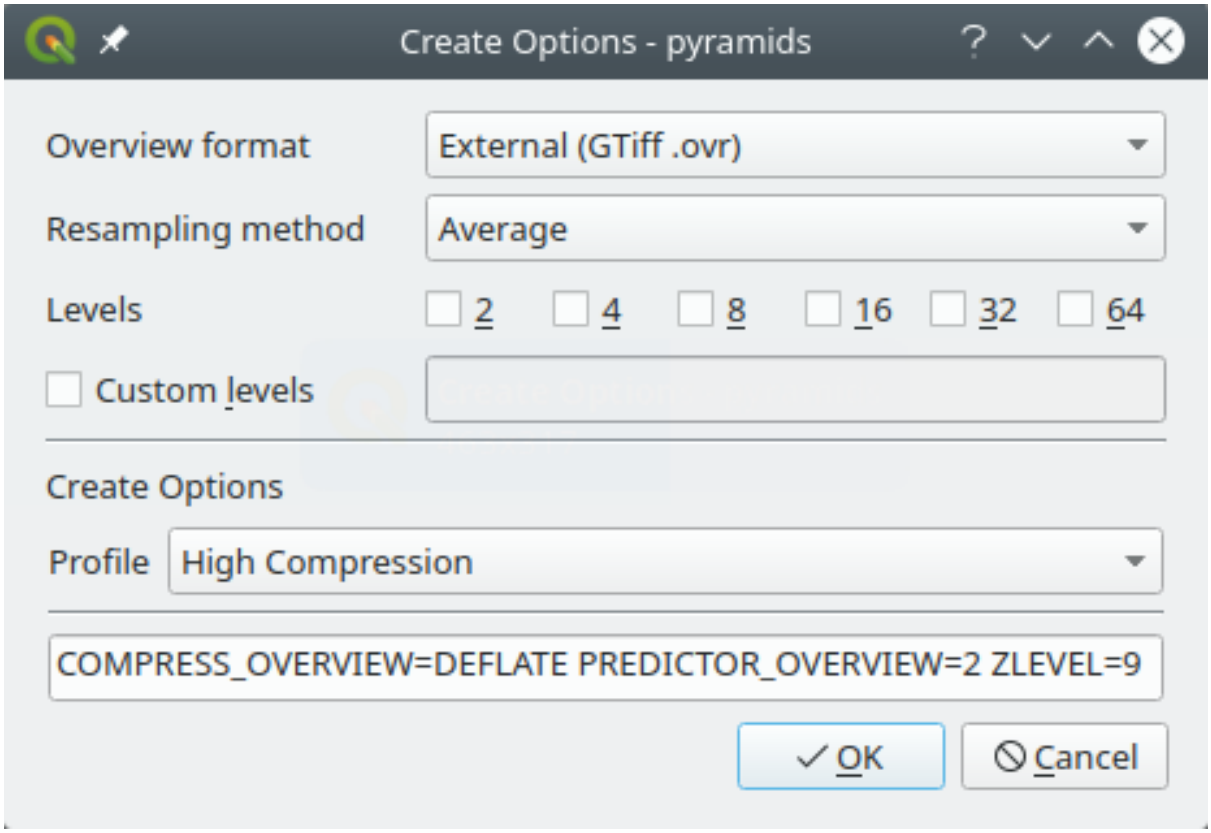


Figura5.9: Perfil de amostra de pirâmides.

5.1.6 Configurações de renderização.

A aba *Renderização* provê configurações para controlar camadas sendo renderizadas na região visível do mapa.

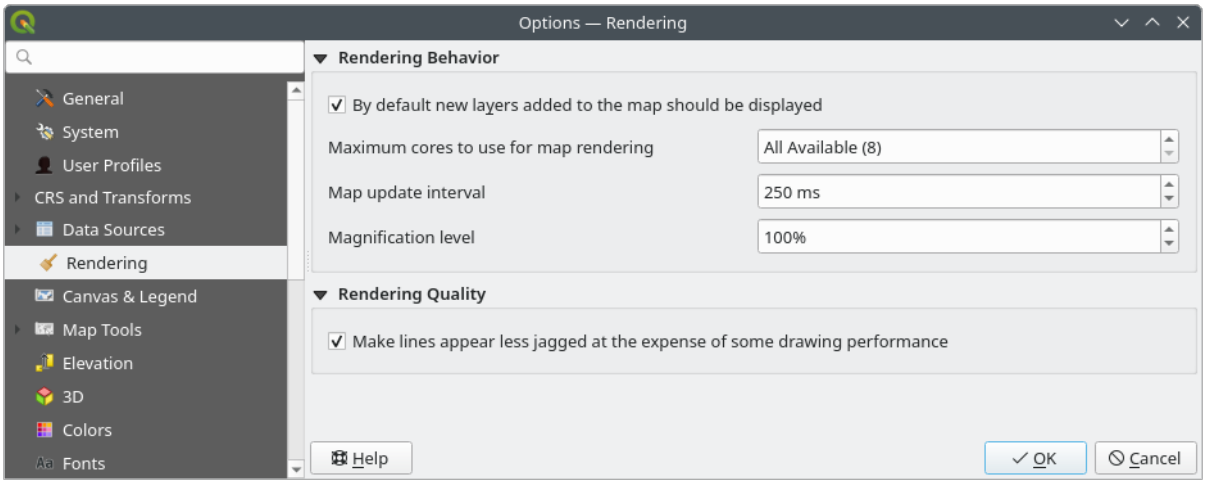


Figura5.10: Configurações de renderização.

Comportamento de renderização

- *Por padrão, novas camadas adicionadas ao mapa devem ser mostradas*: desmarcando esse opção pode ser útil quando carregando múltiplas camadas para evitar cada nova camada de ser renderizada na região visível e

diminuindo a velocidade do processo

- Configure a aba *Núcleos máximos a serem usados para renderização de mapa*.
- A tela do mapa renderiza em plano de fundo em uma imagem separada e a cada :guilabel:`Intervalo de atualização de mapa`(padronizado em 250 ms) o conteúdo dessa imagem (fora de tela) irá ser pego para atualizar a representação visível da tela. De qualquer forma, se renderizar terminar mais rápido que essa duração, será mostrado instantaneamente.
- *Nível de magnificação* (veja a *magnifier*)

Qualidade de Renderização

- *Faz com que as linhas apareçam menos irregulares, em detrimento de algum desempenho do desenho*

Configurações de renderização vetorial

A aba *Vetor* contêm configurações específicas para renderizar camadas vetoriais.

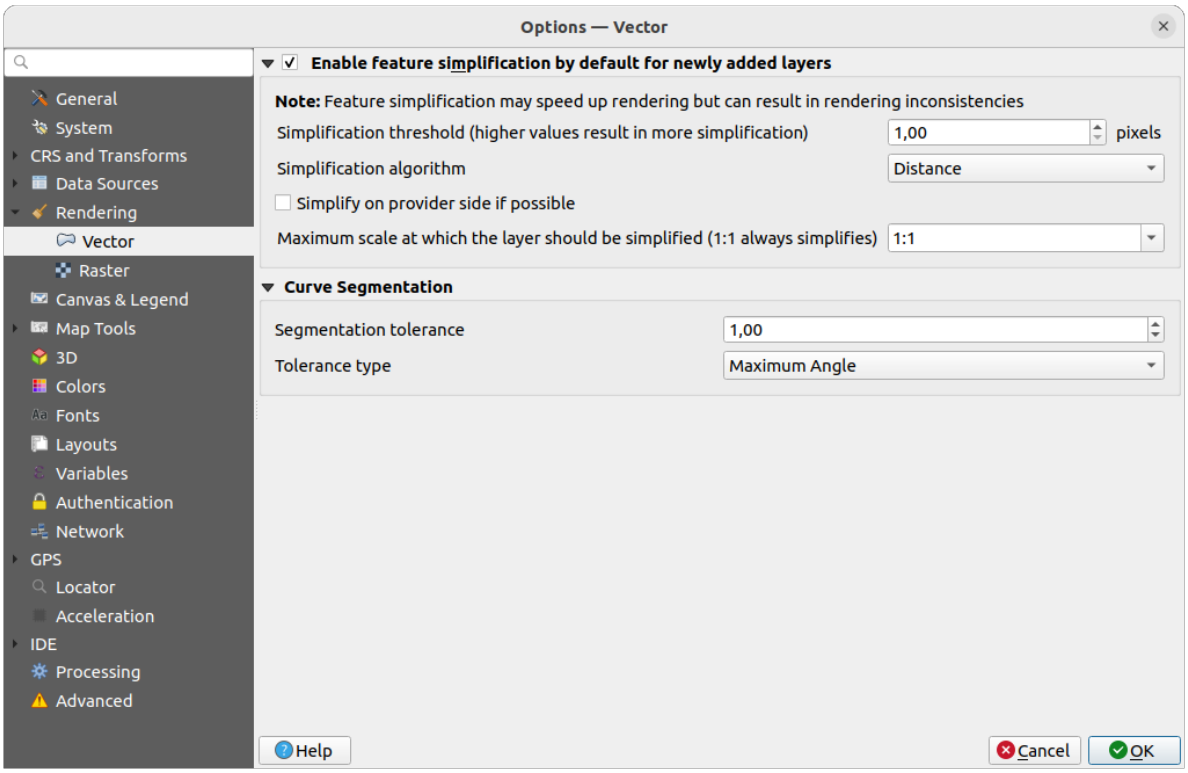


Figura5.11: Configurações de renderização vetorial

- *Disponibiliza a simplificação do atributo por padrão para novas camadas adicionadas*: você simplifica características' da geometria (menos nós) como resultado, eles são mostrados mais rapidamente. Esteja atento que isso pode causar gerar inconsistência de renderização. Configurações disponíveis são:
  - *limite de simplificação (valores maiores resultam em mais simplificação)*
  - *Simplification algorithm*: This option performs a local "on-the-fly" simplification on feature's and speeds up geometry rendering. It doesn't change the geometry fetched from the data providers. This is important when you have expressions that use the feature geometry (e.g., calculation of area) - it ensures that these calculations are done on the original geometry, not on the simplified one. For this purpose, QGIS provides three algorithms: 'Distance' (default), 'SnapToGrid' and 'Visvalingam'.

  - *Simplify on provider side if possible*: the geometries are simplified by the provider (PostgreSQL, Oracle…) and unlike the local-side simplification, geometry-based calculations may be affected
  - *Escala máxima em cada camada deve ser simplificada(1:1 sempre simplifica)*

> **Nota**
>
> Para além da configuração global, simplificação de atributo pode ser configurado para qualquer camada específica por meio do menu *Propriedades da camada* ► *renderização*

- *Segmentação em curva*
  - *Tolerância de segmentação*: essa configuração controla o jeito que os arcos são renderizados. **O menor** ângulo máximo (entre os dois vértices consecutivos e o centro de curva, em graus) ou diferenças máximas (distância entre o segmento de dois vértices e a linha de curva, em unidades de mapa), os segmentos **mais linhas retas** serão usados durante a renderização.
  - *Tipo de tolerância*: Pode ser *Ângulo máximo* ou *Máxima diferença* entre aproximação e curva.

Configurações de renderização Raster

A aba *Raster* contém configurações específicas para camadas de configuração Raster.

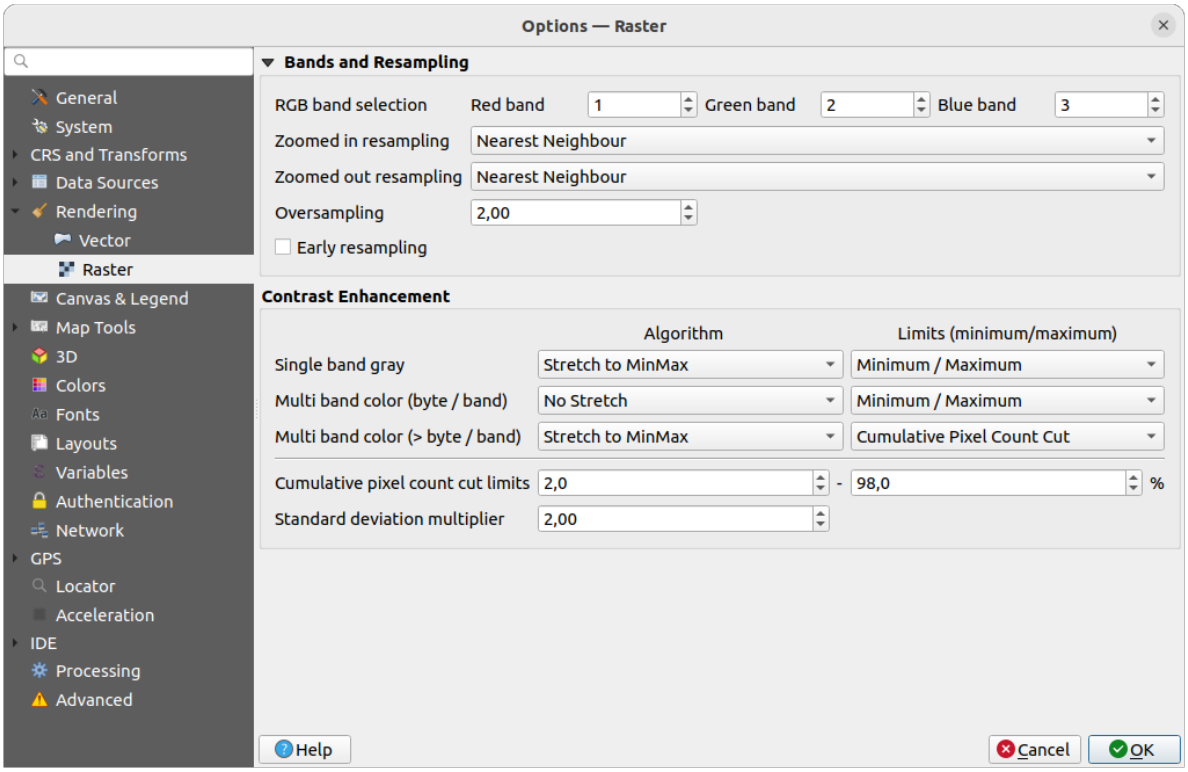


Figura5.12: Configurações de renderização Raster

Embaixo *Bandas e Reamostragem*:

- Com a *Seleção de banda RGB* pode definir o número para a banda Vermelha, Verde e Azul.
- The *Zoomed in resampling* and the *Zoomed out resampling* methods can be defined. For *Zoomed in resampling* you can choose between three resampling methods: 'Nearest neighbour', 'Bilinear (2x2 kernel)' and 'Cubic (4x4 kernel)'. For *Zoomed out resampling* you can choose between 'Nearest Neighbour' and 'Average'. You

can also set the *Oversampling* value (between 0.0 and 99.99 - a large value means more work for QGIS - the default value is 2.0).

- *Reamostragem antecipada `: permite calcular a renderização do raster a nível do provedor onde a resolução da fonte é conhecida, e garante um zoom melhor quando renderizando com a customização de estilo do QGIS. Muito conveniente para quadriculação de rasters carregados usando uma :ref:`Método de interpretação <interpretation>`.* A opção também pode ser configurada a nível de camada (*Simbologia* propriedades)

Opções de *Melhoradores de contraste* podem ser aplicadas para *Banda cinza única*, *Cor multi banda (byte/band)* ou *Cor multi banda (>byte/band)*. Para cada, você pode configurar:

- A *Algoritmo* para usar, cujos valores podem ser 'Sem esticar', 'Esticar para MinMax', 'Esticar e clipar para MinMax' ou 'Clipar para MinMax'
- A *Limites (mínimo/máximo)* para aplicar com valores tais como 'Contagem de cortes de pixels acumulativos, 'Mínimo/Máximo', 'Média +/- Desvio padrão'.

As opções da *Aprimoramento de contraste* também incluem:

- *Limites de contagem cumulativa de pixeis de corte*
- *Multiplicador do desvio-padrão*

5.1.7 Configurações de tela e legenda

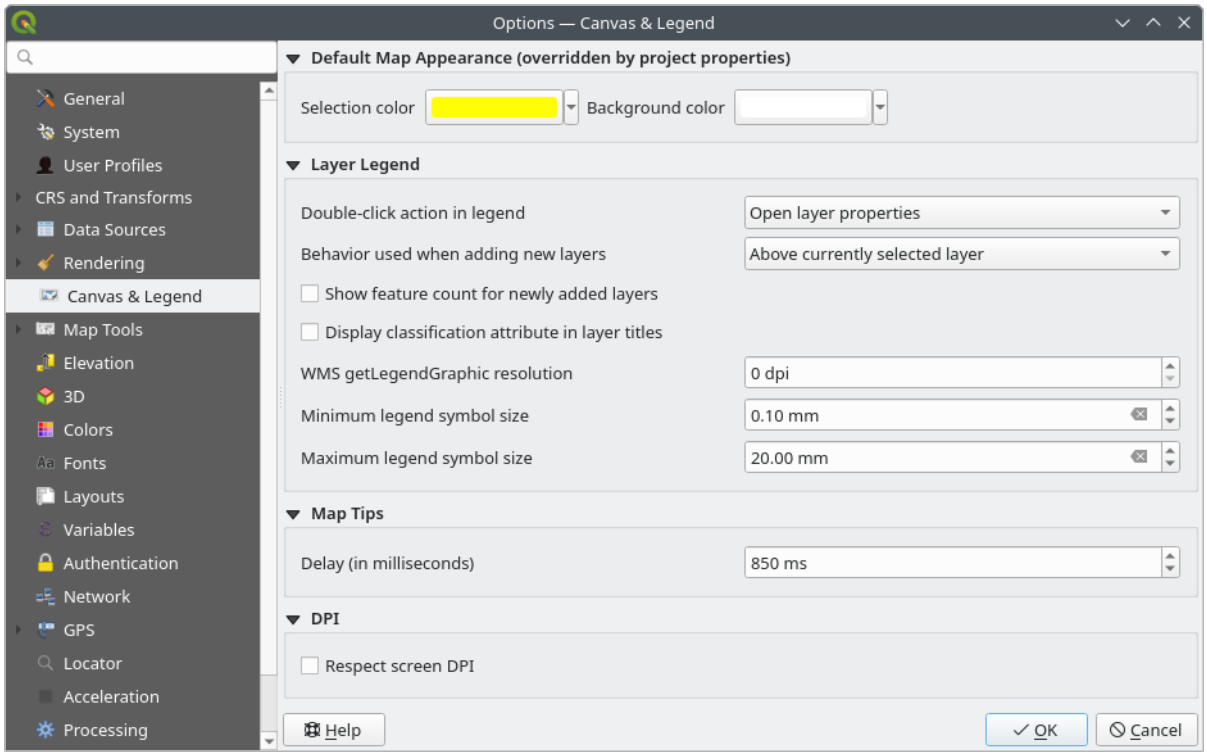


Figura5.13: Configurações de tela & legenda

Essas propriedades permitem que você defina:

- O **Aparência padrão de mapa (sobescrita pelas propriedades do projeto)**: o *Seleção de cor* e *Cor de fundo*
- **Legenda de camada** interação:
  - *Double click action in legend*: whether a double-click on a layer should either *Open layer properties* (default), *Open attribute table* or *Open layer styling dock*.

  - *Behavior used when adding new layers*: determines where layers are placed in the *Layers* panel when loaded into QGIS. It can be:
    * *Above currently selected layer*
    * *Always on top of the layer tree*
    * *Optimal index within current layer tree group*: Unlike the other options that sort the new layers among them and place them as a stack at the desired location, this option extents the sorting logic to the whole layer tree (or the active group) and inserts new layers in an "optimal" fashion by insuring that point layers sit on top of point layers, followed by line layers on top of lines, followed by polygon layers, etc.
  - ☐ *Mostrar contagem de recursos para camadas recém-adicionados*: exibe no painel *Camadas* o número de recursos ao lado do nome da camada. A contagem de recurso de classes, se houver, também é exibida. Você pode clicar com o botão direito do mouse em uma camada para ativar/desativar sua contagem de recursos.
  - ☐ *Display classification attribute names* in the Layers panel, e.g., when applying a categorized or rule-based renderer (see *Propriedades da simbologia* for more information).
  - O *WMS getLegendGraphic resolução*
  - *Minimo* e *Tamanho máximo para símbolo de legenda* para controlar o tamanho do símbolo no painel *Layers* .
- O *Atraso* em milissegundos de camada para mostrar *dicas de mapa*
- Whether QGIS should ☑ *Respect screen DPI*: If enabled, QGIS will attempt to display the canvas with physically accurate scale on screen, depending on the monitor's physical DPI. Symbology with specified display size will also be rendered accurately, e.g., a 10mm symbol will show as 10mm on screen. However, label font sizes on canvas may differ from those in QGIS' UI or other applications. If this setting is turned off, QGIS will use the operating system's logical DPI, which will be consistent with other applications on the system. However, canvas scale and symbology size may be physically inaccurate on screen. In particular, on high-dpi screens, symbology is likely to appear too small.

  Para uma melhor experiência, é recomendável habilitar a **|caixa de seleção|** *Respeite o DPI da tela*, especialmente ao usar monitores múltiplos ou diferentes e preparar mapas visualmente de alta qualidade. Desativando a **|caixa de seleção|** *Respeitar o DPI da tela* gerará uma saída que pode ser mais adequada para mapeamento destinado apenas para uso na tela, especialmente onde os tamanhos de fonte devem corresponder a outros aplicativos.

> **Nota**
>
> A renderização em layouts não é afetada pela configuração *Respeitar tela DPI*; ele sempre respeita o DPI especificado para o dispositivo de saída de destino. Observe também que essa configuração usa o DPI da tela física conforme relatado pelo sistema operacional, que pode não ser preciso para todos os monitores.

5.1.8 Configurações de ferramentas de mapa

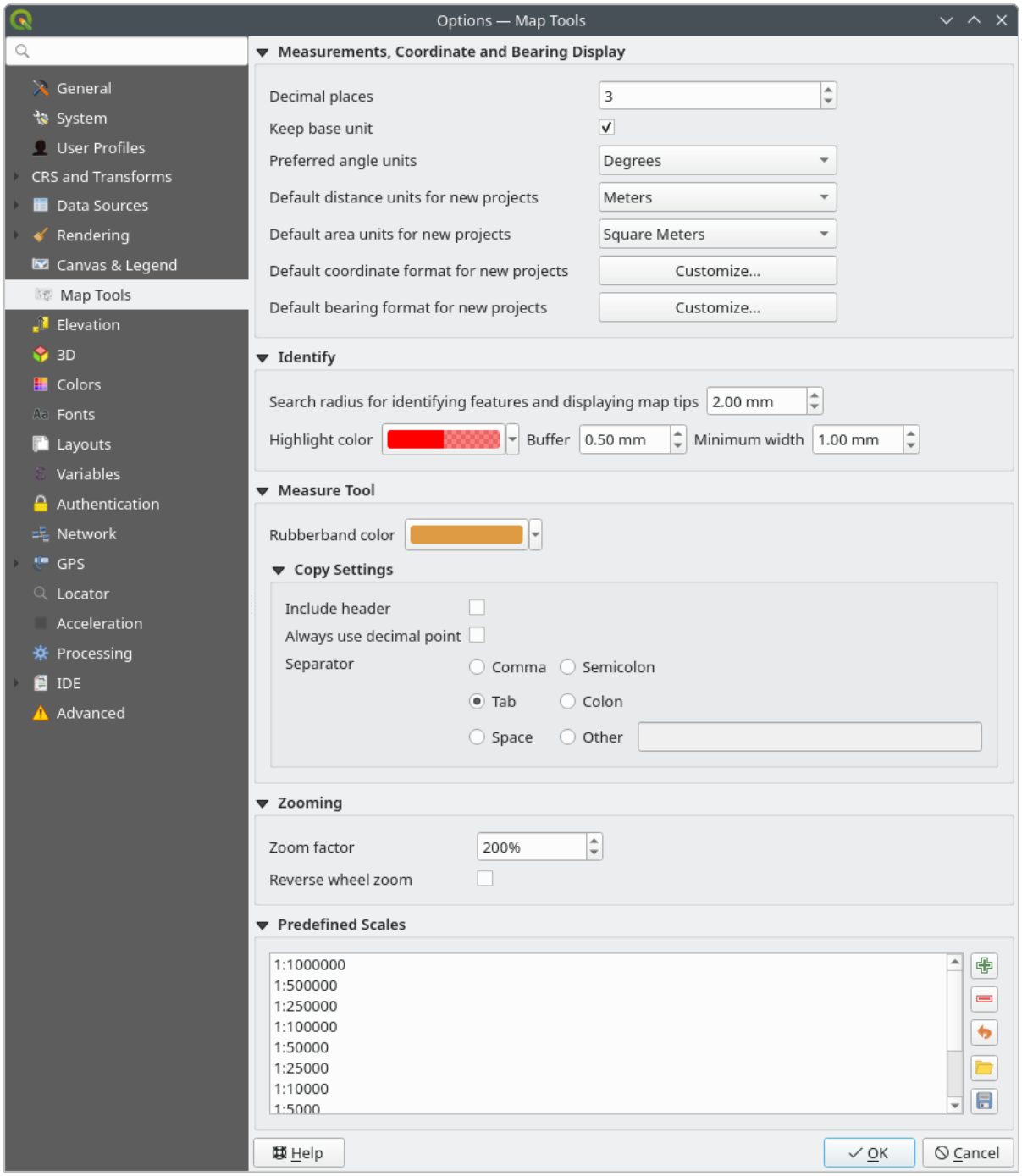


Figura5.14: Configurações de ferramentas de mapa

Measurements, Coordinate and Bearing Display

This section provides ways to configure default units parameters:

- Define *Casas decimais*
- *Manter a unidade base* para não converter automaticamente grandes números (por exemplo, metros para quilômetros)

- *Unidades de ângulo preferidas*: as opções são 'Graus', 'Radianos', 'Gon/gradianos', 'Minutos de arco', 'Segundos de arco', 'Giras/revoluções', miliradianos (definição do SI) ou mil (OTAN/definição militar)
- *Default distance units for new projects*: options are 'Meters', 'Kilometers', 'Feet', 'Yards', 'Miles', 'Nautical Miles', 'Centimeters', 'Millimeters', 'Inches', 'Degrees' or 'Map Units' (automatically matches the project's CRS units)
- *Preferred area units for new projects*: options are 'Square meters', 'Square kilometers', 'Square feet', 'Square yards', 'Square miles', 'Hectares', 'Acres', 'Square nautical miles', 'Square centimeters', 'Square millimeters', 'Square inches', 'Square degrees' or 'Map Units' (automatically matches the project's CRS units)
- *Formato de coordenadas padrão para novos projetos*, conforme exibido na caixa *Coordenadas* na barra de status do QGIS e na seção *Derivado* do :sup:Resultados da ferramenta *'Identificar recursos*.
- *Formato de orientação padrão para novos projetos*, conforme exibido na barra de status para a direção panorâmica da tela do mapa e pelo :sup:Ferramenta *Medir rolamento*.

The global options can be overridden at the project level, *here* for measurement units and *here* for coordinate and bearing formatting.

Identify tool

This group offers some options regarding the behavior of the *Identify tool*.

- *Raio de pesquisa para identificação de feições e exibição de dicas de mapa* é uma distância de tolerância dentro da qual a ferramenta de identificação mostrará os resultados, desde que você clique dentro dessa tolerância.
- *Cor de realce* permite que você escolha com quais características de cor que estão sendo identificadas devem ser destacadas.
- *Buffer* determina uma distância de buffer a ser renderizada a partir do contorno do destaque de identificação.
- *Largura mínima* determina o quão grossa deve ser a espessura do contorno de um objeto realçado.

Ferramenta de medida

- Define *Cor do elástico* para as ferramentas de medida

Measure Tool Copy Settings

These settings allow you to control the behavior of the *Copy* button in the Measure Line window. You can choose to *Include header* to keep column names, and you can also select *Separator* of your choice. You can also choose *Always use decimal point* to keep your coordinates copied to the clipboard with a dot as a decimal separator, even if your language settings in QGIS options are set up to use a comma. In this case, if you don't have the *Always use decimal point* option checked, you will be unable to select comma as a field separator.

Movendo e ampliando

- Define a *FFator zoom* para ferramentas de zoom ou roda do mouse.
- *Reverse wheel zoom* allows you to adjust mouse wheel scrolling direction.

Escalas pré-definidas

Aqui, você encontra uma lista de escalas predefinidas para exibir por padrão nos widgets suspensos relacionados à escala, como a barra de status *Scale*, o seletor de escalas de visibilidade ou configurações secundárias de visualização de mapa 2D,… Com os botões de e você pode adicionar ou remover suas escalas pessoais. Você também pode importar ou exportar escalas de/para um arquivo `.XML`. Observe que você ainda tem a possibilidade de remover suas alterações e redefinir para a lista predefinida.

Na caixa de diálogo de propriedades do projeto, você também pode definir sua própria lista de escalas, substituindo esta global nos widgets.

Definições de digitalização

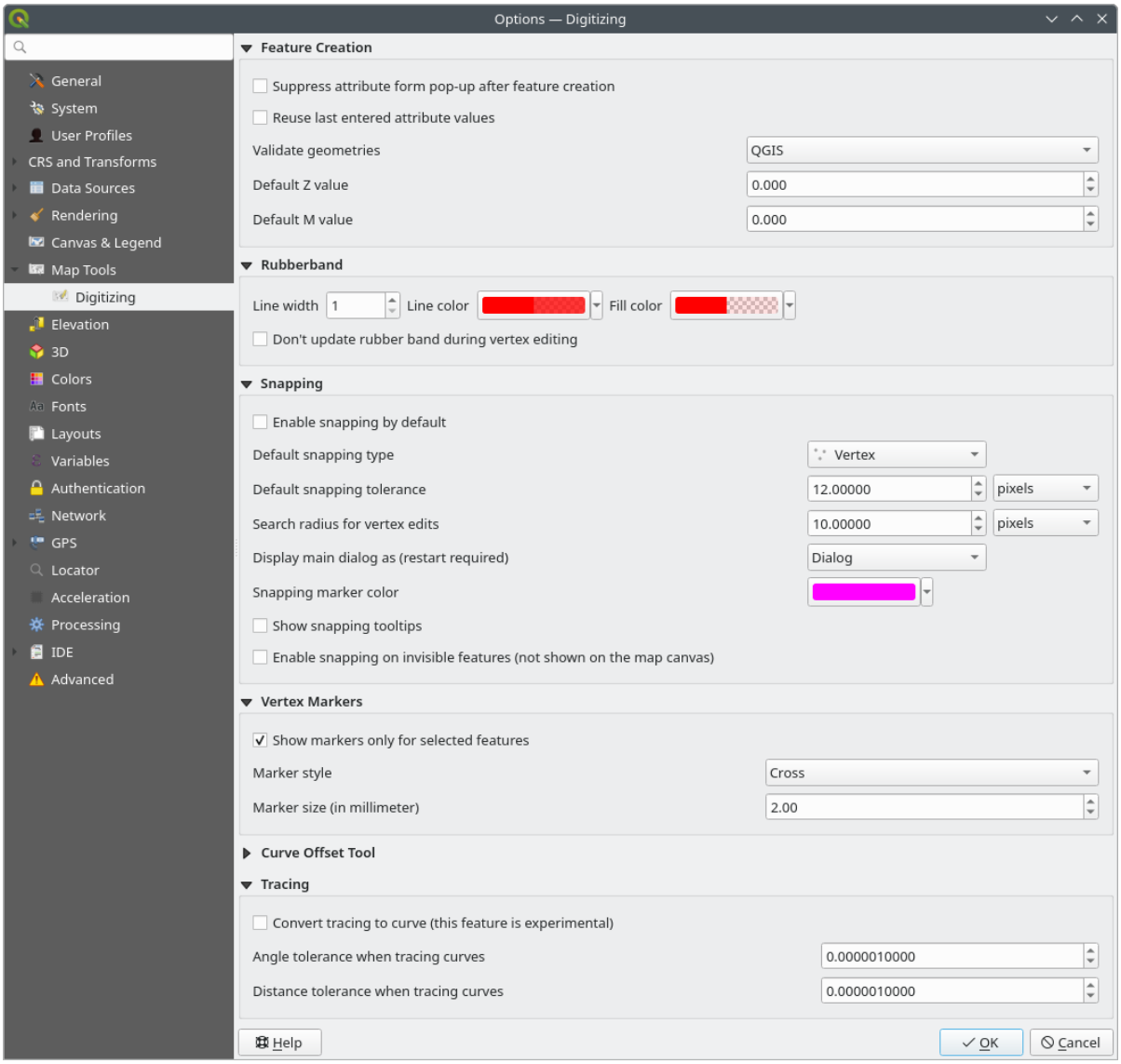


Figura5.15: Definições de digitalização

Esta guia ajuda você a definir as configurações gerais ao *editar camada vetorial* ` (atributos e geometria).

Criação de elementos

- :guilabel:` Suprimir pop-up de formulário de atributo após a criação do recurso`: esta opção pode ser substituída em cada caixa de diálogo de propriedades da camada.
- *Reuse last entered attribute values*: remember the last used value of every attribute and use it as default for the next feature being digitized. Works per layer. This behavior can also be controlled on a per-field basis (see *Configure o comportamento do campo*).
- *Validate geometries*. Editing complex lines and polygons with many nodes can result in very slow rendering. This is because the default validation procedures in QGIS can take a lot of time. To speed up rendering, it is possible to select GEOS geometry validation or to switch it off. GEOS geometry validation is much faster, but the disadvantage is that only the first geometry problem will be reported.

  Note que dependendo da seleção, reportamento de erros de geometria podem diferir (veja ref:*typesofgeomerrors*).

- *Valor Z padrão* para usar ao criar novos recursos 3D.

Elástico

- Defina Rubberband *Largura da linha*, *Cor da linha* e *Cor do preenchimento*.
- *Não atualize o elástico durante a edição do vértice*.

Ajuste

- *Ativar encaixe por padrão* ativa o encaixe quando um projeto é aberto
- Define *Default snap mode* (‘Vertex’, ‘Segment’, ‘Centroid’, ‘Middle of segments’, ‘Line endpoints’, ‘Area’)
- Define *Tolerância de atração pré-definida* em unidades de mapa ou pixeis
- Define o *Raio de pesquisa para editar vértices* em unidades de mapa ou pixeis
- *Exibir a caixa de diálogo principal como (é necessário reiniciar)*: define se a caixa de diálogo de ajuste avançado deve ser exibida como ‘Caixa de diálogo’ ou ‘Acoplamento’.
- *Cor do marcador de encaixe*
- *Show snapping tooltips* such as name of the layer whose feature you are about to snap. Helpful when multiple features overlap.
- *Habilitar encaixe em feições invisíveis (não mostradas na tela do mapa)*

Marcadores de Vértices

- *Mostrar marcadores apenas para elementos selecionados*
- Definir vértice *Estilo do marcador* (‘Cruz’ (padrão), ‘Círculo semitransparente’ ou ‘Nenhum’)
- Definir vértice *Tamanho do marcador (em milímetros)*

Ferramenta de curva de afastamento

The next options refer to the Offset Curve tool in *Digitalização Avançada*. Through the various settings, it is possible to influence the shape of the line offset.

- *Join style*: specifies whether ‘Round’, ‘Miter’ or ‘Bevel’ joins should be used when offsetting corners
- *Quadrant segments*: controls the number of line segments to use to approximate a quarter circle when creating rounded offsets
- *Miter limit*: Sets the maximum distance from the offset geometry to use when creating a mitered join as a factor of the offset distance (only applicable for miter join style)
- *End cap style*: controls how line endings are handled; they can be ‘Round’, ‘Flat’ or ‘Square’.

Rastreamento

Ativando a checkbox *Converter traçado em curva* você pode criar segmentos de curva enquanto digitaliza. Lembre-se de que seu provedor de dados deve oferecer suporte a esse recurso.

5.1.9 Elevation settings

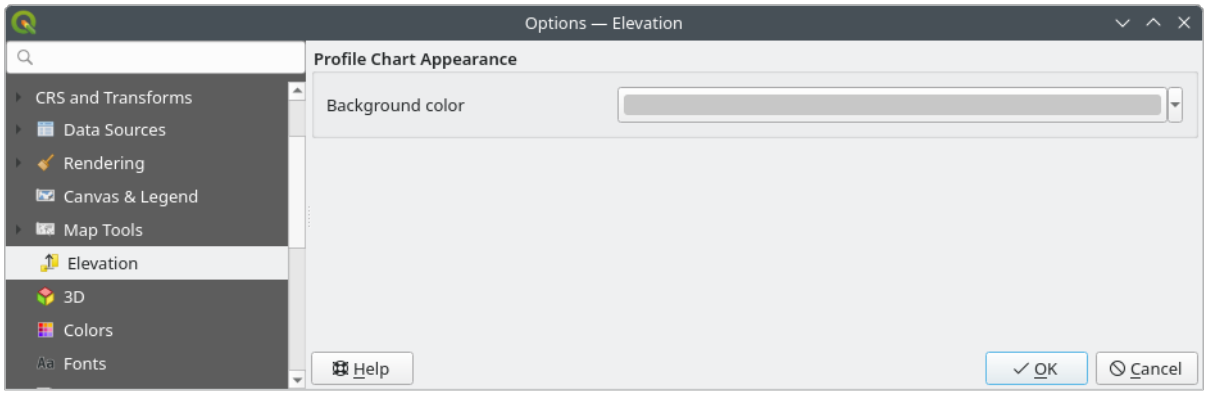


Figura5.16: Elevation settings

In *Elevation* menu, you can set a specific color to use as the *Background color* for *elevation profiles*. This can make the chart more readable for certain datasets, e.g., point clouds with RGB coloring, where the default background color is too similar to point colors to be easily discernable. If unchanged, the elevation profiles will continue to display using the standard system background color.

5.1.10 Configurações 3D.

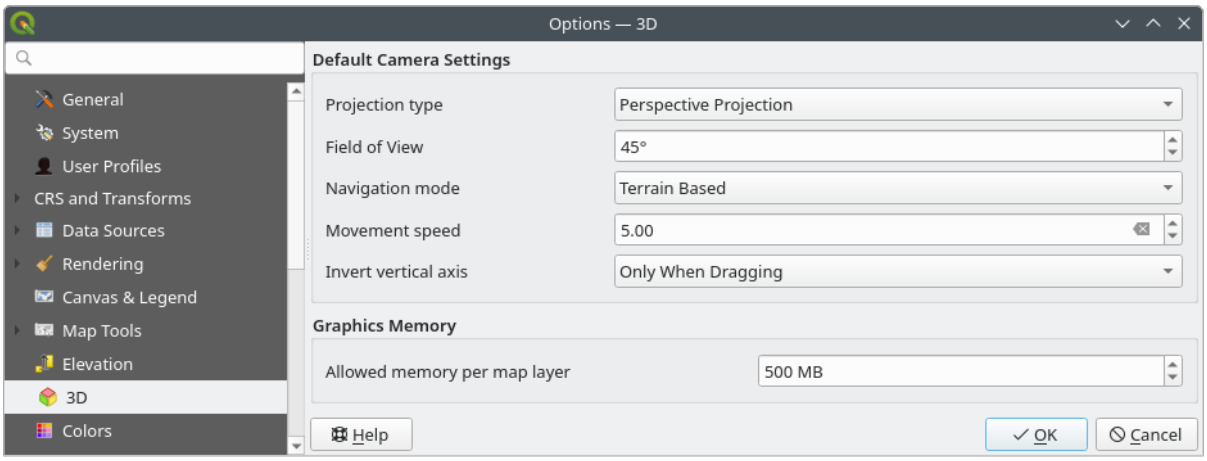


Figura5.17: Configurações 3D.

The *3D* menu helps you configure some default settings to use for any *3D Map view*. These can refer to:

- *Default Camera Settings*:
  - *Tipo de projeção*: permitindo visualizar a cena 3D em:
    - *Projeção em perspectiva* (padrão): Linhas paralelas parecem se encontrar à distância. Os objetos parecem encolher quanto mais longe estão da câmera.
    - ou um *Projeção ortogonal*: Linhas paralelas aparecem paralelas. Os objetos aparecem do mesmo tamanho, independentemente da distância.

  - Camera's *Campo de visão* da câmera: relevante apenas na projeção em perspectiva, especifica o campo de visão vertical atual em graus e determina quanto da cena é visível para a câmera. O valor padrão é 45°.
  - *Modo de navegação*: fornece diferentes meios para interagir com a cena 3D. Os modos disponíveis são:
    * *Baseada no terreno*: a câmera segue em torno de uma posição fixa na superfície do terreno enquanto a cena é navegada.
    * *Modo caminhada (primeira pessoa)*

    Dependendo do modo selecionado, *Comandos de navegação* difere.
  - *Velocidade de movimento*
  - *Inverter eixo vertical*: Controla se os movimentos do eixo vertical devem ser invertidos de seu comportamento normal. Afeta apenas o movimento no *modo de caminhada*. Pode ser definido para:
    * *Nunca*
    * *Apenas ao arrastar*: faz com que o movimento vertical seja invertido apenas ao executar uma rotação de câmera de clicar e arrastar
    * e *Sempre*: faz com que os movimentos sejam invertidos ao clicar e arrastar e quando o movimento da câmera é bloqueado para o cursor (através de um pressionamento de tecla ~)
- Under *Graphics memory*, the *Allowed memory per layer* option lets you set the GPU memory limit configuration on each layer. This is useful for users utilizing large 3D scenes which exhaust the available GPU memory resources. When a limit is hit, a warning is also displayed, which should assist in troubleshooting large scenes.

5.1.11 Configurações de cores

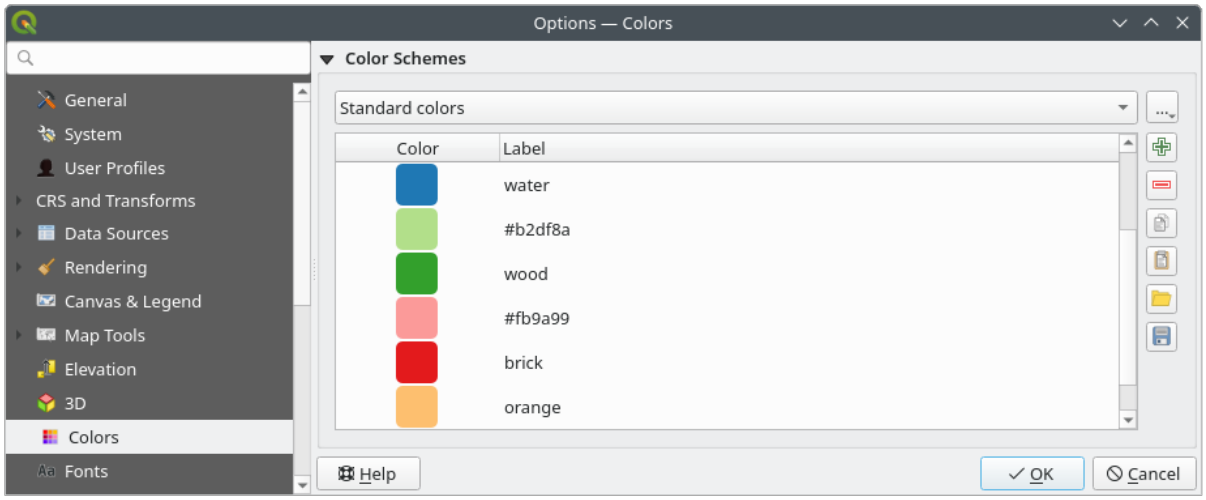


Figura5.18: Configurações de cores

Este menu permite criar ou atualizar paletas de cores usadas em todo o aplicativo no *widget seletor de cores*. YVocê pode escolher do:

- *Cores recentes* mostrando as cores usadas recentemente.
- *Standard colors*, the default palette of colors
- *Project colors*, a set of colors specific to the current project (see *Styles Properties* for more details)
- *Novas cores de camada*, um conjunto de cores para usar por padrão quando novas camadas são adicionadas ao QGIS

- ou paleta(s) personalizada(s) que você pode criar ou importar usando o botão … ao lado da caixa de combinação da paleta.

Por padrão, as paletas *Cores recentes*, *Cores padrão* e *Cores do projeto* não podem ser removidas e são definidas para aparecer no menu suspenso de botões de cores. Paletas personalizadas também podem ser adicionadas a este widget graças à opção *Mostrar nos botões coloridos*.

Para qualquer uma das paletas, você pode gerenciar a lista de cores usando o conjunto de ferramentas ao lado do quadro, ou seja:

- *Adicionar* ou *Remover* cor
- *Copy* or *Paste* color
- *Importar* ou *Exportar* o conjunto de cores do/para `.gpl` arquivo.

Double-click a color in the list to tweak or replace it in the *Color Selector* dialog. You can also rename it by double-clicking in the *Label* column.

5.1.12 Configurações de fontes

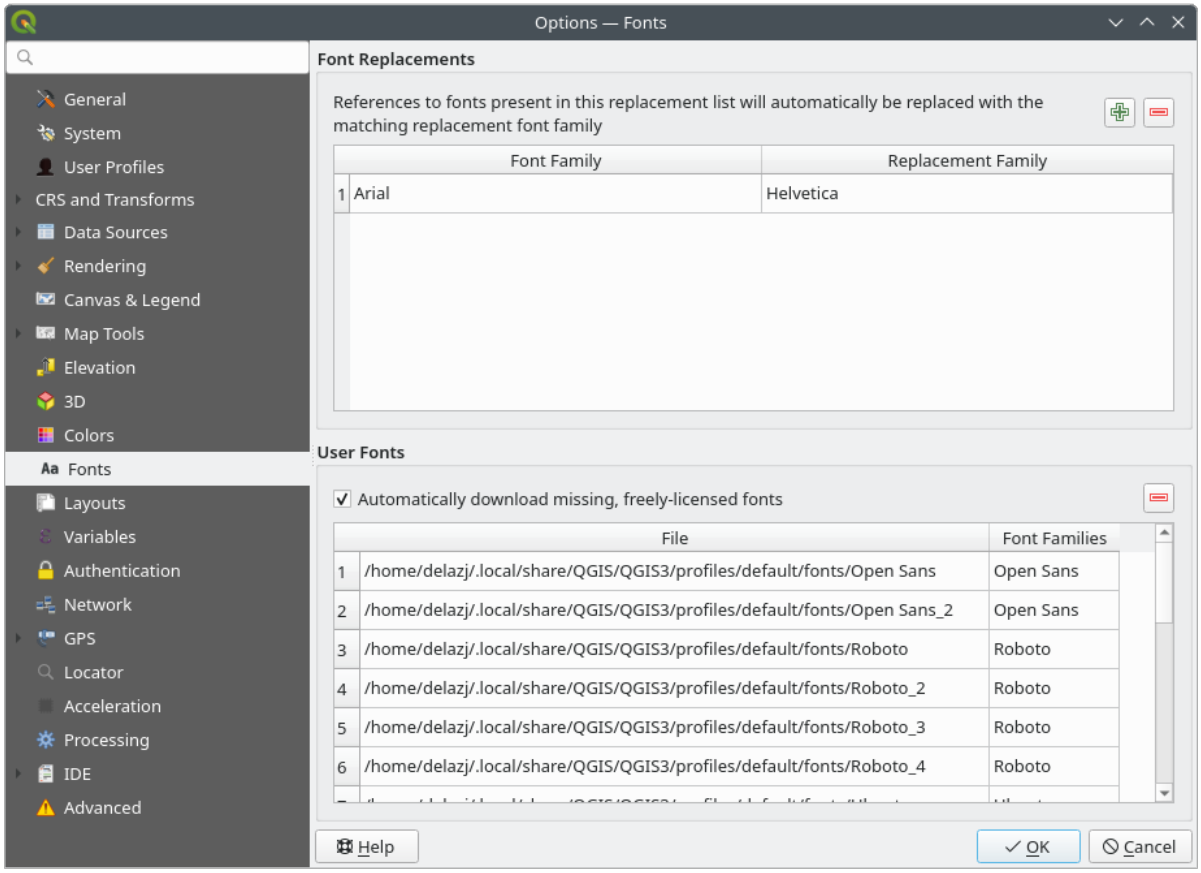


Figura5.19: Configurações de fontes

The *Fonts* tab provides support to manage fonts used across the projects:

- *Font Replacements*: gives ability to populate a list of automatic font replacements to apply when loading projects or styles, allowing better support for projects and styles to be used across different operating systems (e.g., replace “Arial” with “Helvetica”).

- *User Fonts*: Allows you to place TTF or OTF fonts in the `fonts` sub-folder of the *user profile*. These fonts can be automatically loaded at QGIS startup time. This provides a way to use fonts without requiring them to be installed on an operating system level, which is often blocked in enterprise environments. The panel lists all installed user fonts and allows you to manage (i.e., remove) previously installed user fonts.

  It is also possible to *Automatically download missing, freely-licensed fonts*: E.g., if you open a project or style, or try to load a vector tile layer that references fonts that aren't currently available, then a hard-coded list of freely licensed fonts to download via URL is consulted to determine whether it's possible to automatically download the font to the user profile font directory (with notification of the font license).

5.1.13 Configurações de compositor

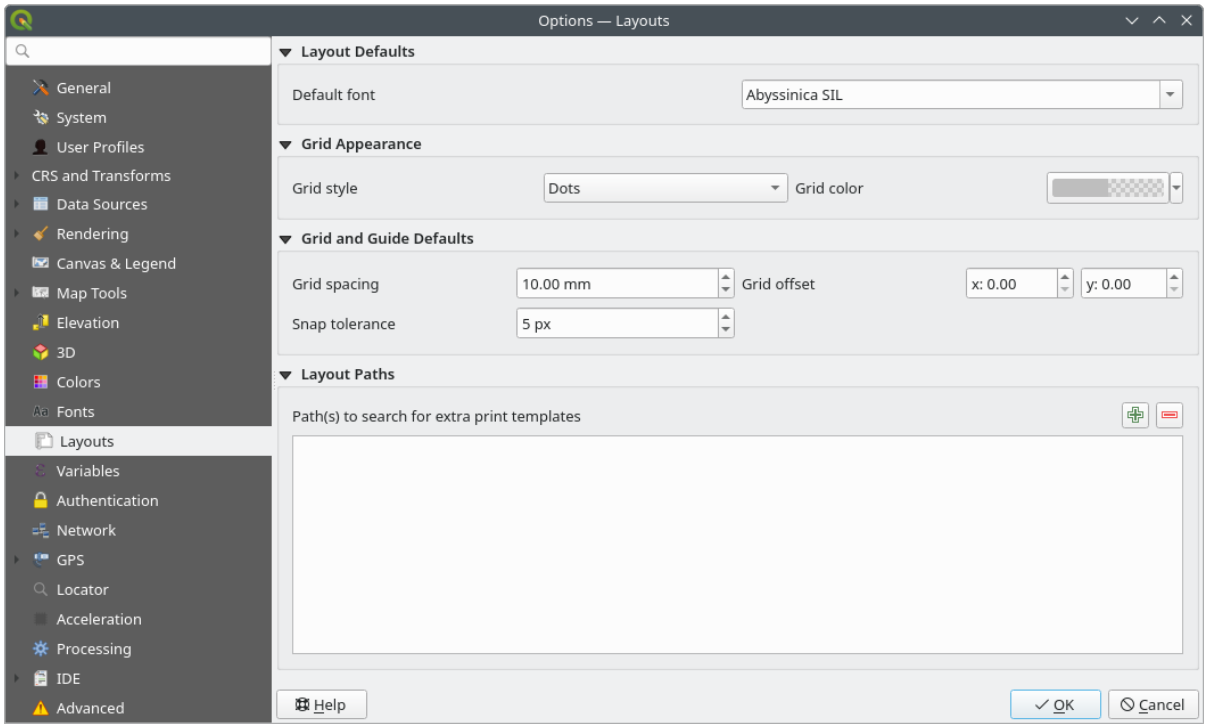


Figura5.20: Configurações de compositor

Composição padrão

Você pode definir o guilabel:*Fonte padrão* usado dentro do :ref:`Compositor de impressão `.

Aparência do Gride

- Defina o *Grid style* ('Solid', 'Dots', 'Crosses')
- Define a *Cor da grade*

Configurações de guia e grade

- Defina *Estilo de grade* ('Sólido', 'Pontos', 'Cruz')
- Defina o *Deslocamento da grade* de X e Y
- Define the *Snap tolerance*

Caminhos de compositor

- Defina *Caminho(s) para procurar modelos de impressão extras*: uma lista de pastas com modelos de layout personalizados para usar ao criar um novo.

5.1.14 Configurações de variáveis

A aba *Variáveis* lista todas as variáveis disponíveis no nível global.

Também permite ao usuário gerenciar variáveis de nível global. Clique no botão para adicionar uma nova variável de nível global personalizada. Da mesma forma, selecione uma variável de nível global personalizada na lista e clique em no botão para removê-lo.

More information about variables in the *Armazenando valores em Variáveis* section.

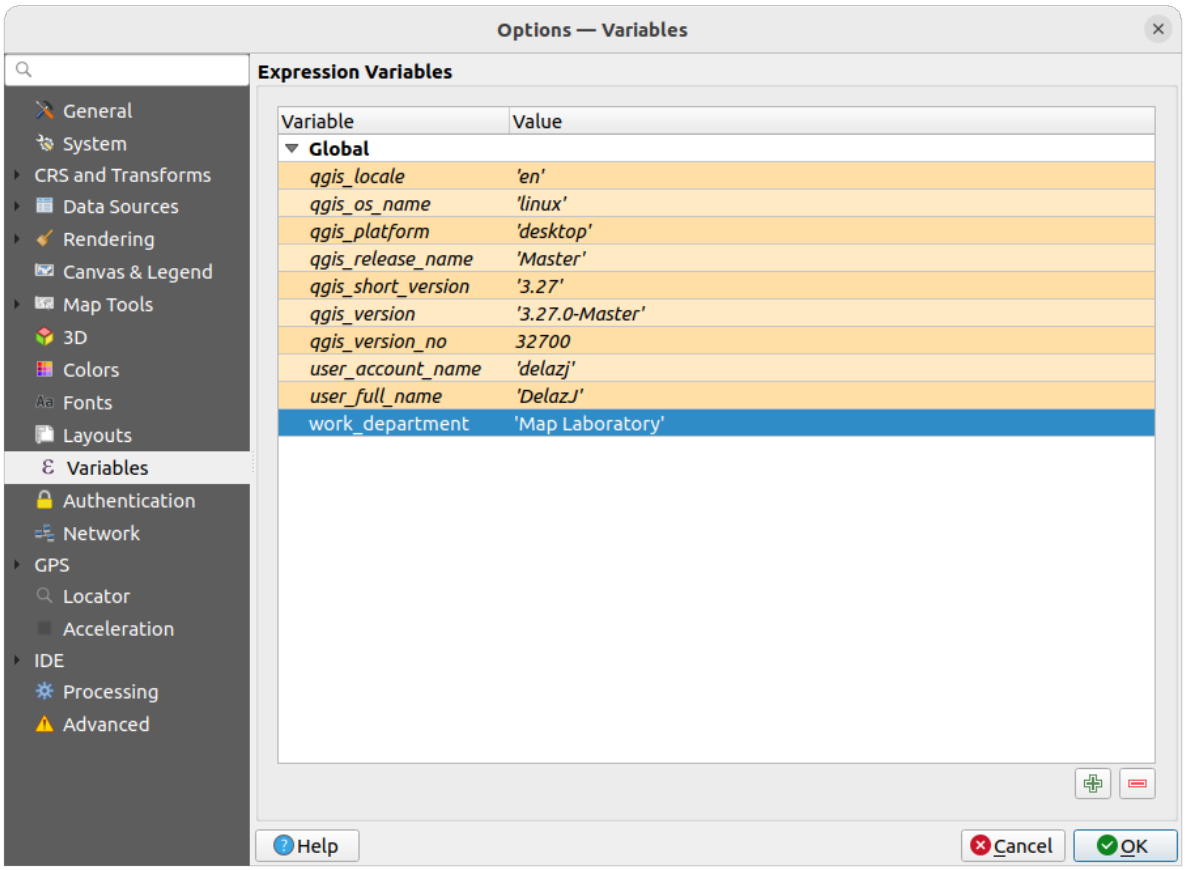


Figura5.21: Configurações de variáveis

5.1.15 Configurações de autenticação

Na aba *Autenticação* você pode definir configurações de autenticação e gerenciar certificados PKI. Veja Autentificação_índice para mais detalhes.

Para gerenciar as autenticações, você pode usar a lista de ferramentas ao lado do quadro, ou seja:

- Adicionar nova configuração de autenticação
- Remove a configuração de autentificação selecionada
- Editar configuração de autenticação selecionada

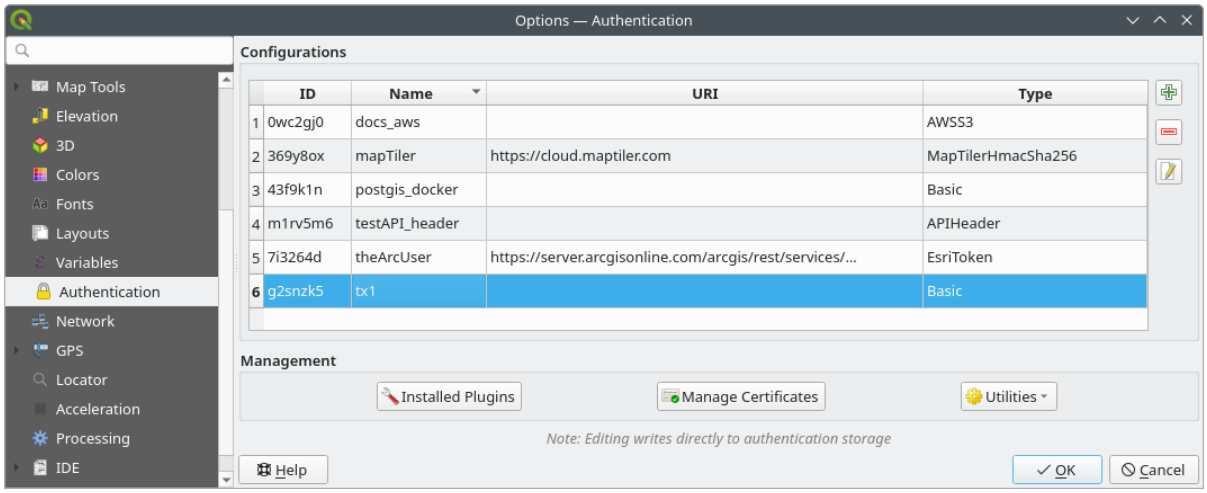


Figura5.22: Configurações de autenticação

5.1.16 Configurações de rede

Geral

- Define *Tempo esgotado para pedidos de rede (ms)* - o padrão é 60000
- Defina *Período de expiração padrão para recursos WMS (horas)* - o padrão é 24
- Defina *Período de expiração padrão para blocos WMS-C/WMTS (horas)* - o padrão é 24
- Defina *Tentativa máxima em caso de erros de solicitação de bloco ou recurso*
- Define *User-Agent prefix* which defaults to `Mozilla/5.0`. This value will be prepended to both QGIS and OS version e.g., `Mozilla/5.0 QGIS/33801/Ubuntu 22.04.4 LTS` to shape the user-agent.

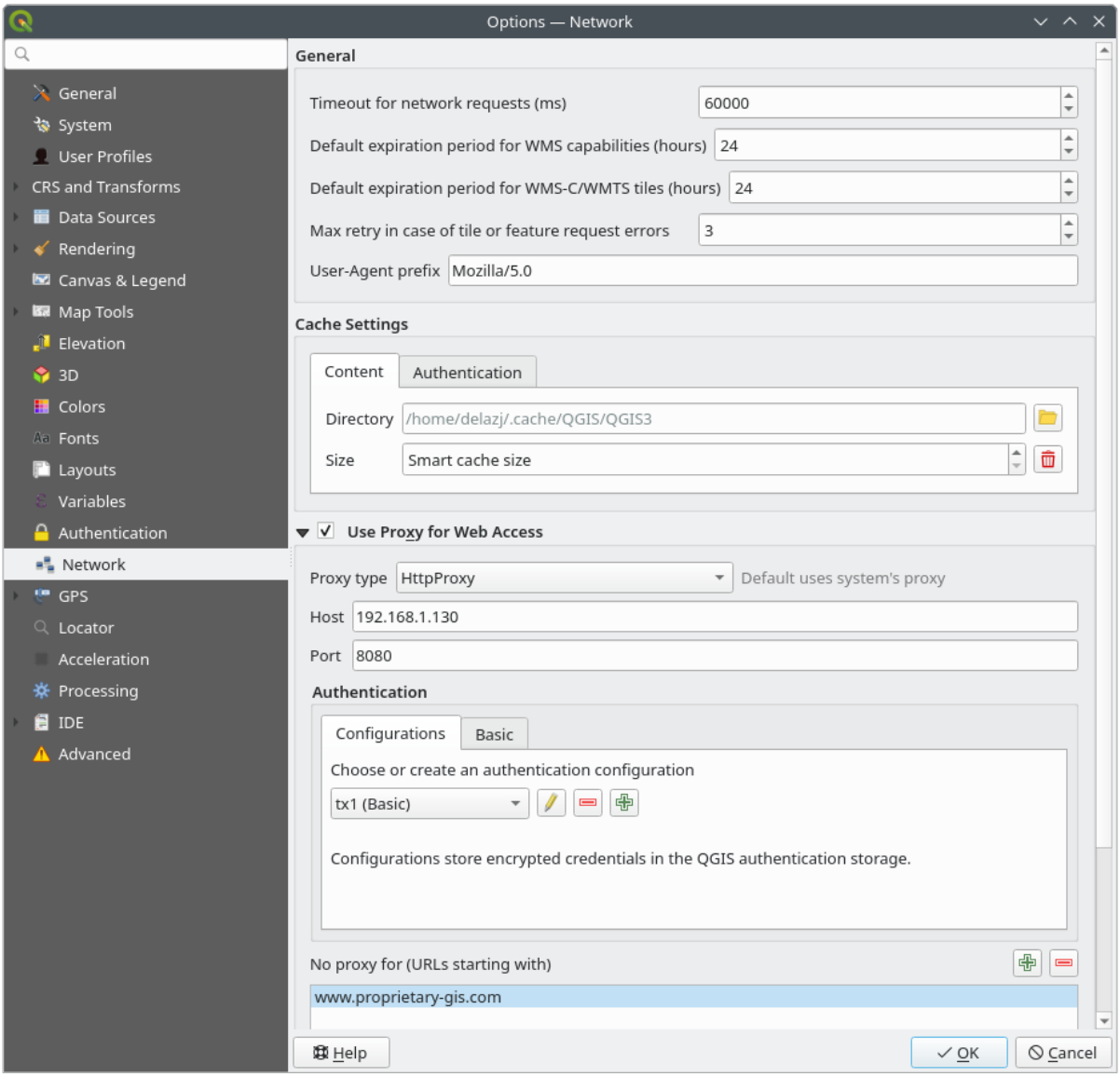


Figura5.23: Configurações de rede e proxy

Configurações de cache

Defines the *Directory* and a *Size* for the cache specified in megabytes. You can clear the value to enable the *Smart cache size*, which sets the maximum cache size based on available disk space. Also offers tools to *automatically clear the connection authentication cache on SSL errors (recommended)*.

Proxy para acesso à internet

- :*Usar proxy para acesso à web*
- Defina o *Tipo de proxy* de acordo com suas necessidades e defina 'Host' e 'Porta'. Os tipos de proxy disponíveis são:
  - *Proxy padrão*: O proxy é determinado com base no proxy do sistema
  - *Socks5Proxy*: Proxy genérico para qualquer tipo de ligação. Suporta TCP, UDP, unindo a uma porta (ligações de entrada) e autenticação.
  - *HttpProxy*: Implementado usando o comando "LIGAR" , apenas suporta ligações TCP de saída; suporta autenticação.

- *HttpCachingProxy*: Implementado usando comandos HTTP normais, é útil apenas em pedidos no contexto do HTTP.
- *FtpCachingProxy*: Implementado usando um proxy FTP, é útil no contexto de pedidos FTP.

As credenciais do proxy são definidas usando o:ref:*autentificação de widget <authentication>*.

A exclusão de algumas URLs pode ser adicionada à caixa de texto abaixo das configurações de proxy (consulte `Figura_Rede_Aba`). Nenhum proxy será usado se o URL de destino começar com uma das strings listadas nesta caixa de texto

Se você precisar de informações mais detalhadas sobre as diferentes configurações de proxy, consulte o manual da documentação da biblioteca QT subjacente em https://doc.qt.io/archives/qt-5.9/qnetworkproxy.html#ProxyType-enum

Dica

Usando Proxies

O uso de proxies às vezes pode ser complicado. É útil proceder por 'tentativa e erro' com os tipos de proxy acima, para verificar se eles funcionam no seu caso.

5.1.17 Configurações de GPS

The *GPS* dialog helps you configure GPS devices connections and properties in QGIS. It also provides settings for GPS tracking and data digitizing.

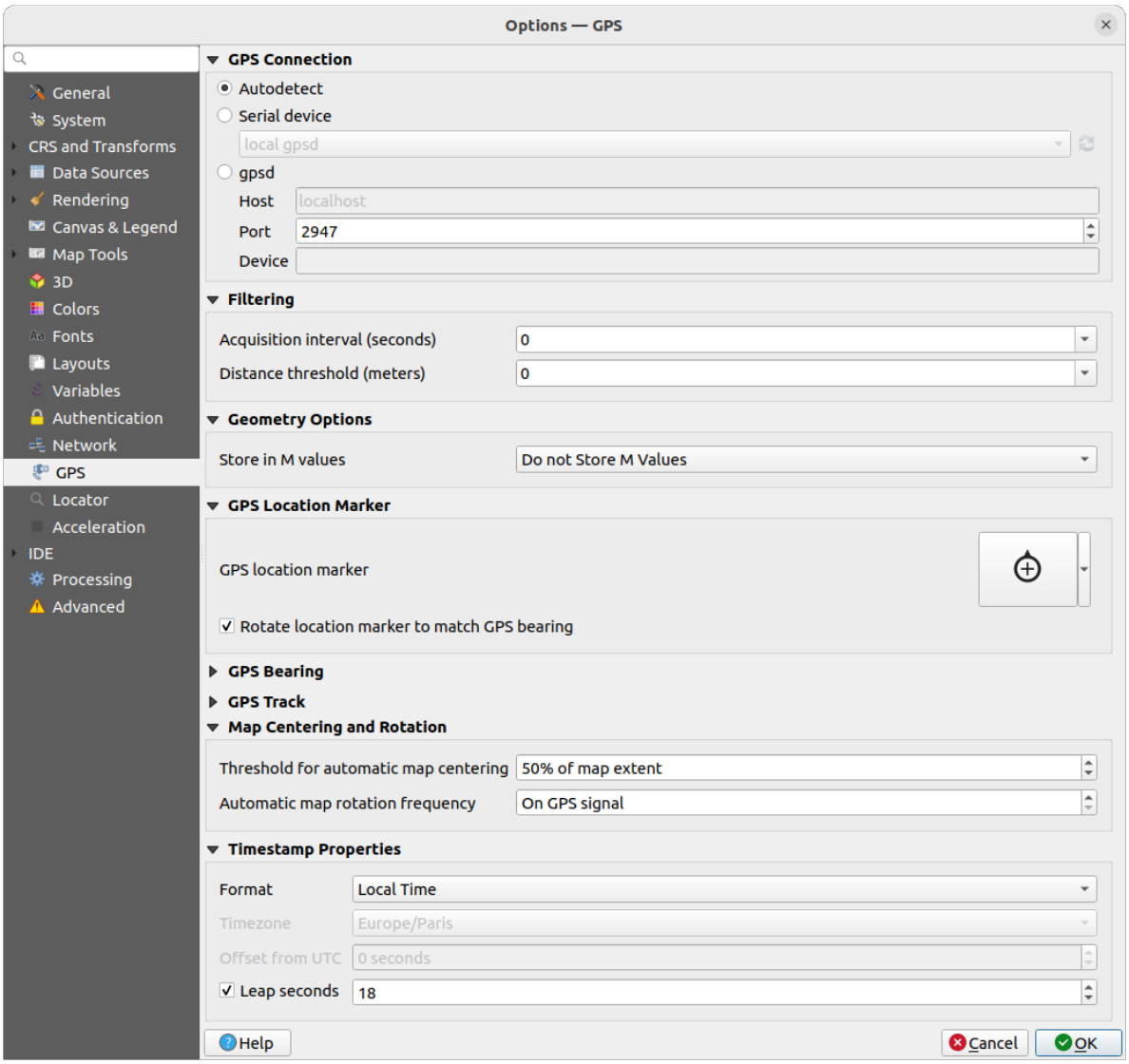


Figura5.24: Configurações de GPS

Você pode especificar:

- *GPS Connection*: provides different means to connect the device to QGIS
  - *Autodetectar*
  - *Dispositivo serial* (é necessário recarregar se um novo dispositivo GPS estiver conectado)
  - *gpsd* (selecting the *Host* and *Port* your GPS is connected to, and providing a *Device* name)

  In case of connection problems, you can try to switch from one to another.
- *Filtering*: You can set an *Acquisition interval (seconds)* and/or a *Distance threshold (meters)* parameters to keep the cursor still active when the receiver is in static conditions.
- Under *Geometry Options*, the *Store in M values* option allows to create geometries with M values from the inbuilt GPS logging tools. This applies to both features digitized from GPS logs and from the *Log to Geopackage/SpatiaLite* functionality… Options include storing timestamps (as ms since epoch), ground speed, altitudes, bearings, and accuracy components as m values.
- *GPS Location Marker*
  - Configure a symbol for the *GPS location marker*, indicating the current GPS position

  - *Rotate location marker to match GPS bearing*: whether the marker symbol should be rotated to match the GPS direction

- *GPS Bearing*:
  - configure a *Bearing line style* using QGIS line symbol properties
  - set whether to *Calculate Bearings from travel direction*: If checked, the bearing reported by the GPS device will be ignored and the bearing will instead be calculated by the angle between the previous two GPS locations.
- *GPS Track*: set symbol to use for the *Track line style*
- *Map Centering and Rotation*: defines when the map canvas is updated according to the user displacement on the field:
  - *Threshold for automatic map centering*: defines the minimal offset of the GPS position from the map canvas center to trigger an automatic *map recentering* if enabled.
  - *Automatic map rotation frequency*: defines how often the map rotation to match the GPS bearing could happen; it can be on a custom duration or *On GPS signal*.
- *Timestamp Properties* to configure how time values are displayed and stored in the data. Parameters include the *Format* which can be:
  - *Local time*
  - *UTC*
  - *UTC with offset*, to account for daylight savings offsets or other complex time zone issues
  - a specific *Time zone*

  Moreover, *Leap seconds* correction can be applied, by adding the seconds to GPS timestamp.

GPSBabel

GPSBabel converte waypoints, trilhas e rotas entre receptores GPS populares como Garmin ou Magellan e programas de mapeamento como Google Earth ou Basecamp. Literalmente centenas de receptores GPS e programas são suportados. O QGIS depende do GPSBabel para interagir com esses dispositivos e *Manipular as informações*.

For details on how-to, please refer to *Loading to or from a device*.

5.1.18 Configurações do localizador

A aba *Locator* deixa você configurar o *barra de localização*, um widget de pesquisa rápida disponível na barra de status para ajudá-lo a realizar pesquisas no aplicativo. Ele fornece alguns filtros padrão (com prefixo) para usar:

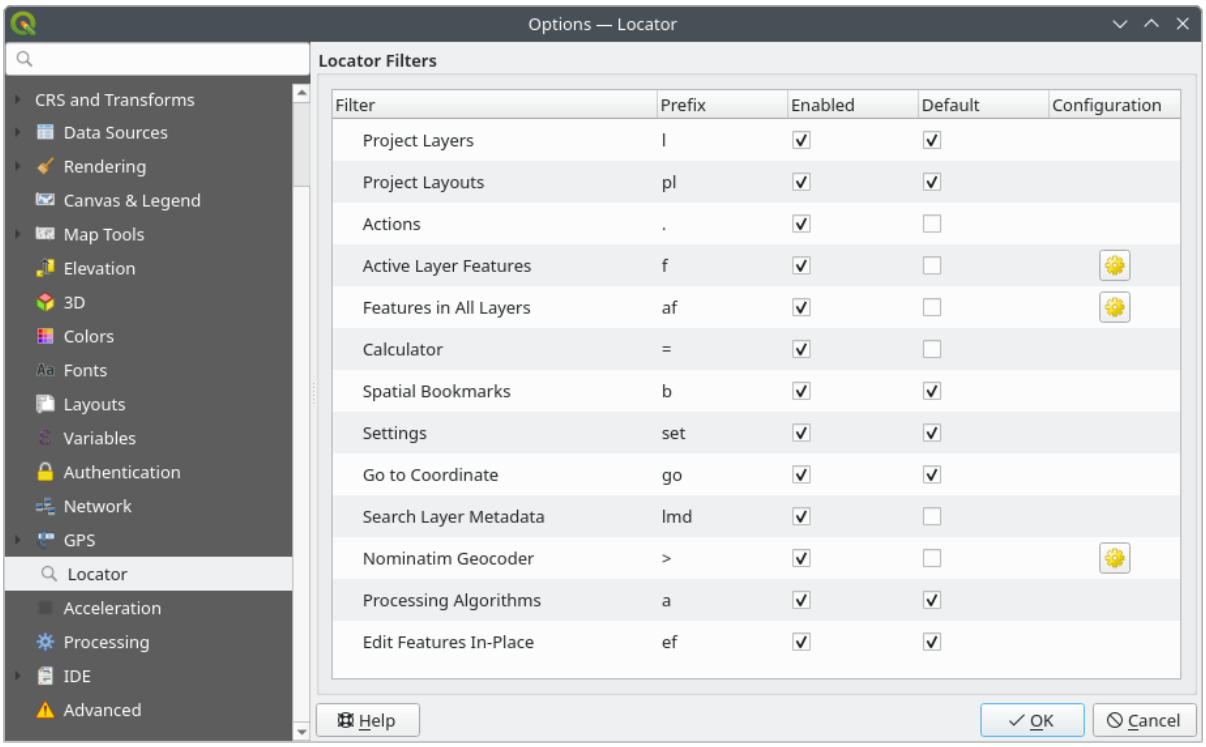


Figura5.25: Configurações do localizador

- *Camadas do projeto* (`l`): encontra e seleciona uma camada no painel *Camadas*.
- *Compositores do projeto* (`pl`): procura e abre um compositor de impressão.
- Ações (`.`): encontra e executa uma ação QGIS; ações podem ser qualquer ferramenta ou menu no QGIS, abrindo um painel…
- *Recursos de camada ativa* (`f`): procura por atributos correspondentes em qualquer campo da camada ativa atual e amplia para a feição selecionada. Pressione para configurar o número máximo de resultados.
- *Recursos em todas as camadas (``af``)*: procura por atributos correspondentes no:ref:*nome de exibição <maptips>* de cada *camadas pesquisáveis* e dá zoom para selecionar os recursos. Pressione para configurar o número máximo de resultado e o número máximo de resultados por camada.
- *Calculadora* (=): permite a avaliação de qualquer expressão do QGIS e, se válido, oferece a opção de copiar o resultado para a área de transferência.
- *Marcadores Espaciais* (`b`): localiza e amplia a extensão do marcador.
- *Configurações* (`configurações`): navega e abre as caixas de diálogo de propriedades do projeto e do aplicativo.
- *Go to Coordinate* (`vá`): desloca a tela do mapa para um local definido por uma vírgula ou um par de coordenadas x e y separados por vírgula ou um URL formatado (por exemplo, OpenStreetMap, Leaflet, OpenLayer, Google Mapas, …). A coordenada é esperada em WGS 84 (`epsg:4326`) e/ou mapa canvas CRS.
- *Nominatim Geocoder* (`>`): geocodes using the Nominatim geocoding service of the OpenStreetMap Foundation. Press to limit results to one or more countries using list of country codes. This helps narrow down the search to specific countries, avoiding ambiguous results from other locations. For example, entering `DE` or `DE,FR` restricts results to Germany or Germany and France, respectively.
- Algoritmos de processamento (`a`): procura e abre uma caixa de diálogo de algoritmo de processamento.
- *Editar recursos selecionados* (`ef`): dá acesso rápido e executa um *modify-in-place compatível* Algoritmo de processamento na camada ativa.

Na caixa de diálogo, você pode:

- customize the filter *Prefix*, i.e., the keyword to use to trigger the filter
- Definir se o filtro é *Ativado*: o filtro pode ser usado nas pesquisas e um atalho está disponível no menu da barra de localização
- Defina se o filtro é *Padrão*: uma pesquisa que não usa um filtro retorna resultados apenas das categorias de filtros padrão.
- Alguns filtros fornecem uma maneira de configurar o número de resultados em uma pesquisa.

O conjunto de filtros do localizador padrão pode ser estendido por plug-ins, por exemplo, para pesquisas nominatim OSM, pesquisa direta no banco de dados, pesquisas no catálogo de camadas, …

5.1.19 Configurações de aceleração

Configurações de aceleração OpenCL

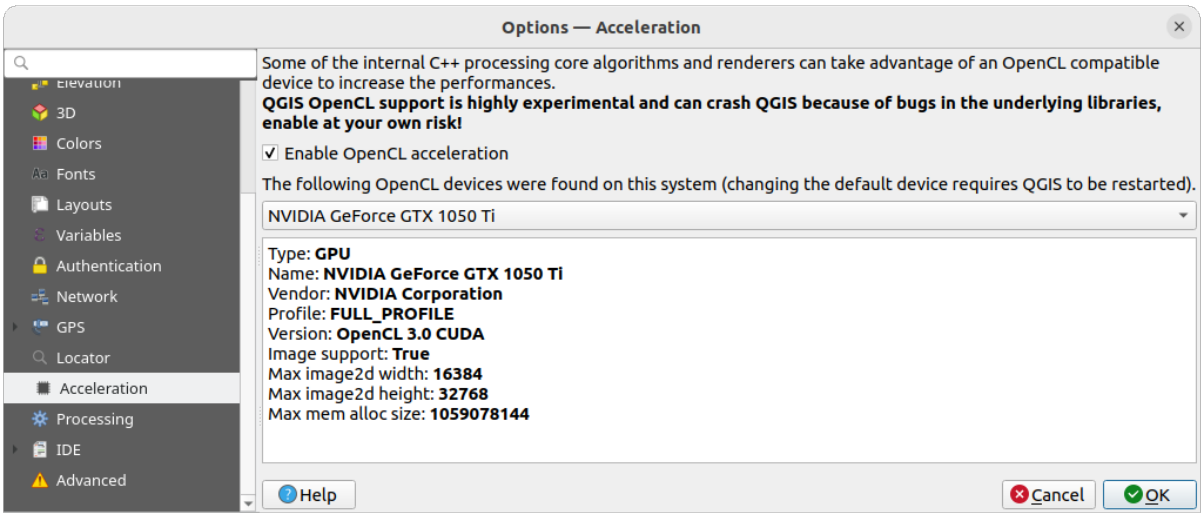


Figura5.26: Configurações de aceleração

Dependendo do seu hardware e software, pode ser necessário instalar bibliotecas adicionais para habilitar a aceleração do OpenCL.

5.1.20 Configurações de IDE

Under *GitHub access token*, you can generate a personal token allowing you to share code snippets from within the Python code editor. More details on GitHub authentication

Configurações de editor de código

Na aba *Editor de código*, você pode controlar a aparência e o comportamento dos widgets do editor de código (console e editor interativo Python, widget de expressão e editor de funções, …).

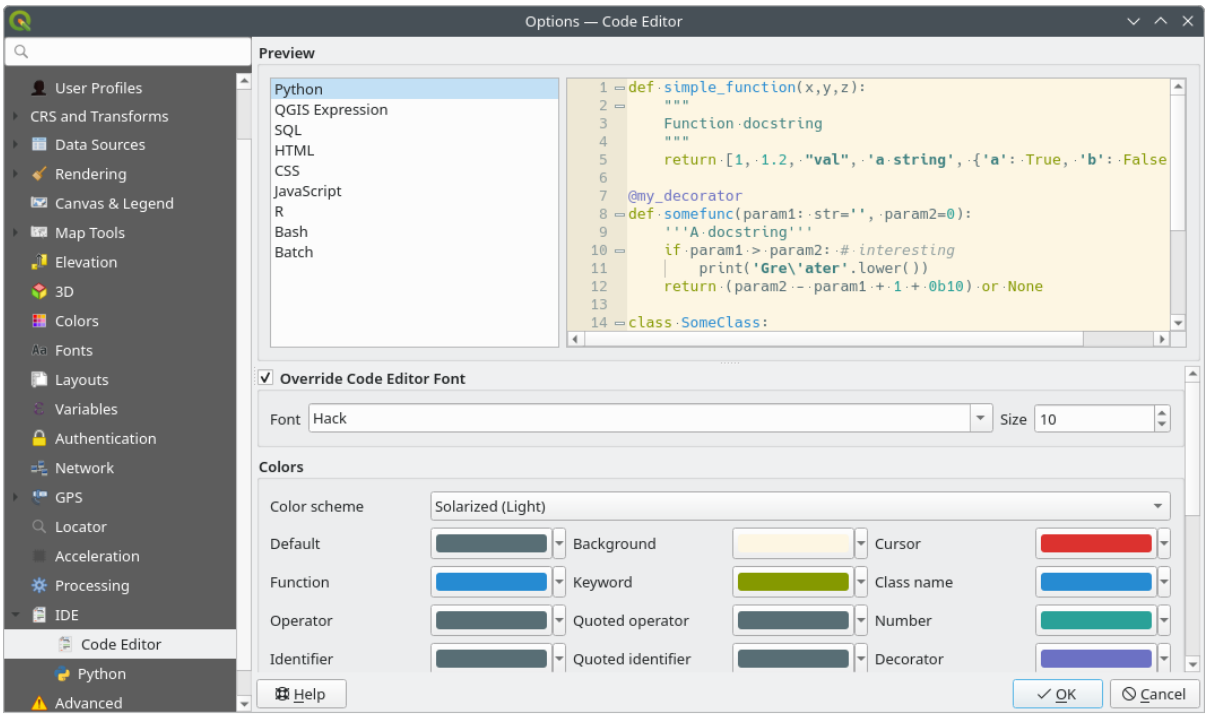


Figura5.27: Configurações de editor de código

Na parte superior da caixa de diálogo, um widget fornece uma visualização ao vivo das configurações atuais, em várias linguagens de codificação (Python, expressão QGIS, HTML, SQL, JavaScript). Uma maneira conveniente de ajustar as configurações.

- Marque *Substituir Fonte do Editor de Código* para modificar a família padrão *Fonte* e *Tamanho*.
- No grupo *Cores*, você pode:
  - selecione um *Esquema de cores*: as configurações predefinidas são `Default`, `Solarized Dark` e `Solarized Light`. Um esquema `Customizado` é acionado assim que você modifica uma cor e pode ser redefinido com a seleção de um esquema predefinido.
  - Mudar o *cor* de cada elemento na escrita do código, como as cores a serem usadas para comentários, citações, funções, plano de fundo, …

Python settings

The *Python* settings help you manage and control the behavior of the Python editors (*interactive console*, *code editor*, *project macros*, *custom expressions*, …). It can also be accessed using the Options… button from:

- A barra de ferramentas *Python console*
- O menu contextual do widget *Python console*
- E o menu contextual do editor de código.

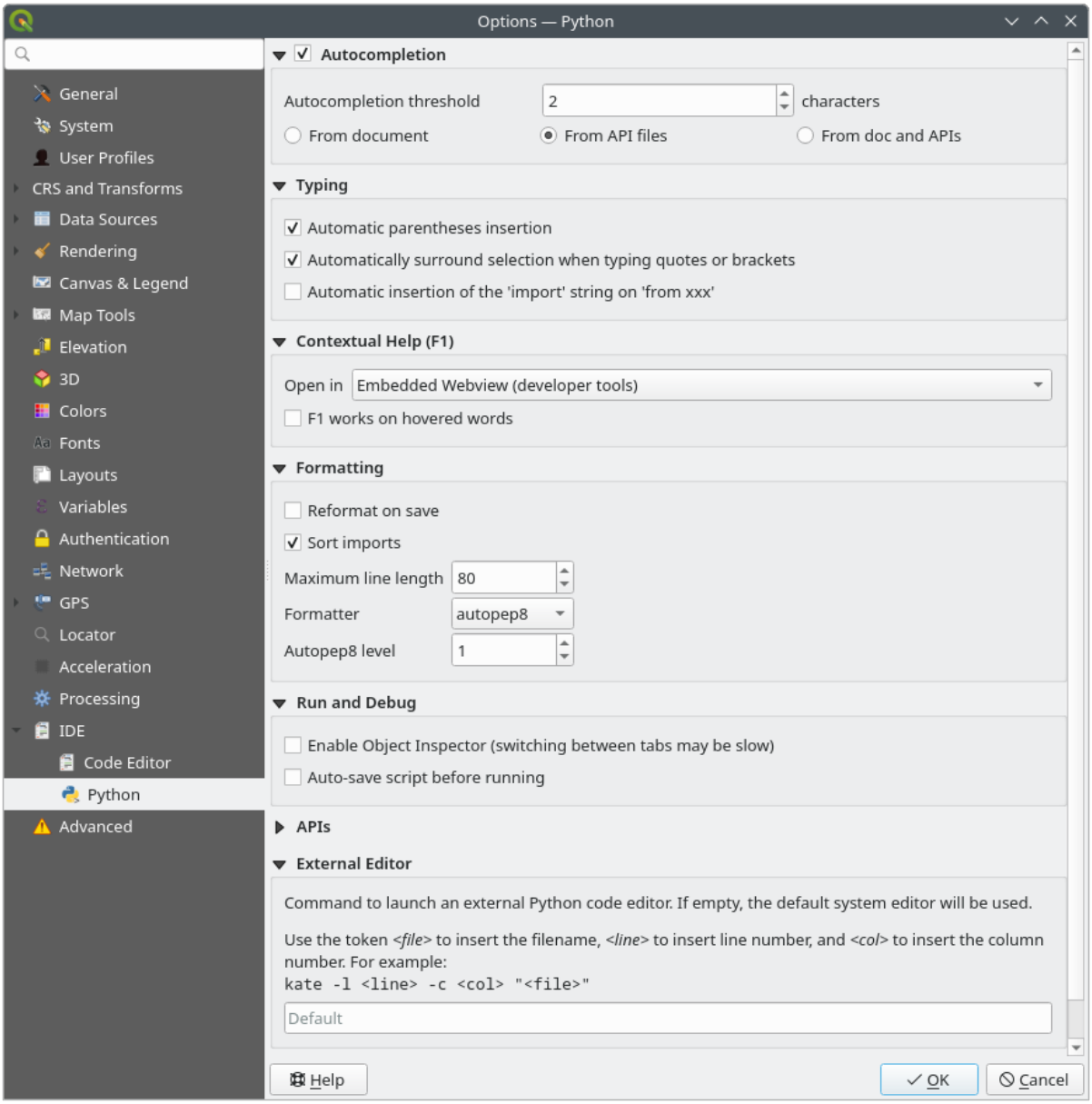


Figura5.28: Python settings

Você pode especificar:

- *Autocompletion*: Enables code completion. You can get autocompletion from the current document, the *installed API* files or both.
  - *Autocompletion threshold*: Sets the threshold for displaying the autocompletion list (in number of characters)
- Embaixo *Digitando*
  - *Automatic parentheses insertion*: When no text is selected, if an opening character (parentheses, quotes, brackets, …) is entered, inserts the matching closing character just after the cursor. Note that this behavior is disabled if the current cursor is inside a string or comment.
  - *Automatically surround selection when typing quotes or brackets*: When an opening character is entered, the selected text is enclosed in the opening/closing pair. Selection remains the same, so it is possible to quote a selected word and enclose it in parentheses just by typing `"` then `(`.

Special case for multiline selection with quotes and double quotes: selection is enclosed in triple single/double quotes.

  - ☐ *Automatic insertion of the 'import' string on 'from xxx'*: Enables insertion of 'import' when specifying imports

- *Contextual Help (F1)* provides options to control QGIS behavior when requesting help on a PyQGIS or Qt object in a Python code sample. When pressing `F1`, QGIS will open the API page (or a search results page) matching:

  1. the selected text if any OR
  2. the word at the mouse position (if ☑ *F1 works on hovered words* is enabled) OR
  3. the word at the text cursor position

  This works for both Qt and PyQGIS API documentation. The page is opened either in QGIS, as an *Embedded webview (developer tools)*, or in the *Default system web browser*.

- under *Formatting*, you can add automated tools to reformat the code you are writing:
  - ☐ *Reformat on save*: formatting is applied just before saving the script
  - ☑ *Sort imports*: sorts 'import' statements using the isort library
  - *Maximum line length*: controls how the formatter will wrap the lines, and controls the editor ruler
  - *Formatter* - supported tools are *autopep8* and *black*, with dedicated option:
    - *Autopep8 level* - more details at autopep8 advanced usage
    - ☐ *Normalize quotes*: replaces all single quotes with double quotes if possible
- Embaixo *Run and Debug*
  - ☐ *Habilitar Inspetor de Objetos (alternar entre abas pode ser lento)*
  - ☐ *Salvamento automático do script antes de executá-lo*: Salva o script automaticamente quando executado. Esta ação armazenará um arquivo temporário (no diretório temporário do sistema) que será excluído automaticamente após a execução.

Under *APIs*, you can specify which APIs your code should rely on:

- ☑ *Using preloaded APIs file*: You can choose if you would like to use the preloaded API files. By default, supported APIs are:
  - Python
  - PyQGIS
  - PyQt
  - QScintilla2
  - osgeo-gdal-ogr
  - osgeo-geos

  You can also choose to add specific API files.

- ☐ *Usando o arquivo de APIs preparado*: Se marcado, o arquivo `*.pap` escolhido será usado para completar o código. Para gerar um arquivo API preparado você deve carregar pelo menos um arquivo `*.api` e então compilá-lo clicando no botão *Compile APIs…*.

The *External Editor* group allows you to provide command line instructions to launch an external Python code editor, given a file name, a line and column number. If empty, the default system editor will be used.

5.1.21 Configurações de processamento

The *Processing* tab provides you with general settings of tools and data providers that are used in the QGIS Processing framework. More information at *Configuring the Processing Framework*.

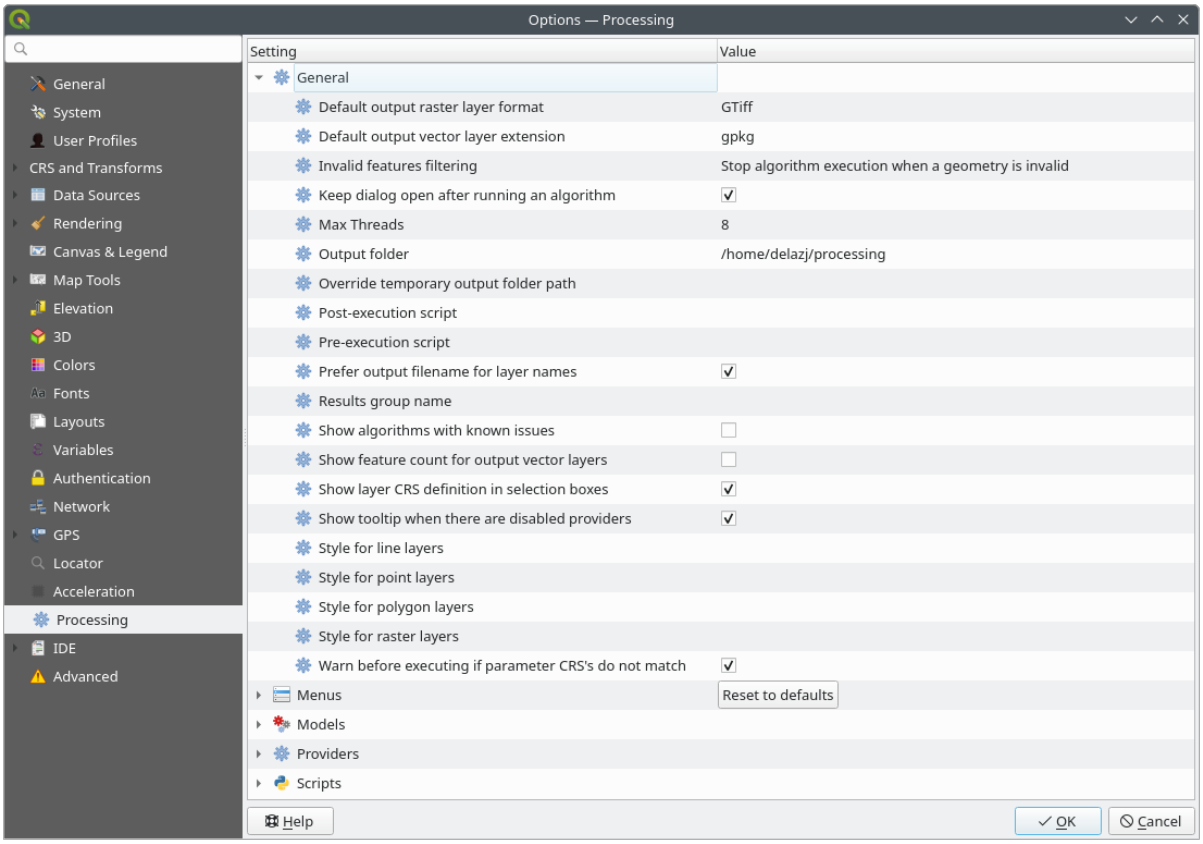


Figura5.29: Configurações de processamento

5.1.22 Configurações avançadas

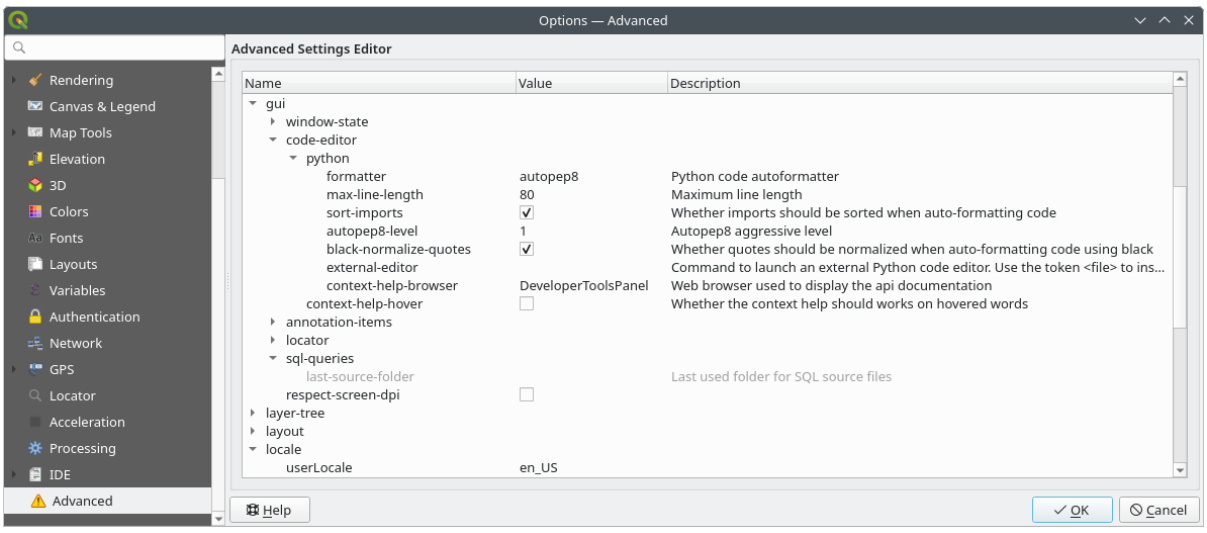


Figura5.30: Configurações avançadas

Todas as configurações relacionadas ao QGIS (UI, ferramentas, informações provedores, configurações de processamento, valores and caminhos padrões, opções de plugins, expressões, verificadores de geometria…) são salvos em um arquivo `QGIS/QGIS3.ini` debaixo do diretório ativo *perfil do usuário* . Configurações podem ser compartilhadas copiando esse arquivo para outras instalações.

From within QGIS, the *Advanced* tab offers a way to manage these settings through the *Advanced Settings Editor*. After you promise to be careful, the widget is populated with a tree of all the existing settings, and you can edit their value. Right-click over a setting or a group and you can delete it (to add a setting or group, you have to edit the `QGIS3.ini` file). Changes are automatically saved in the `QGIS3.ini` file.

Aviso

Avoid using the Advanced tab settings blindly

Be careful while modifying items in this dialog given that changes are automatically applied. Doing changes without knowledge can break your QGIS installation in various ways.

5.2 Trabalhando com Perfis de Usuário

5.2.1 The concept

The *Settings* ► *User Profiles* menu provides functions to set and access user profiles. A user profile is a unified application configuration that allows to store in a single folder:

- all the *global settings*, including locale, projections, authentication settings, color palettes, shortcuts…
- GUI configurations and *customization*
- grid files and other proj helper files installed for datum transformation
- installed *plugins* and their configurations
- project templates and history of saved project with their image preview
- *processing settings*, logs, scripts, models.

By default, a QGIS installation contains a single user profile named `default`. But you can create as many user profiles as you want:

1. Click the *Settings* ► *User Profiles* ► *New profile…* entry.
2. You'll be prompted to provide a profile name, creating a folder of the same name under `~/<UserProfiles>/` where:
   - `~` represents the **HOME** directory, which on Windows is usually something like `C:\Users\<username>`.
   - and `<UserProfiles>` represents the main profiles folder, i.e.:
     - `.local/share/QGIS/QGIS3/profiles/`
     - `%AppData%\Roaming\QGIS\QGIS3\profiles\`
     - `Library/Application Support/QGIS/QGIS3/profiles/`

   The user profile folder can be opened from within QGIS using the *Open Active Profile Folder*.
3. Uma nova instância do QGIS é iniciada, usando uma configuração limpa. Você pode então definir suas configurações personalizadas.

Se você tiver mais de um perfil em sua instalação do QGIS, o nome do perfil ativo é mostrado na barra de título do aplicativo entre colchetes.

As each user profile contains isolated settings, plugins and history they can be great for different workflows, demos, users of the same machine, or testing settings, etc. And you can switch from one to the other by selecting them in the *Settings* ► *User Profiles* menu. You can also run QGIS with a specific user profile from the *command line*.

Dica

Run QGIS under a new user profile to check for bug persistence

Bugs you may encounter with functions in QGIS can be related to leftovers in the current user profile. Running QGIS under another user profile can help you solve them or check the origin of the issue. The general advice is to launch QGIS under a new user profile, thus a cleaner configuration, and run the commands again.

If the bug prevents you to create a new user profile from within the *Settings* ► *User Profiles* menu, you can either:

- Rename in the file explorer, the "broken" user profile folder in the `QGIS3/profiles` folder and restart QGIS. A new `default` user profile will be created and executed.
- Start QGIS from the command line, using the new *profile name* argument:

```
qgis-ltr --profile newprofilename
```

5.2.2 Setting user profile

By default, QGIS opens a new session with the profile of the last closed session. This, among other settings, can be customized in *Settings* ► *Options* ► *User Profiles* tab:

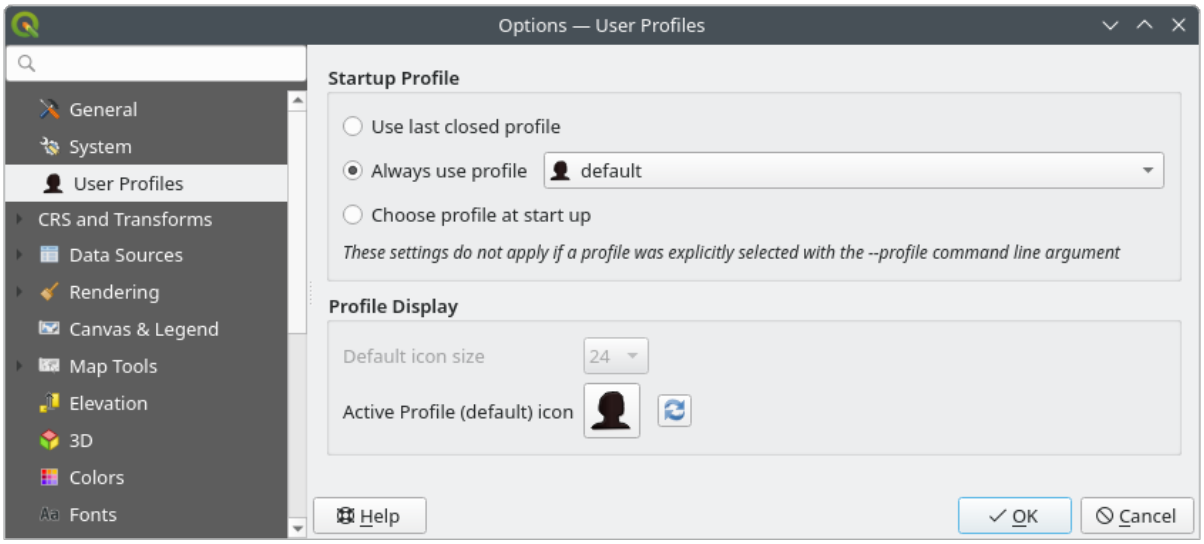


Figura5.31: User profiles settings

- *Startup profile*: indicates the user profile to use when starting a QGIS session. It can be:
  - *Use last closed profile*
  - a specific user profile to select from *Always use profile* drop-down menu
  - *Choose profile at start up*: Opens the *User Profile Selector* dialog listing available user profiles. Double-click an entry or select a profile and press *OK* to use that user profile for the starting session. It is also possible to *Add new profile* to the list: a profile is created from scratch and automatically run with the opening session.
- Under *Profile display*, you can set:

- The size of icons to use when selecting a profile from the *User Profile Selector* dialog.
- A specific icon to display next to the current profile in the *Settings* ► *User profiles* menu or the *User Profile Selector* dialog. Press Reset profile icon to remove custom modifications.

5.3 Propriedades do Projeto

In the properties window for the project under *Project* ► *Project Properties*, you can set project-specific options. The project-specific options overwrite their equivalent in the *Options* dialog described above.

5.3.1 Propriedades Gerais

Na aba *Geral*, a aba *Configurações Gerais* permite a você:

- veja a localização do arquivo do projeto
- set the folder for the project home (available in the *Project home* item of the *Browser* panel). The path can be relative to the folder of the project file (type it in) or absolute. The project home can be used for storing data and other content that is useful for the project. Convenient when dataset and project files are not stored at the same place. If not filled, the *Project home* defaults to the project file folder.
- give a title to the project beside the project file path
- escolha a cor a ser usada para as feições quando elas são selecionadas
- choose the background color: the color to use for the map canvas
- set whether the path to layers in the project should be saved as absolute (full) or as relative to the project file location. You may prefer relative path when both layers and project files can be moved or shared or if the project is accessed from computers on different platforms.
- choose to avoid artifacts when project is rendered as map tiles. Note that checking this option can lead to performance degradation.
- *Remember attribute tables windows and docks between sessions*: If checked for a project, then any opened attribute tables will be saved into the project and immediately restored when loading that project. This can improve workflows when you constructed a project with a particular set of attribute table configurations for your requirements, and re-setting up these attribute tables is a hassle.

Calculating areas and distances is a common need in GIS. However, these values are really tied to the underlying projection settings. The *Measurements* frame lets you control these parameters. You can indeed choose:

- the *Ellipsoid*, on which distance, area and angle calculations are entirely based; it can be:
  - **None/Planimetric**: returned values are in this case cartesian measurements. This option can be set as default for new projects from the *Settings* ► *Options* ► *CRS Handling* menu
  - a **Custom** one: you'll need to set values of the semi-major and semi-minor axes.
  - or an existing one from a predefined list (Clarke 1866, Clarke 1880 IGN, New International 1967, WGS 84…).
- the *Units for distance measurements* for length and perimeter, and the *Units for area measurements*. These settings default to their corresponding *global options* and can be manually changed for the current project. They are used in:
  - Attribute table field update bar
  - Field calculator calculations
  - *Identify tool* derived length, perimeter and area values
  - *measure dialog*

  - *scale bar decoration*

> **Nota**
>
> To ensure measurement units (for distance and area) always reflect the units of the current project's CRS, select *Map Units* for these options.

- the *Scale calculation method*: when working on large map extents, especially with geographic CRS, the scale bar may not reflect actual distance measurement depending on the latitude at which the calculation is done. This setting helps you define the area you want the calculation to be taken from:
  - *Average top, middle and bottom scales*
  - *Calculate along top of map*
  - *Calculate along middle of map* (the default)
  - *Calculate along bottom of map*
  - *Always calculate at equator*, regardless of the actual visible map extent. It can be used to provide a consistent, static scale for maps in geographic reference systems, regardless of the latitudes actually visible in the map (permitting consistent appearance of these maps when rendering relies on scale based visibility or calculations). Otherwise a project in e.g. `EPSG:4326` which uses scale based visibility of layers and symbols will see layers and features "randomly" disappear as the map is panned, even though you have not zoomed in or out of the map.

    > **Nota**
    >
    > This method is only applicable when calculating scales with a degree based reference system, and while it ensures that the scale remains constant and does not change as the map is panned, it will calculate misleading scales when the map extent is not close to the equator.

  Changing the scale calculation method has the following impacts:
  - New layout scale bars will default to the project's scale calculation method
  - The scale calculations for the status bar widget and map renders will be changed (including flow on impacts like the value of `@map_scale` variable, scale based visibility of layers and symbols)
  - This also affects QGIS server map rendering
  - Processing algorithms which render maps will respect the project's scale calculation method

> **Nota**
>
> Symbology sizes in map units are NOT affected by this setting.

The *Coordinate and Bearing display* allows you to customize the display of:

- the coordinates shown in the *Coordinates* box on QGIS status bar and in the *Derived* section of the Identify features tool's results
- the bearing value displayed in the status bar for the map canvas panning direction and by the Measure bearing tool.

Available parameters are:

- *Display coordinates using* either:
  - `Map Units`, based on the project CRS

  - `Map Geographic (degrees)`: based on the project CRS if it is of geographic type, otherwise uses its associated geographic CRS. This is helpful e.g., for non-earth celestial bodies.
  - or `Custom Projection Units`: allows to rely on any CRS you desire for coordinates display
- In the *Coordinate CRS* option, you can view or define the CRS to use depending on your display mode.
- *Coordinate format*: you can configure it as `Decimal Degrees`, `Degrees, Minutes` or `Degrees, Minutes, Seconds`, and whether it should display:
  - ☐ *Show directional suffix*
  - ☐ *Show leading zeros for minutes and seconds*
  - ☐ *Show leading zeros for degrees*
  - ☐ *Show trailing zeros*
- *Coordinate precision*: the number of decimal places can be automatic (derived from the type of CRS) or set manually
- *Coordinate order*: you can opt to display the coordinates in the native order of the CRS (`Default`) or switch it to either `Easting, Northing (Longitude, Latitude)` or `Northing, Easting (Latitude, Longitude)` order
- *Bearing format* possible values are `0 to 180°, with E/W suffix`, `-180 to +180°` or `0 to 360°`. The number of *Decimal places* as well as whether to *Show trailing zeros* can be set.

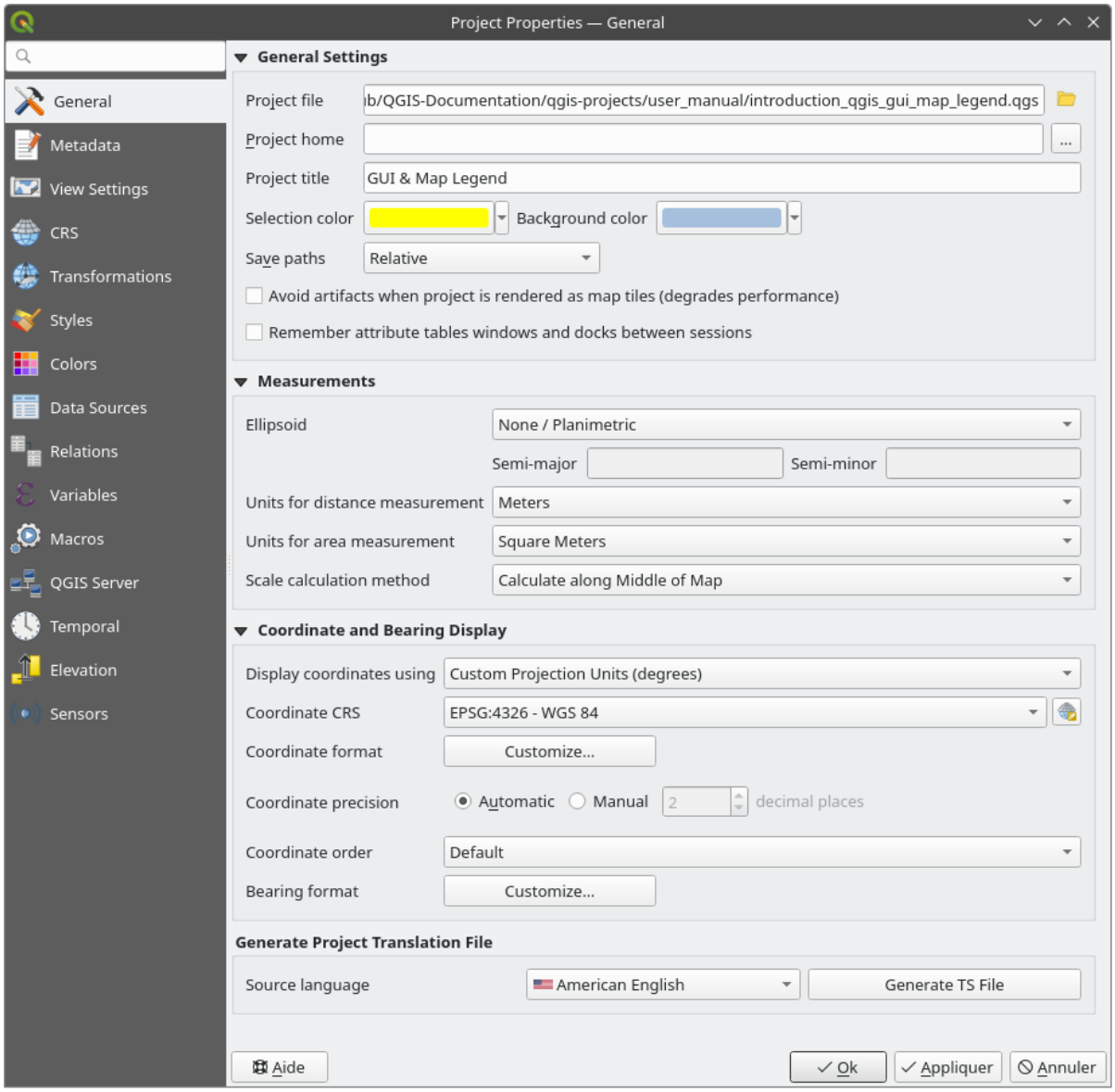


Figura5.32: General tab of the Project Properties dialog

5.3.2 Propriedades dos Metadados

The *Metadata* tab provides you with options to create and edit a metadata report on your project. See *Metadados* for more information.

5.3.3 View Settings

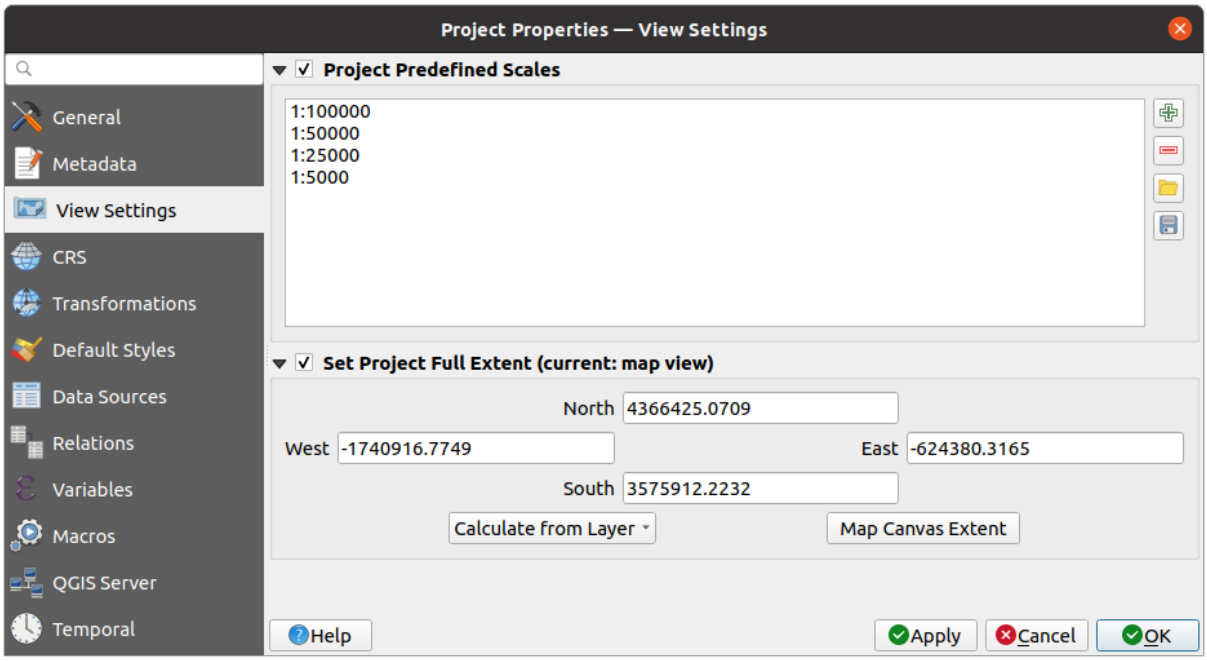


Figura5.33: View Settings tab of the Project Properties dialog

The *View Settings* tab provides means to control the project map canvas. You can:

- set *Project predefined scales*: the list of scales to display in scale-related drop-down widgets, such as the status bar *Scale*, the visibility scales selector or secondary 2D map view settings,… in replacement of the global *predefined scales*.
- *Set Project full Extent*: this extent will be used instead of the extent of all layers when zooming to full map extent ( ). It's useful when a project contains web layers/national layers/global layers yet the actual area of interest for the project is a smaller geographic area. The project full extent coordinates can be set with the *extent selector* widget.

5.3.4 CRS Properties

> **Nota**
>
> For more information on how QGIS handles project projection, please read the dedicated section at *Trabalhando com Projeções*.

The *CRS* tab helps you set the coordinate reference system to use in this project. It can be:

- *No CRS (or unknown/non-Earth projection)*: layers are drawn based on their raw coordinates
- or an existing coordinate reference system that can be *geographic*, *projected* or *user-defined*. Layers added to the project are translated on-the-fly to this CRS in order to overlay them regardless their original CRS.

5.3.5 Transformations Properties

The *Transformations* tab helps you control the layers reprojection settings by configuring the datum transformation preferences to apply in the current project. As usual, these override any corresponding global settings. See *Transformações de Datum* for more details.

5.3.6 Styles Properties

Under *Styles* tab, you can configure symbols and colors inherent to the project, allowing to safely share the project among different machines.

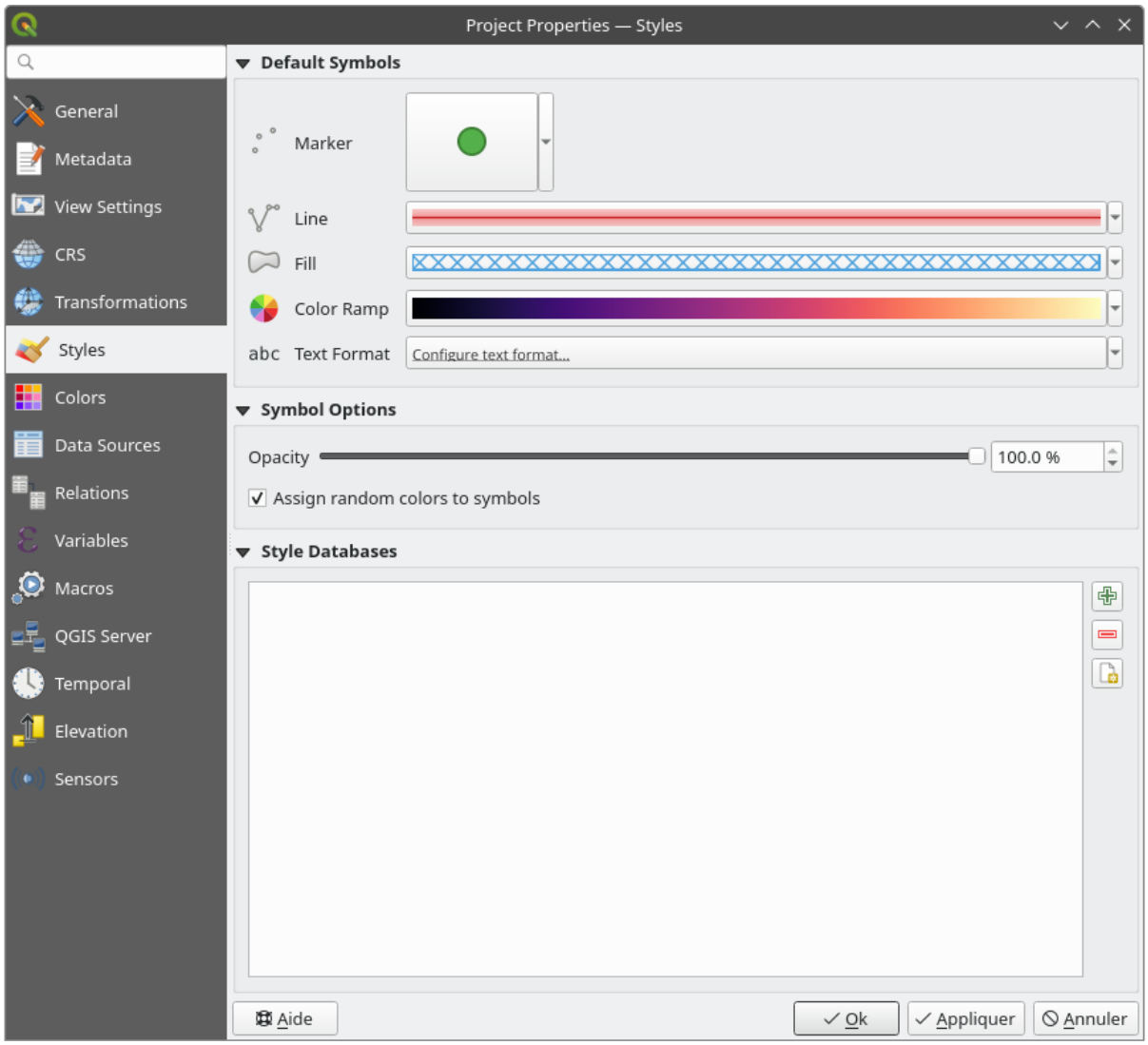


Figura5.34: Styles tab

Default symbols

The *Default Symbols* group lets you control how new layers will be drawn in the project when they do not have an existing `.qml` style defined. You can set *Marker*, *Line*, *Fill* to apply depending on the layer geometry type as well as default *Color Ramp* and *Text Format* (e.g., when enabling labeling). Any of these items can be reset using the *Clear* entry from the corresponding drop-down widget.

Symbol options

In the *Symbol options* group, you can:

- Apply a default *Opacity* to new layers
- *Assign random colors to symbols*, modifying the symbols fill colors, hence avoiding same rendering for all layers.

Style Database

A style database in QGIS is a structured repository designed to store symbols, text formats, and other styling elements. It serves as a centralized location where you can organize and manage your symbology resources efficiently. You can create a dedicated style database for a specific client, housing symbols tailored to that client's need. This ensures a clean and organized approach, without cluttering the default style database. In multi-user environments, it's possible to store project-specific styles in a shared location. By linking a project to these styles, all users within the project gain access to common symbology, eliminating the need for manual import or updates to individual local style database. Storing symbols in a project file offers a practical solution, preventing users from overcrowding their global style database with project-specific symbology. This approach guarantees that other users working on the same project immediately have access to all the necessary symbology upon loading the project.

In the *Style Database* section you can choose to *Add* or *Remove* style database or you can *Create new style database*. When you add or remove a style database in this section, the changes will be automatically reflected in the *Symbology Properties*.

5.3.7 Colors properties

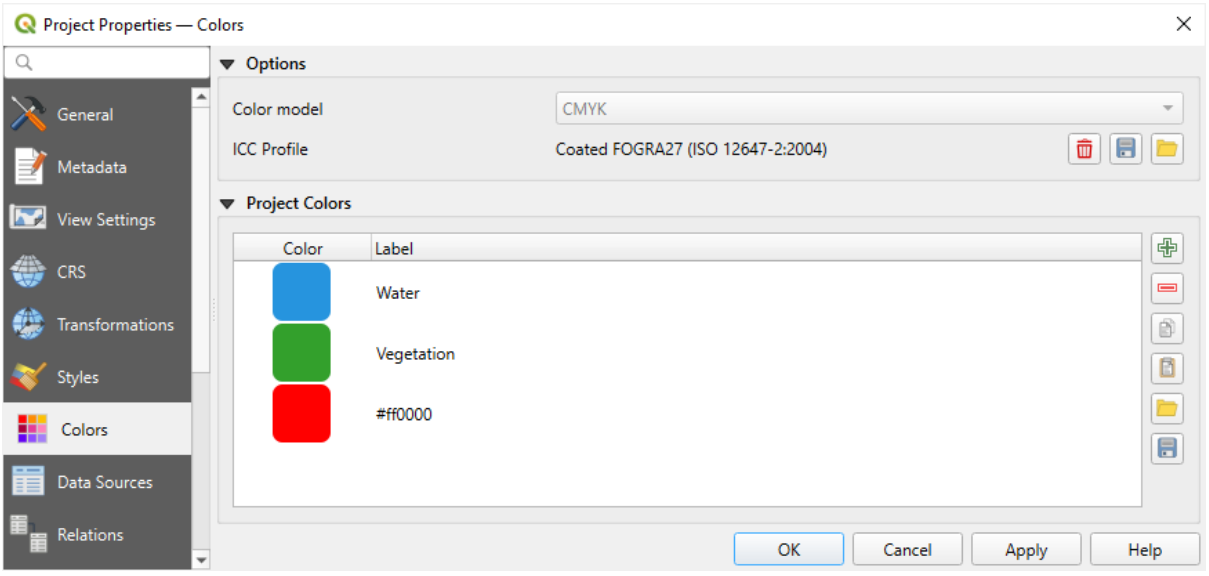


Figura5.35: Colors tab

In the *Colors* tab, you can select the *Color model*, i.e., whether selecting a color should by default be defined as `RGB` or `CMYK`. Any color defined in a different color model than the one specified here will be converted to this color model when exporting a layout.

Additionally, an option is available for selecting an *ICC Profile* which allows you to *Load* a specific ICC profile that defines the color space for your project. After you load the ICC profile, you can choose to *Save ICC Profile* to a file on disk or to *Remove selected ICC Profile*. Once applied, QGIS will enforce the selected color model to ensure consistent color representation.

Nota

Note that the *ICC Profile* option will only be visible if QGIS is built with Qt 6.8.0 or higher.

There is also an additional section where you can define specific colors for the running project. Like the *global colors*, you can:

- *Adicionar* ou *Remover* cor
- *Copy* or *Paste* color
- *Importar* ou *Exportar* o conjunto de cores do/para `.gpl` arquivo.

Double-click a color in the list to tweak or replace it in the *Color Selector* dialog. You can also rename it by double-clicking in the *Label* column.

These colors are identified as *Project colors* and listed as part of *color widgets*.

Dica

Use project colors to quickly assign and update color widgets

Project colors can be refered to using their label and the color widgets they are used in are bound to them. This means that instead of repeatedly setting the same color for many properties and, to avoid a cumbersome update you can:

1. Definir a cor como uma cor do projeto
2. Click the *data defined override widget* next to the color property you want to set
3. Hover over the *Color* menu and select the project color. The property is then assigned the expression `project_color('color_label')` and the color widget reflects that color.
4. Repita os passos 2 e 3 o quanto for necessário
5. Atualize a cor do projeto uma vez e a mudança será refletida EM TODOS OS LUGARES em que estiver em uso.

5.3.8 Data Sources Properties

Na tabela :guilabel:`Fonte de Dados', você pode:

- *Transaction mode*, defines how edits are sent to the data provider:
  - *Local Edit Buffer*: edits are buffered locally and sent to the provider when toggling layer editing mode or clicking *Save edits*.
  - *Automatic Transaction Groups*: on supported datasources (postgres and geopackage databases) the edit state of all tables that originate from the same database are synchronized and executed in a server side

transaction. Also, instead of buffering edit changes locally, they are directly sent to a transaction in the database which gets committed when toggling layer editing mode or clicking *Save edits*.

- *Buffered Transaction Groups*: all editable layers, disregarding from which provider, are toggled synchronously and all edits are saved in a local edit buffer. Saving changes is executed within a single transaction on all layers (per provider).

Note that you can change this option only if no layer is being edited in the project.

- *Evaluate default values on provider side*: When adding new features in a PostgreSQL table, fields with default value constraint are evaluated and populated at the form opening, and not at the commit moment. This means that instead of an expression like `nextval('serial')`, the field in the *Add Feature* form will display expected value (e.g., `25`).
- *Remember editable layer status between sessions*: makes sure that all layers that are editable in a project will be remembered as such when saving the project, as well as making sure that those layers are immediately made editable whenever the project is restored.
- Configure the *Layers Capabilities*, i.e.:
  - Set (or disable) which layers are `identifiable`, i.e., will respond to the *identify tool*. By default, layers are set queryable.
  - Set whether a layer should appear as `read-only`, meaning that it can not be edited by the user, regardless of the data provider's capabilities. Although this is a weak protection, it remains a quick and handy configuration to avoid end-users modifying data when working with file-based layers.
  - Define which layers are `searchable`, i.e., could be queried using the *locator widget*. By default, layers are set searchable.
  - Define which layers are defined as `required`. Checked layers in this list are protected from inadvertent removal from the project.
  - Define which layers are `private`, i.e., hidden from the *Layers* panel. This is meant for accessory layers (basemap, join, lookups for value-relations, most probably aspatial layers, ...) that you still need in a project but you don't want them to pollute the legend tree and other layer selection tools. If set visible, they are still displayed in the map canvas and rendered in the print layout legend. Use the *Filter legend* ► *Show private layers* option in the *Layers* panel top toolbar to temporarily turned them on for any interaction.

  The *Layers Capabilities* table provides some convenient tools to:
  - Select multiple cells and press *Toggle Selection* to have them change their checkbox state;
  - *Show spatial layers only*, filtering out non-spatial layers from the layers list;
  - *Filter layers...* and quickly find a particular layer to configure.
- Under the *Advanced Settings* group, you can select *Trust project when data source has no metadata*: To speed up project loading by skipping data checks. Useful in QGIS Server context or in projects with huge database views/materialized views. The extent of layers will be read from the QGIS project file (instead of data sources) and when using the PostgreSQL provider the primary key unicity will not be checked for views and materialized views.

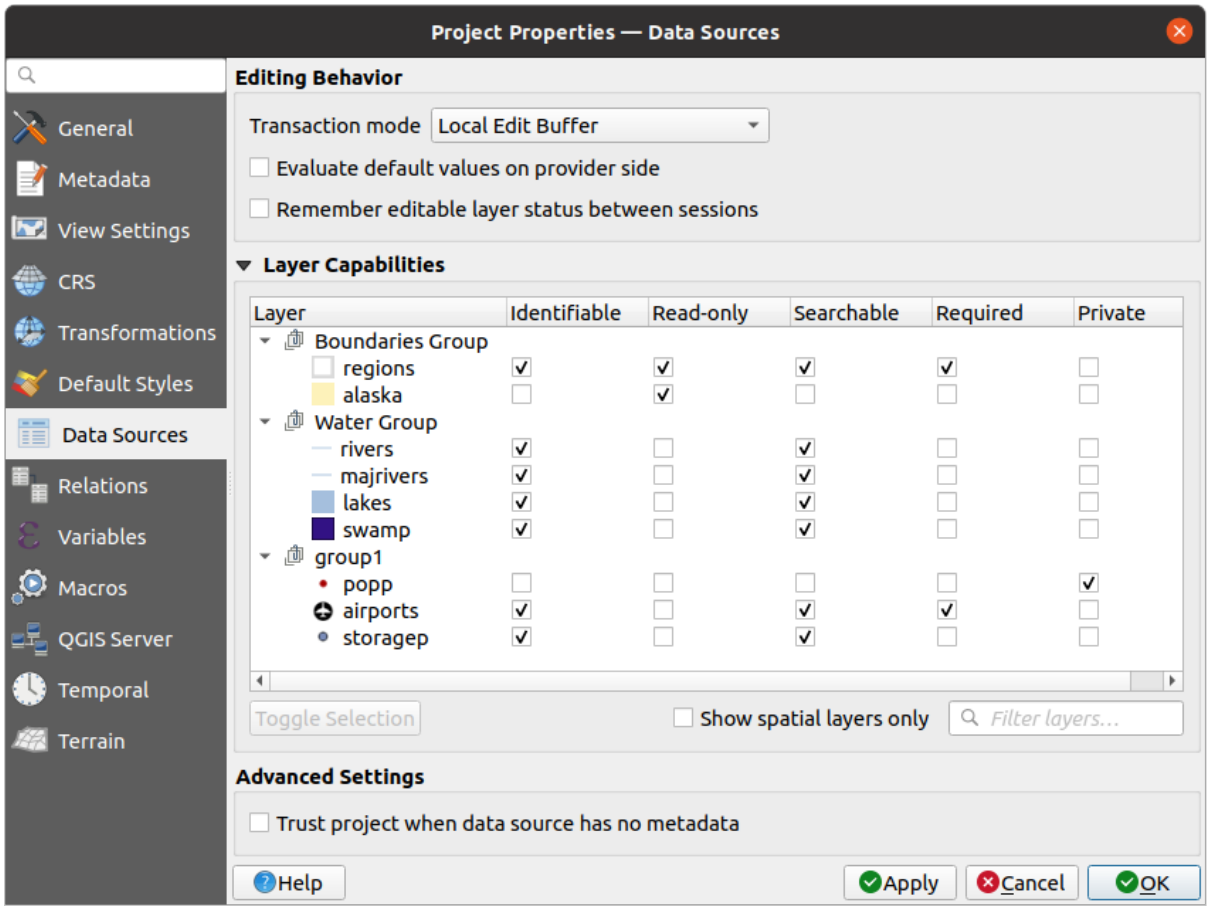


Figura5.36: Data Sources tab

5.3.9 Propriedades de Relações

The *Relations* tab is used to define relations between layers. The relations can be of one to one, many to many or polymorphic type. They are defined in the project properties dialog. Once relations exist for a layer, a new user interface element in the form view (e.g., when identifying a feature and opening its form) will list the related entities. This provides a powerful way to express e.g., the inspection history on a length of pipeline or road segment. You can find out more about relations support in section *Setting relations between multiple layers*.

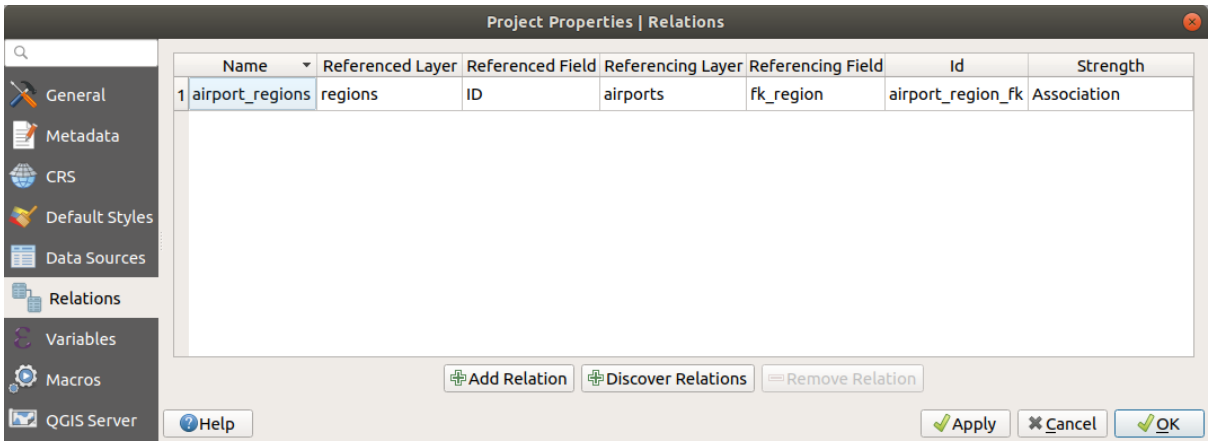


Figura5.37: Aba de relações

5.3.10 Propriedades das Variáveis

The *Variables* tab lists all the variables available at the project's level (which includes all global variables). Besides, it also allows the user to manage project-level variables. Click the button to add a new custom project-level variable. Likewise, select a custom project-level variable from the list and click the button to remove it. More information on variables usage in the General Tools *Armazenando valores em Variáveis* section.

5.3.11 Propriedades das Macros

A aba *Macros* é utilizada para editar macros Python para projetos. Atualmente, apenas três macros estão disponíveis: `openProject()`, `saveProject()` and `closeProject()`.

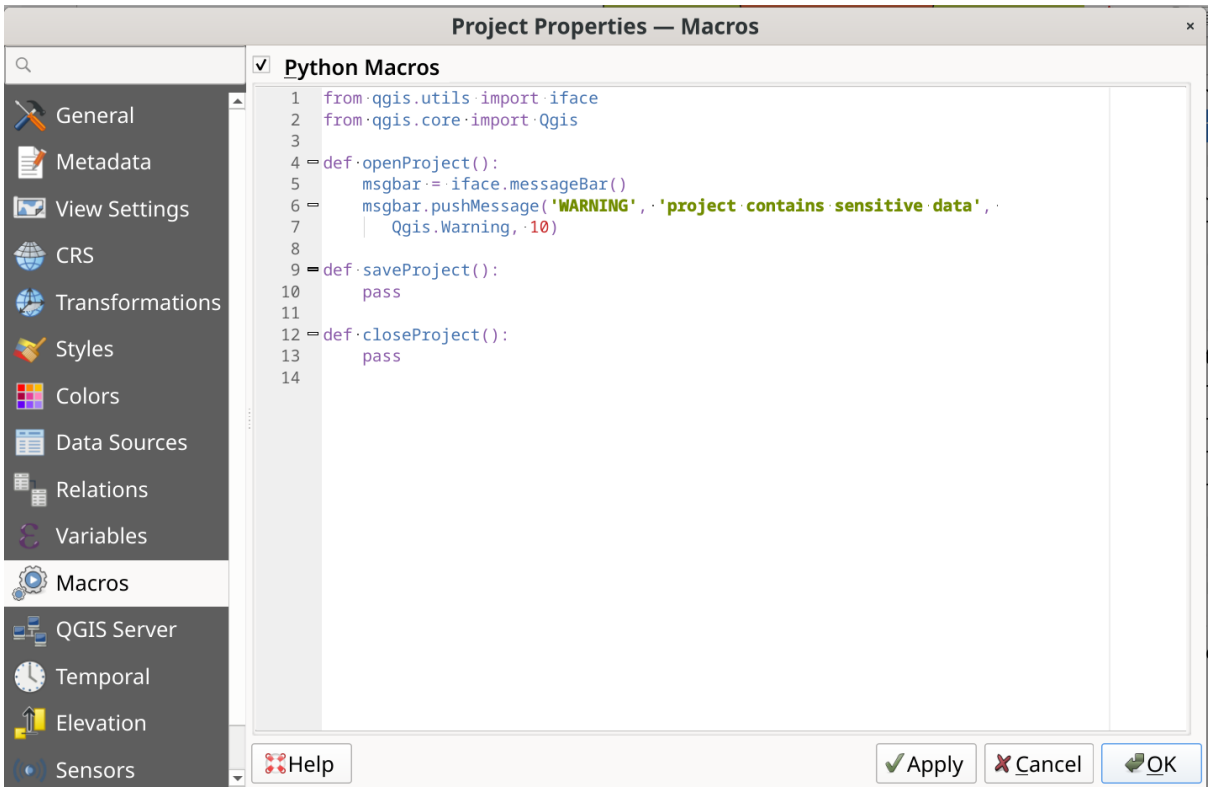


Figura5.38: Macro settings

5.3.12 Propriedades do Servidor QGIS

The *QGIS Server* tab allows you to configure your project in order to publish it online. Here you can define information about the QGIS Server WMS and WFS capabilities, extent and CRS restrictions. More information available in section Creatingwmsfromproject and subsequent.

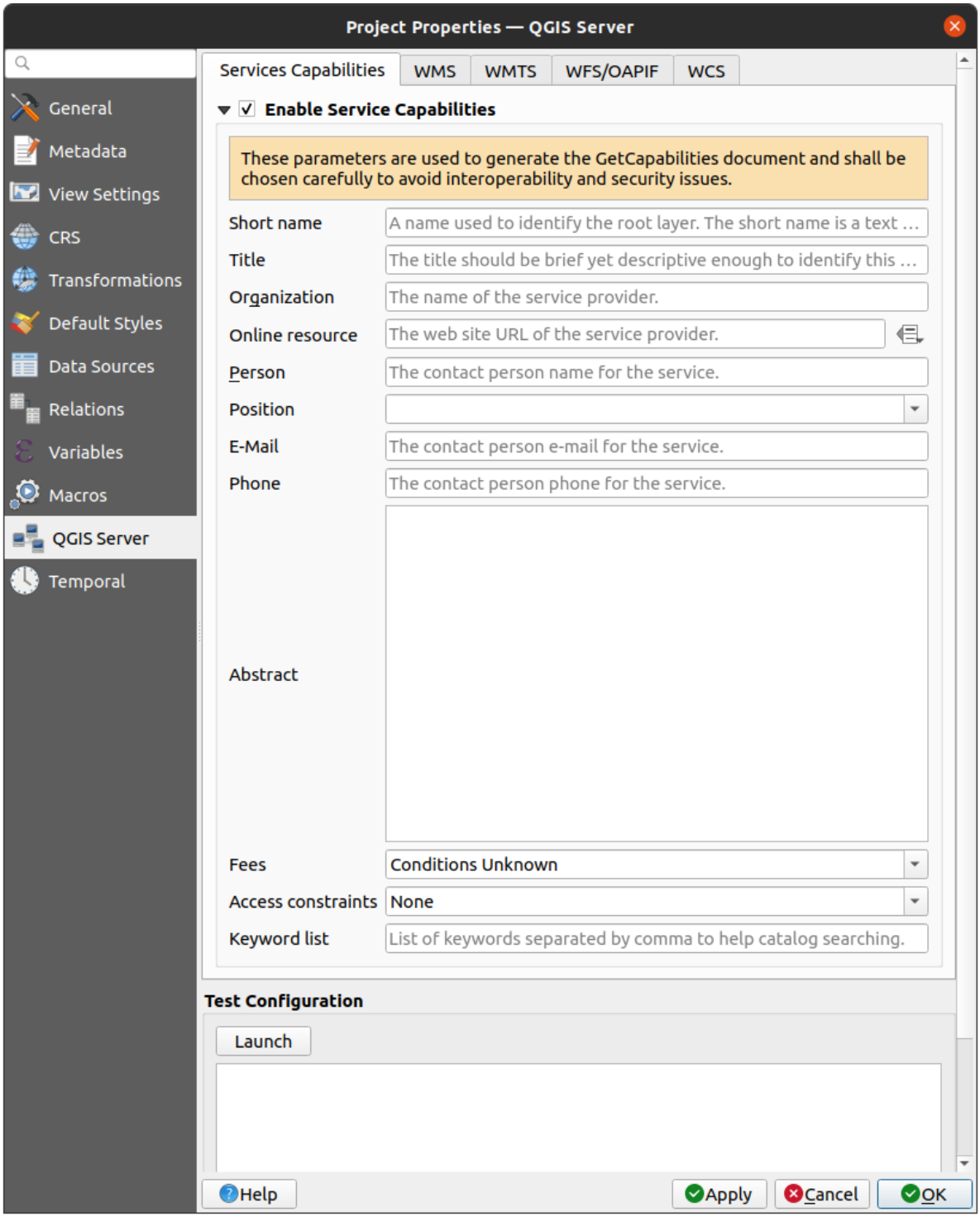


Figura5.39: QGIS Server settings

5.3.13 Propriedades Temporais

The *Temporal* tab is used to set the temporal range of your project, either by using manual *Start date* and *End date* inputs or by calculating it from the current project temporal layers. The project time range can then be used in the *Temporal controller panel* to manage the map canvas *temporal navigation*.

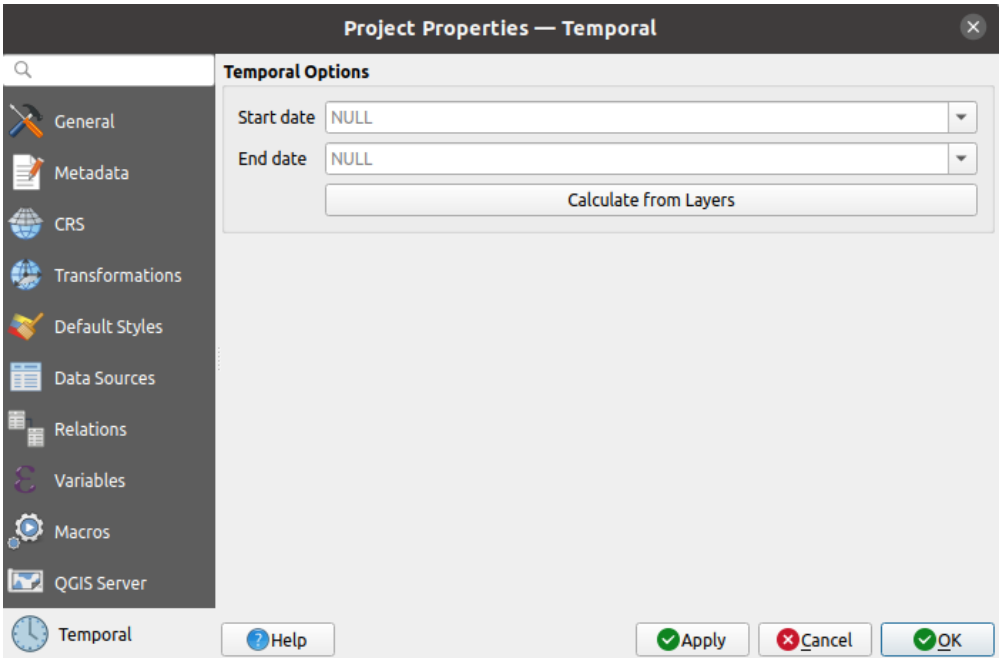


Figura5.40: Project Temporal tab

5.3.14 Elevation Properties

The *Elevation* tab helps you configure default settings for the terrain and elevation. When any new *3d map* is created in the project, the map will default to using the same terrain settings as are defined for the project. The project elevation settings will also be respected by the *elevation profile* tool.

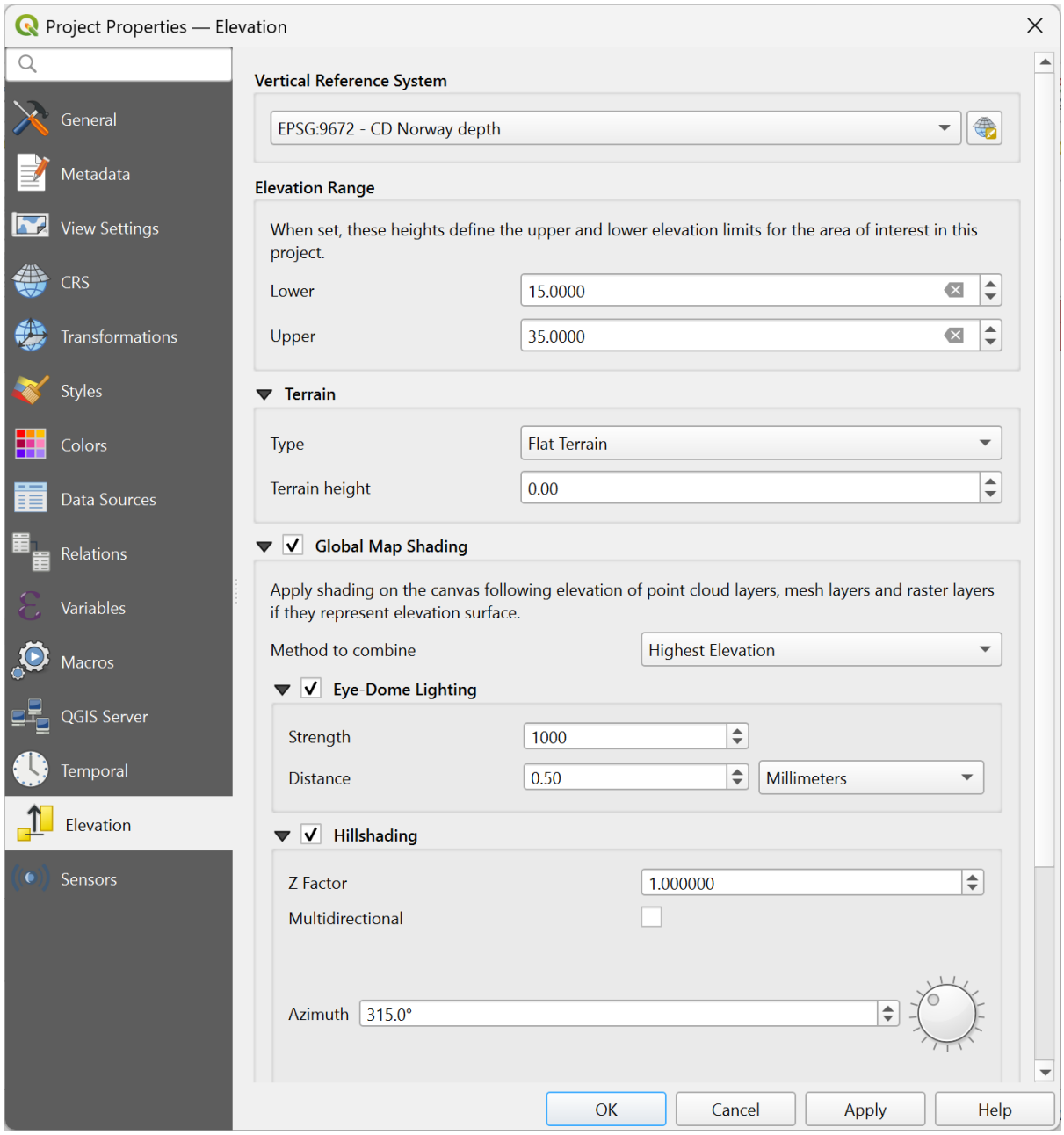


Figura5.41: Project Elevation tab

- *Vertical Reference System*: If the CRS of your project is compound one (including a Z dimension), then the vertical CRS used for the project will be the vertical component of the project CRS. If your project CRS is horizontal, then you can select a specific vertical CRS by clicking on the Select CRS.
- *Elevation Range*: helps you define the *Lower* and *Upper* limits for the *elevation controller* in the project.
- *Terrain* and elevation options are available for:
  - *Flat terrain* with *Terrain height* setting
  - *DEM (Raster Layer)*: with setting for defining the *Raster layer*, a *Vertical scale* factor to apply to band values and a vertical *Offset*
  - *Mesh*: with setting for defining the *Mesh layer*, a *Vertical scale* factor to apply to vertices Z value and a vertical *Offset*

  These settings can be overwritten from the 3D map *configuration dialog*.

- With *Global map shading* settings, you apply a global shading effect to the map, based on the elevation of all the active layers that have elevation properties enabled, including:
  - *raster layers*
  - *mesh layers*
  - *point cloud layers*

  Method to combine: the elevation value at any particular position for all the relevant layers are combined, and the considered elevation is chosen depending on one of these methods:
  - *Highest elevation* value
  - *Based on layer's order*: the elevation on the topmost layer in the layer tree is considered.

  Supported shading options are:
  - *Eye-dome lighting*: applies shading effects to the map canvas for a better depth rendering. Following parameters can be controlled:
    - *Strength*: increases the contrast, allowing for better depth perception
    - *Distance*: represents the distance of the used pixels off the center pixel and has the effect of making edges thicker.
  - *Hillshading*, shaping some reliefs on the map using shading (levels of gray):
    - *Z Factor*: Scaling factor for the elevation value
    - *Multidirectional*: Specify if multidirectional hillshading is to be used
    - *Azimuth*: The azimuth of the light source
    - *Altitude*: The elevation angle of the light source

> **Nota**
>
> A shortcut to *Global map shading* properties is available through the *Layer Styling* panel.

5.3.15 Sensors Properties

The *Sensors* tab is used to configure sensors and toggle their connection status. When active, sensors will passively collect data in the background and make available their latest data to expressions and python scripts.

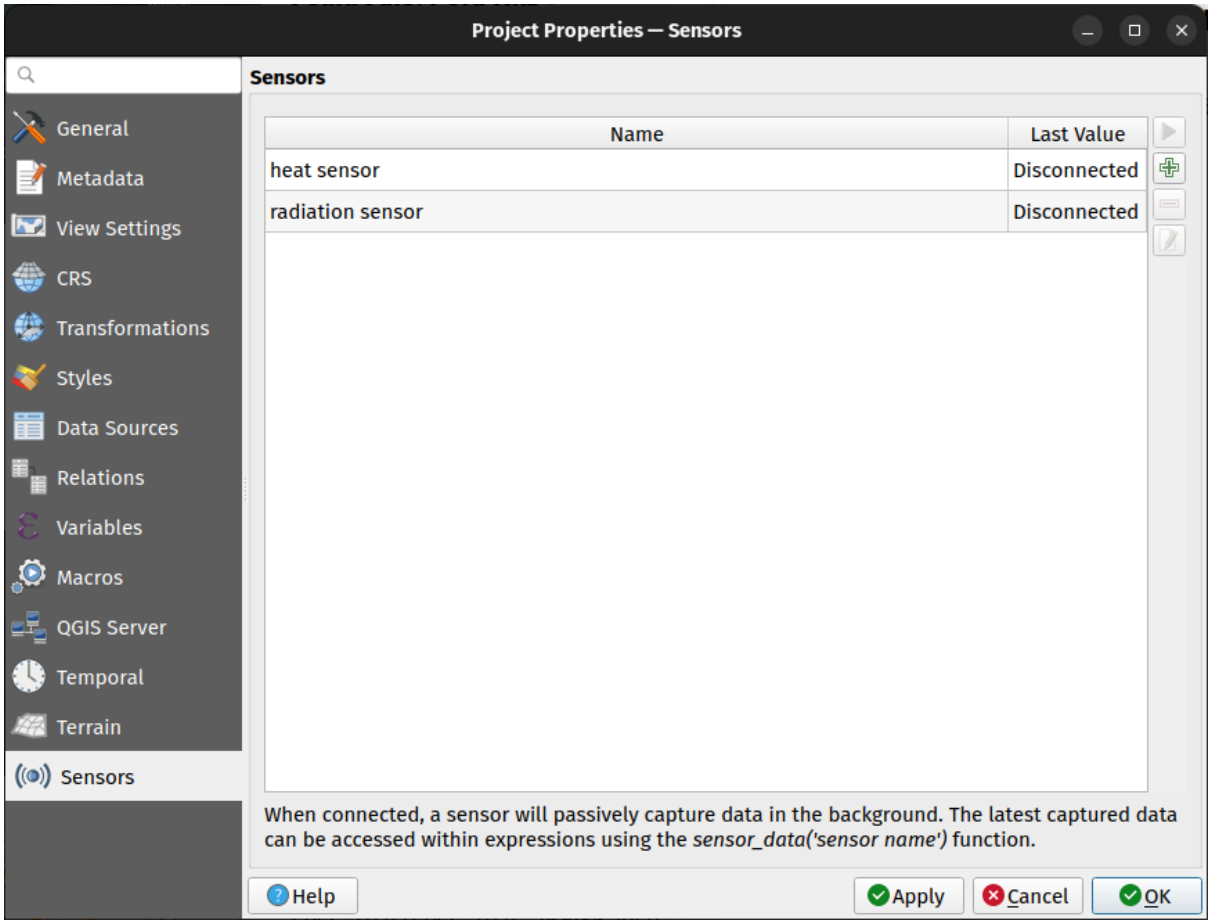


Figura5.42: Project Sensors tab

To add a new sensor, click on the button. A settings sub-panel will open and allow you to configure:

- the *Sensor name*: used to retrieve sensor values in expressions and python scripts;
- the *Sensor type*: TCP, UDP, serial port, etc.; and
- additional type-specific details (e.g., host name and port).

  For serial port sensors, comboboxes with the system's available serial ports and connection baud rate are available for their selection. A *Data frame delimiter* can be used to parse returned data. In case multiple data frames are found in the last chunk of data pushed by the serial port, all the data frames are returned. You can then use an expression to further split the returned sensor last data value into individual frames (e.g., using *string_to_array* function).

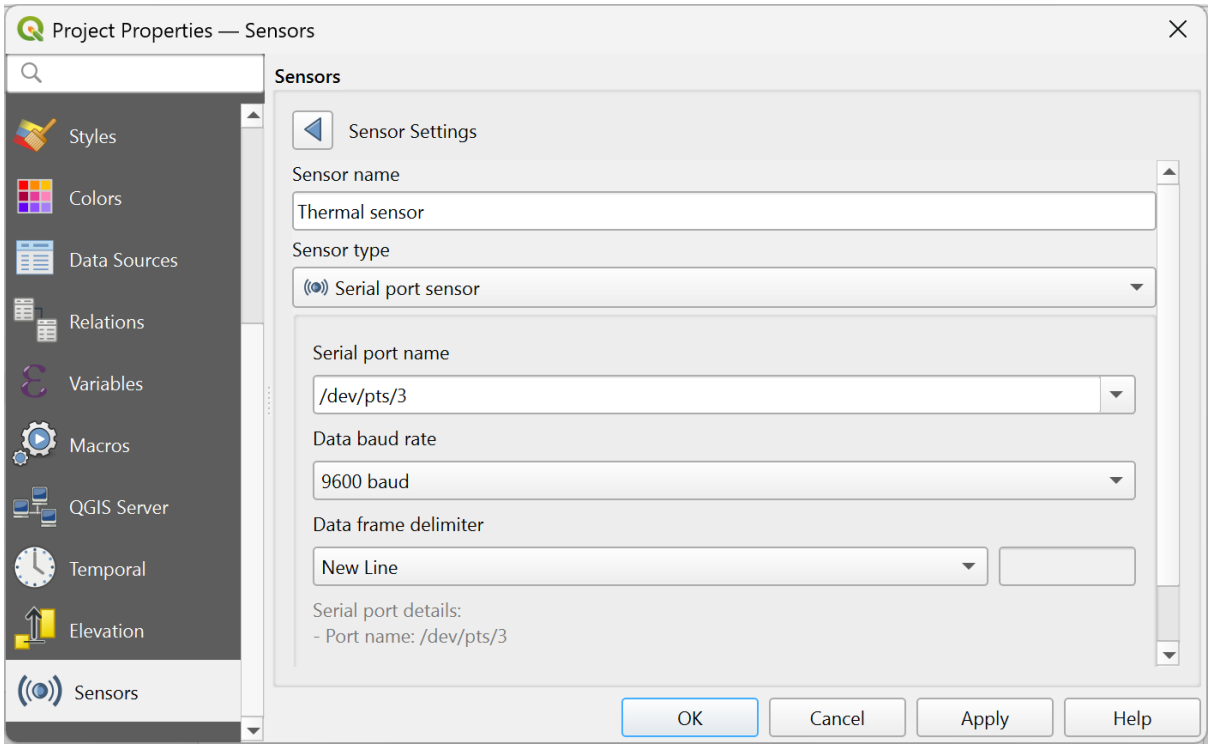


Figura5.43: Sensor Settings sub-panel

Once a sensor is configured, you can use the Start button to connect the sensor. Once active, the last collected data will be displayed in the sensors table's *Last value* column.

5.4 Personalização

The *Customization* dialog lets you (de)activate almost every element in the QGIS user interface. This can be very useful if you want to provide your end-users with a 'light' version of QGIS, containing only the icons, menus or panels they need.

Nota

Antes que suas mudanças sejam aplicadas, você precisa reiniciar o QGIS.

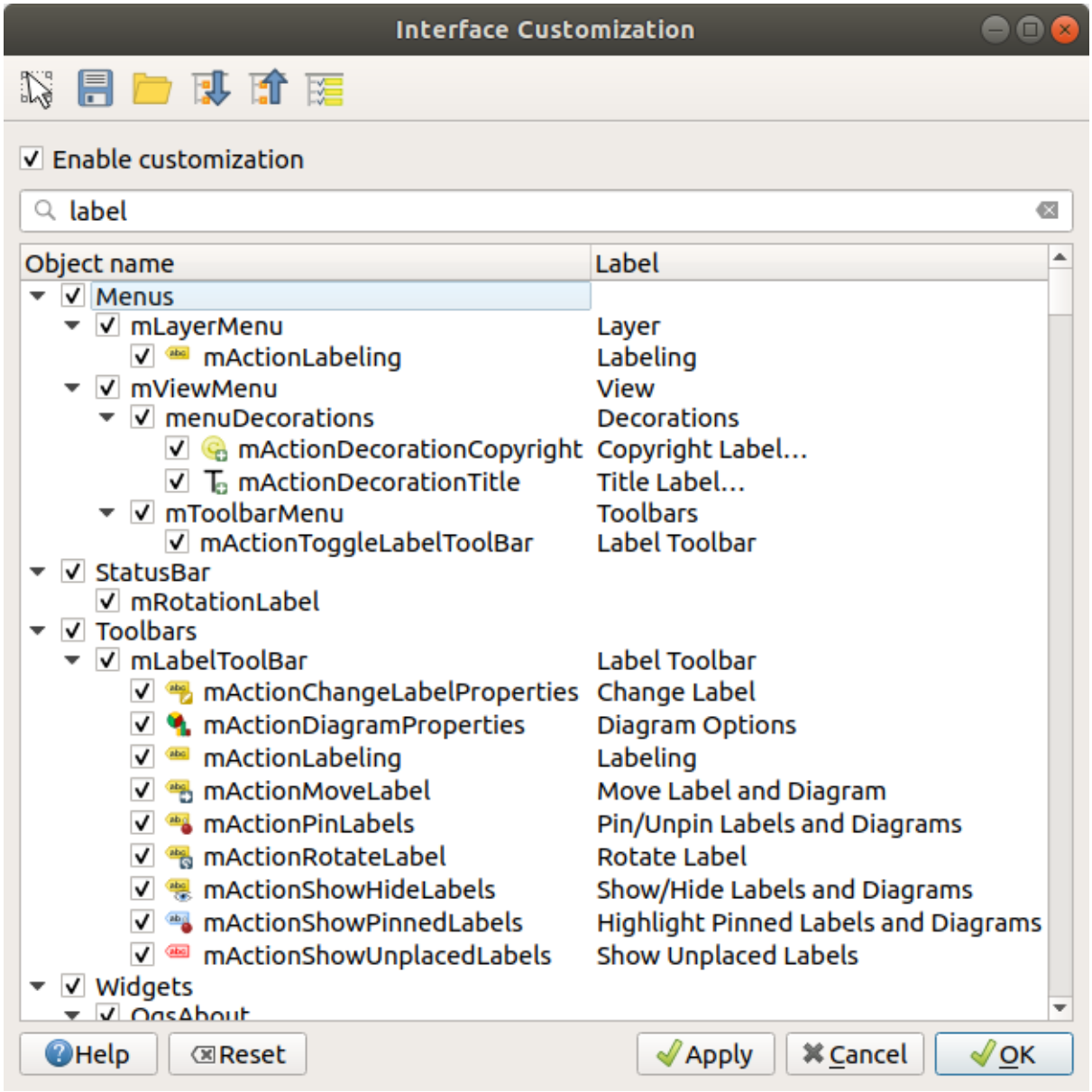


Figura5.44: The Customization dialog

Ticking the *Enable customization* checkbox is the first step on the way to QGIS customization. This enables the toolbar and the widget panel from which you can uncheck and thus disable some GUI items.

O item configurável pode ser:

- a **Menu** or some of its sub-menus from the *Barra de Menu*
- a whole **Panel** (see *Painéis e Barras de Ferramentas*)
- the **Status bar** described in *Barra de Status* or some of its items
- a **Toolbar**: the whole bar or some of its icons
- or any **widget** from any dialog in QGIS: label, button, combobox…

With Switch to catching widgets in main application, you can click on an item in QGIS interface that you want to be hidden and QGIS automatically unchecks the corresponding entry in the Customization dialog. You can also use the *Search* box to find items by their name or label.

Once you setup your configuration, click *Apply* or *OK* to validate your changes. This configuration becomes the one used by default by QGIS at the next startup.

The modifications can also be saved in a `.ini` file using Save To File button. This is a handy way to share a common QGIS interface among multiple users. Just click on Load from File from the destination computer in order to import the `.ini` file. You can also run *command line tools* and save various setups for different use cases as well.

Dica

Restaure facilmente o QGIS predefinido

A configuração inicial do GUI QGIS pode ser restaurada por um dos métodos abaixo:

- unchecking *Enable customization* option in the Customization dialog or click the Check All button
- pressing the *Reset* button in the **Settings** frame under *Settings* ► *Options* menu, *System* tab
- launching QGIS at a command prompt with the following command line `qgis --nocustomization`
- setting to `false` the value of *UI* ► *Customization* ► *Enabled* variable under *Settings* ► *Options* menu, *Advanced* tab (see the *warning*).

Na maioria dos casos, você precisa reiniciar o QGIS para que a alteração seja aplicada.

5.5 Atalhos do Teclado

QGIS provides default keyboard shortcuts for many features. You can find them in section *Barra de Menu*. Additionally, the menu option *Settings* ► *Keyboard Shortcuts...* allows you to change the default keyboard shortcuts and add new ones to QGIS features.

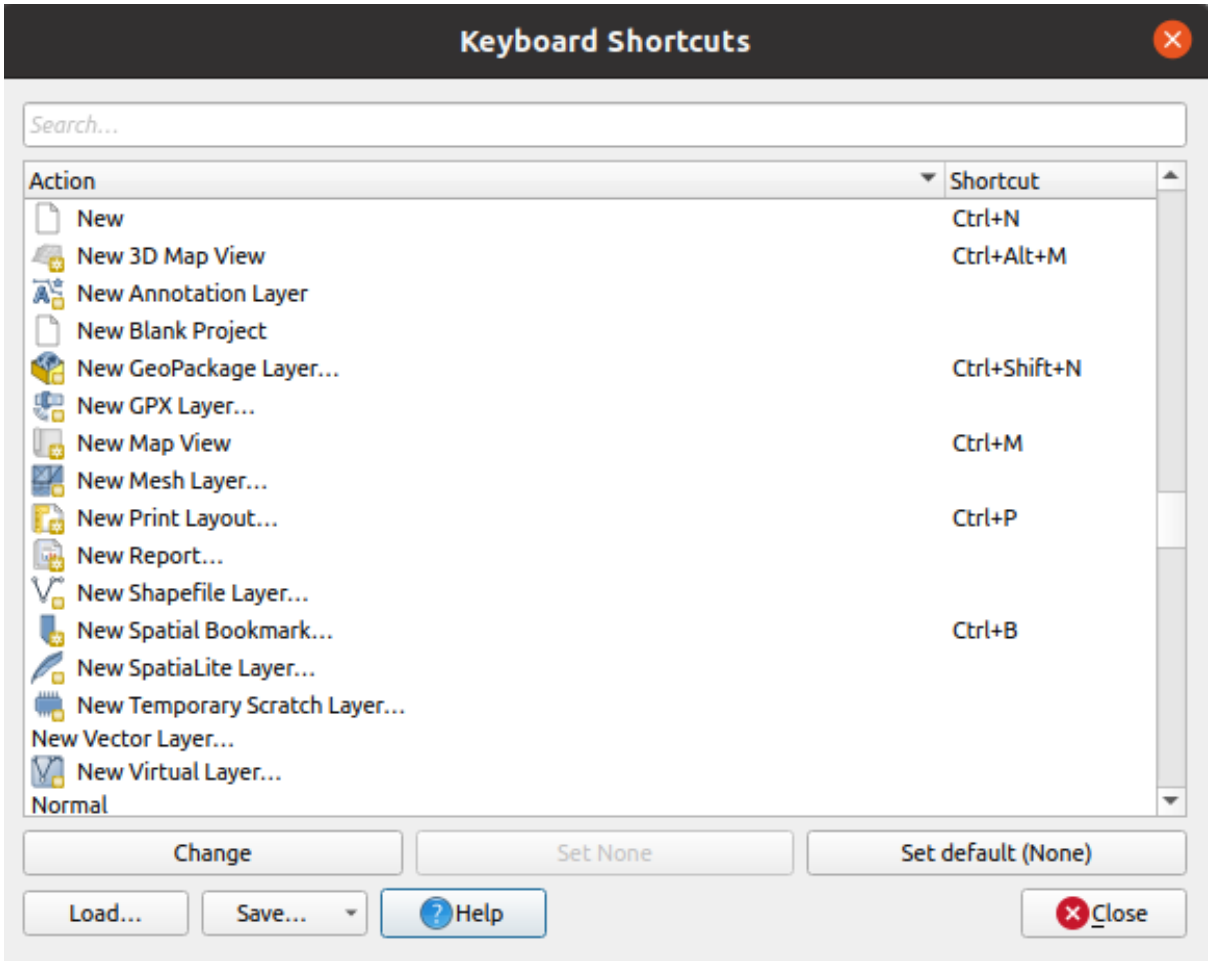


Figura5.45: Definir opções de atalho

A configuração é muito simples. Use a caixa de busca no topo para encontrar uma determinada ação, selecione-a da lista e clique em :

- *Change* and press the new combination you want to assign as new shortcut
- *Set None* to clear any assigned shortcut
- or *Set Default* to backup the shortcut to its original and default value.

Proceed as above for any other tools you wish to customize. Once you have finished your configuration, simply *Close* the dialog to have your changes applied. You can also *Save* the changes either as an `.XML` file with only the User Shortcuts or with all Shortcuts or as an `.PDF` file with all Shortcuts and *Load* them into another QGIS installation.

5.6 Executando o QGIS com configurações avançadas

5.6.1 Command line and environment variables

We've seen that *launching QGIS* is done as for any application on your OS. QGIS provides command line options for more advanced use cases (in some cases you can use an environment variable instead of the command line option). To get a list of the options, enter `qgis --help` on the command line, which returns:

```
QGIS is a user friendly Open Source Geographic Information System.
Usage: /usr/bin/qgis.bin [OPTION] [FILE]
  OPTION:
```

(continua na próxima página)

(continuação da página anterior)

```
        [-v, --version]     display version information and exit
        [-s, --snapshot filename]   emit snapshot of loaded datasets to given file
        [-w, --width width] width of snapshot to emit
        [-h, --height height]       height of snapshot to emit
        [-l, --lang language]       use language for interface text (changes␣
↪existing override)
        [-p, --project projectfile] load the given QGIS project
        [-e, --extent xmin,ymin,xmax,ymax]  set initial map extent
        [-n, --nologo]      hide splash screen
        [-V, --noversioncheck]      don't check for new version of QGIS at startup
        [-P, --noplugins]   don't restore plugins on startup
        [-B, --skipbadlayers]     don't prompt for missing layers
        [-C, --nocustomization]     don't apply GUI customization
        [-z, --customizationfile path]      use the given ini file as GUI␣
↪customization
        [-g, --globalsettingsfile path]     use the given ini file as Global␣
↪Settings (defaults)
        [-a, --authdbdirectory path] use the given directory for authentication␣
↪database
        [-f, --code path]   run the given python file on load
        [-F, --py-args arguments]   arguments for python. These arguments will be␣
↪available for each python execution via 'sys.argv' including the file specified␣
↪by '--code'. All arguments till '--' are passed to python and ignored by QGIS.
        [-d, --defaultui]   start by resetting user ui settings to default
        [--hide-browser]        hide the browser widget
        [--dxf-export filename.dxf]     emit dxf output of loaded datasets to␣
↪given file
        [--dxf-extent xmin,ymin,xmax,ymax]      set extent to export to dxf
        [--dxf-symbology-mode none|symbollayer|feature] symbology mode for dxf␣
↪output
        [--dxf-scale-denom scale]       scale for dxf output
        [--dxf-encoding encoding]       encoding to use for dxf output
        [--dxf-map-theme maptheme]      map theme to use for dxf output
        [--take-screenshots output_path]        take screen shots for the user␣
↪documentation
        [--screenshots-categories categories]   specify the categories of␣
↪screenshot to be used (see QgsAppScreenShots::Categories).
        [--profile name]        load a named profile from the user's profiles␣
↪folder.
        [-S, --profiles-path path]  path to store user profile folders. Will␣
↪create profiles inside a {path}\profiles folder
        [--version-migration]   force the settings migration from older version if␣
↪found
        [--openclprogramfolder]         path to the folder containing the sources␣
↪for OpenCL programs.
        [--help]                this text
        [--]            treat all following arguments as FILEs

 FILE:
   Files specified on the command line can include rasters,
   vectors, and QGIS project files (.qgs and .qgz):
    1. Rasters - supported formats include GeoTiff, DEM
       and others supported by GDAL
    2. Vectors - supported formats include ESRI Shapefiles
       and others supported by OGR and PostgreSQL layers using
       the PostGIS extension
```

Dica

Example Using command line arguments

You can start QGIS by specifying one or more data files on the command line. For example, assuming you are in the `qgis_sample_data` directory, you could start QGIS with a vector layer and a raster file set to load on startup using the following command: `qgis ./raster/landcover.img ./gml/lakes.gml`

`--version`

Esta opção retorna informações da versão do QGIS.

`--snapshot`

This option allows you to create a snapshot in PNG format from the current view. This comes in handy when you have many projects and want to generate snapshots from your data, or when you need to create snapshots of the same project with updated data.

Currently, it generates a PNG file with 800x600 pixels. The size can be adjusted using the `--width` and `--height` arguments. The filename can be added after `--snapshot`. For example:

```
qgis --snapshot my_image.png --width 1000 --height 600 --project my_project.qgs
```

`--width`

This option returns the width of the snapshot to be emitted (used with `--snapshot`).

`--height`

This option returns the height of the snapshot to be emitted (used with `--snapshot`).

`--lang`

Based on your locale, QGIS selects the correct localization. If you would like to change your language, you can specify a language code. For example, `qgis --lang it` starts QGIS in Italian localization.

`--project`

Starting QGIS with an existing project file is also possible. Just add the command line option `--project` followed by your project name and QGIS will open with all layers in the given file loaded.

`--extent`

To start with a specific map extent use this option. You need to add the bounding box of your extent in the following order separated by a comma:

```
--extent xmin,ymin,xmax,ymax
```

This option probably makes more sense when paired with the `--project` option to open a specific project at the desired extent.

--nologo

This option hides the splash screen when you start QGIS.

--noversioncheck

Skip searching for a new version of QGIS at startup.

--noplugins

If you have trouble at start-up with plugins, you can avoid loading them at start-up with this option. They will still be available from the Plugins Manager afterwards.

--nocustomization

Using this option, any existing *GUI customization* will not be applied at startup. This means that any hidden buttons, menu items, toolbars, and so on, will show up on QGIS start up. This is not a permanent change. The customization will be applied again if QGIS is launched without this option.

Esta opção é útil para permitir temporariamente o acesso a ferramentas que foram removidas por personalização.

--skipbadlayers

Using this option, you can avoid QGIS prompting the *Handle Unavailable Layers* dialog at startup. The project file is loaded, with missing layers kept as unavailable. More details on the topic at *Manipulação de caminhos de arquivos quebrados*.

--customizationfile

Using this option, you can define a UI customization file, that will be used at startup.

--globalsettingsfile

The equivalent environment variable is `QGIS_GLOBAL_SETTINGS_FILE`.

Using this option, you can specify the path for a Global Settings file (`.ini`), also known as the Default Settings. The settings in the specified file replace the original inline default ones, but the user profiles' settings will be set on top of those.

QGIS looks for the default global settings file in the following order and only the first found file will be used:

- path specified by the commandline parameter
- path defined by the environment variable
- the AppDataLocation folder, where persistent application data can be stored; it is managed by the user or system administrator and is not touched by installer and does not require any additional setup like passing commandline parameters or settings environment variable. Depending on the OS, it is:
  - `$HOME/.local/share/QGIS/QGIS3/`
  - `C:\Users\<username>\%AppData%\Roaming\QGIS\QGIS3\`
  - `$HOME/Library/Application Support/QGIS/QGIS3/`
- the installation directory, i.e., `your_QGIS_package_path/resources/qgis_global_settings.ini`.

Atualmente, não há como especificar um arquivo no qual gravar configurações; portanto, você pode criar uma cópia de um arquivo de configurações original, renomeá-lo e adaptá-lo.

Setting the `qgis_global_setting.ini` file path to a network shared folder, allows a system administrator to change global settings and defaults in several machines by only editing one file.

`--authdbdirectory`

This option is similar to `--globalsettingsfile`, but defines the path to the directory where the authentication database will be stored and loaded.

`--code`

Esta opção pode ser usada para executar um determinado arquivo python diretamente após o início do QGIS.

For example, when you have a python file named `load_alaska.py` with following content:

```
from qgis.utils import iface
raster_file = "/home/gisadmin/Documents/qgis_sample_data/raster/landcover.img"
layer_name = "Alaska"
iface.addRasterLayer(raster_file, layer_name)
```

Assuming you are in the directory where the file `load_alaska.py` is located, you can start QGIS, load the raster file `landcover.img` and give the layer the name 'Alaska' using the following command:

```
qgis --code load_alaska.py
```

`--py-args`

This option allows to pass arguments to the script provided via the *–code* parameter and more generally to each python execution. After `--py-args`, it is not possible to pass any other parameter than – to QGIS. All arguments after `--py-args` till `--` are passed over to the python interpreter and ignored by QGIS.

```
qgis --code /home/user/test.py --py-args --specialScriptArguments "a text arg"
↪'and another arg' -- layer1 layer2
```

In the above code, `test.py` will have this content in `sys.argv`: ['/home/user/test.py', '–specialScriptArguments', 'a text arg', 'and another arg']. `layer1` and `layer2` will be normally handled by QGIS as layers to load.

`--defaultui`

On load, **permanently resets** the user interface (UI) to the default settings. This option will restore the panels and toolbars visibility, position, and size. Unless it's changed again, the default UI settings will be used in the following sessions.

Notice that this option doesn't have any effect on *GUI customization*. Items hidden by GUI customization (e.g., the status bar) will remain hidden even using the `--defaultui` option. See also the `--nocustomization` option.

--hide-browser

On load, hides the *Browser* panel from the user interface. The panel can be enabled by right-clicking a space in the toolbars or using the *View* ► *Panels* (*Settings* ► *Panels* in Linux KDE).

Unless it's enabled again, the Browser panel will remain hidden in the following sessions.

--dxf-*

Estas opções podem ser usadas para exportar um projeto QGIS para um arquivo DXF. Várias opções estão disponíveis:

- *–dxf-export*: the DXF filename into which to export the layers;
- *–dxf-extent*: the extent of the final DXF file;
- *–dxf-symbology-mode*: several values can be used here: `none` (no symbology), `symbollayer` (Symbol layer symbology), `feature` (feature symbology);
- *–dxf-scale-denom*: the scale denominator of the symbology;
- *–dxf-encoding*: the file encoding;
- *–dxf-map-theme*: choose a *map theme* from the layer tree configuration.

--take-screenshots

Takes screenshots for the user documentation. Can be used together with `--screenshots-categories` to filter which categories/sections of the documentation screenshots should be created (see QgsAppScreenShots::Categories).

--profile

Loads QGIS using a specific profile from the user's profile folder. If the named profile doesn't exist, it will be created. This option takes precedence over *user profile startup setting*.

--profiles-path

With this option, you can choose a path to load and save the profiles (user settings). It creates profiles inside a `{path}\profiles` folder, which includes settings, installed plugins, processing models and scripts, and so on.

This option allows you to, for instance, carry all your plugins and settings in a flash drive, or, for example, share the settings between different computers using a file sharing service.

The equivalent environment variable is `QGIS_CUSTOM_CONFIG_PATH`.

--version-migration

If settings from an older version are found (*e.g.*, the `.qgis2` folder from QGIS 2.18), this option will import them into the default QGIS profile.

--openclprogramfolder

Using this option, you can specify an alternative path for your OpenCL programs. This is useful for developers while testing new versions of the programs without needing to replace the existing ones.

The equivalent environment variable is `QGIS_OPENCL_PROGRAM_FOLDER`.

--

This option allows to pass following arguments as files to open in QGIS. They can be supported rasters or vectors files, QGIS layer definition files or project files.

```
qgis -- /home/user/project.qgs /home/user/path_to_data/layer1.gpkg
```

The above code will start QGIS, open the `project` file and load `layer1` in it.

5.6.2 Implantando o QGIS em uma organização

If you need to deploy QGIS within an organization with a custom configuration file, first you need to copy/paste the content of the default settings file located in `your_QGIS_package_path/resources/qgis_global_settings.ini`. This file already contains some default sections identified by a block starting with `[]`. We recommend that you keep these defaults values and add your own sections at the bottom of the file. If a section is duplicated in the file, QGIS will take the last one from top to bottom.

You can change `allowVersionCheck=false` to disable the QGIS version check.

Se você não quiser exibir a janela de migração após uma nova instalação, precisará da seguinte seção:

```
[migration]
fileVersion=2
settings=true
```

Se você deseja adicionar uma variável personalizada no escopo global:

```
[variables]
organisation="Your organization"
```

A lot of settings can be set using the `.INI` file such as WMS/WMTS, PostgreSQL connections, proxy settings, maptips, default values for algorithms parameters… For example, if you want to configure organization wide default values for some algorithms it would be handy if you could configure it in the prepared ini-file for all users. If the users then open the toolbox to execute an algorithm, the default values will be there. To set the default value, e.g., for `GRID_SIZE` parameter for the `native:intersection` algorithm, you could write the following into the `.INI` file:

```
[Processing]
DefaultGuiParam\native%3Aintersection\GRID_SIZE=0.01
```

Not all settings can be prepared from QGIS GUI, but to discover the possibilities of the settings `.INI` file, we suggest that you set the config you would like in QGIS Desktop and then search for it in your `.INI` file located in your profile using a text editor.

Finally, you need to set the environment variable `QGIS_GLOBAL_SETTINGS_FILE` to the path of your customized file.

In addition, you can also deploy files such as Python macros, color palettes, layout templates, project templates… either in the QGIS system directory or in the QGIS user profile.

- Layout templates must be deployed in the `composer_templates` directory.
- Project templates must be deployed in the `project_templates` directory.
- Custom Python macros must be deployed in the `python` directory.

CAPÍTULO 6

Trabalhando com Projeções

Um Sistema de Referência de Coordenadas, ou SRC, é um método de associar coordenadas numéricas com uma posição na superfície da Terra. O QGIS tem suporte para aproximadamente 7.000 SRCs padrão, cada um com diferentes casos de uso, prós e contras! Escolher um sistema de referência apropriado para seus projetos e dados QGIS pode ser uma tarefa complexa, mas felizmente o QGIS ajuda a guiá-lo através desta escolha, e torna o trabalho com diferentes CRSs tão transparente e preciso quanto possível.

6.1 Visão geral do Suporte a Projeções

QGIS has support for approximately 7,000 known CRSs. These standard CRSs are based on those defined by the European Petroleum Search Group (EPSG) and the Institut Geographique National de France (IGNF), and are made available in QGIS through the underlying "Proj" projection library. Commonly, these standard projections are identified through use of an authority:code combination, where the authority is an organisation name such as "EPSG" or "IGNF", and the code is a unique number associated with a specific CRS. For instance, the common WGS 84 latitude/longitude CRS is known by the identifier `EPSG:4326`, and the web mapping standard CRS is `EPSG:3857`.

Custom, user-created CRSs are stored in a user CRS database. See section *Sistema de Referência de Coordenadas personalizado* for information on managing your custom coordinate reference systems.

6.2 Sistemas de referência de coordenadas da camada

In order to correctly project data into a specific target CRS, either your data must contain information about its coordinate reference system or you will need to manually assign the correct CRS to the layer. For PostgreSQL layers, QGIS uses the spatial reference identifier that was specified when that PostgreSQL layer was created. For data supported by GDAL, QGIS relies on the presence of a recognized means of specifying the CRS. For instance, for the Shapefile format this is a file containing an ESRI Well-Known Text (WKT) representation of the layer's CRS. This projection file has the same base name as the `.shp` file and a `.prj` extension. For example, `alaska.shp` would have a corresponding projection file named `alaska.prj`.

Whenever a layer is loaded into QGIS, QGIS attempts to automatically determine the correct CRS for that layer. In some cases this is not possible, e.g. when a layer has been provided without retaining this information. You can configure QGIS behavior whenever it cannot automatically determine the correct CRS for a layer:

1. Abrir **:menuseleção:`Configurações-->`** *Opções...* ► *SRC*`

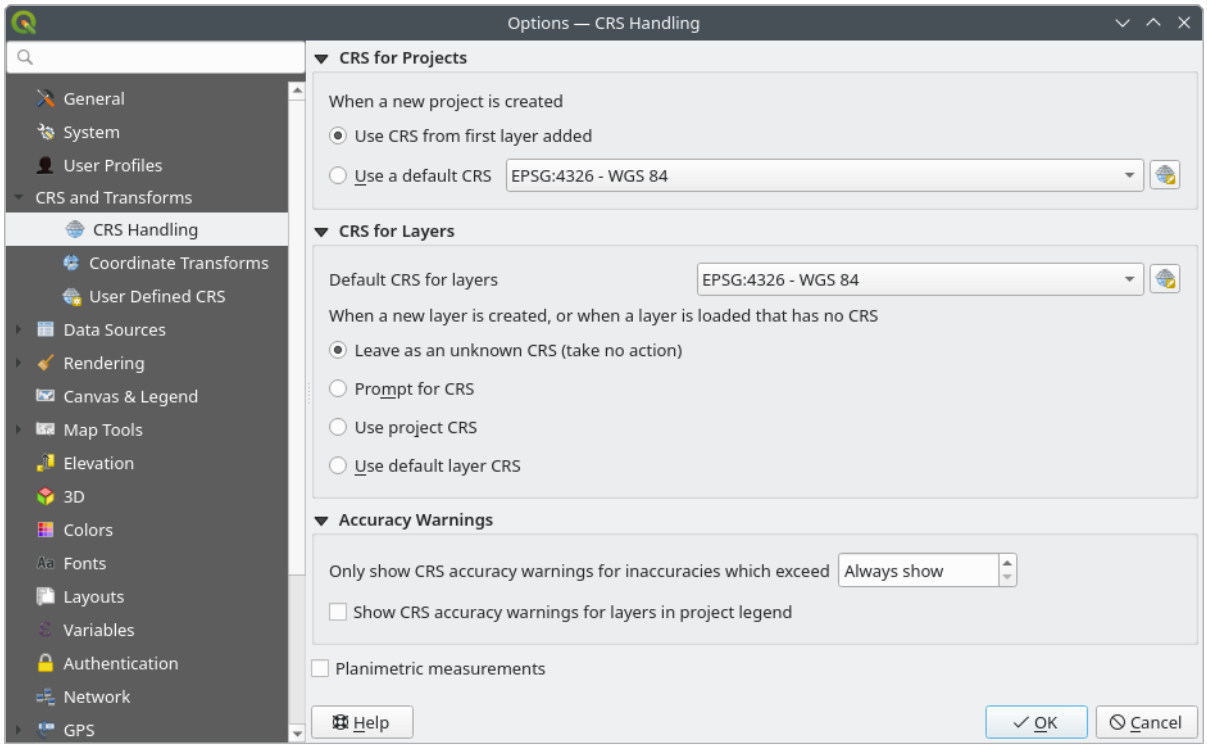


Figura6.1: A guia SRC na Caixa de Diálogo de Opções QGIS

2. Under the *CRS for layers* group, set the action to do *when a new layer is created, or when a layer is loaded that has no CRS*. One of:

   - *Leave as unknown CRS (take no action)*: there will be no prompt to select a CRS when a layer without CRS is loaded, defering CRS choice to a later time. Convenient when loading a lot of layers at once. Such layers will be identifiable in the *Layers* panel by the icon next to them. They'll also be un-referenced, with coordinates from the layer treated as purely numerical, non-earth values, i.e. the same behavior as all layers get when *a project is set to have no CRS*.
   - *Prompt for CRS*: it will prompt you to manually select the CRS. Selecting the correct choice is crucial, as a wrong choice will place your layer in the wrong position on the Earth's surface! Sometimes, accompanying metadata will describe the correct CRS for a layer, in other cases you will need to contact the original author of the data to determine the correct CRS to use.
   - *Usar SRC do projeto*
   - *Use default layer CRS*, as set in the *Default CRS for layers* combobox above.

Dica

Atribuir o mesmo SRC a várias camadas que não tenham SRC ou que tenham um SRC errado em uma única operação:

1. Selecione as camadas no painel :guilabel:`Camadas'.
2. Pressione `Ctrl+Shift+C`. Você também pode clicar com o botão direito sobre uma das camadas selecionadas ou ir para *Camada* ► *Definir SRC da(s) camada(s)*
3. Encontre e selecione o CRS adequado para usar
4. E pressione *OK*. Você pode confirmar que ela foi definida corretamente na aba *Fonte* das propriedades das camadas.

Observe que mudar o SRC neste cenário não altera de forma alguma a fonte de dados subjacente, mas apenas muda a forma como o QGIS interpreta as coordenadas brutas da camada no projeto QGIS atual.

6.3 Sistemas referência de coordenadas projetadas

Cada projeto no QGIS também tem um Sistema de Referência de Coordenadas associado. O projeto SRC determina como os dados são projetados de suas coordenadas brutas subjacentes para o mapa plano renderizado em sua tela de mapa QGIS.

QGIS supports “on the fly” CRS transformation for both raster and vector data. This means that regardless of the underlying CRS of particular map layers in your project, they will always be automatically transformed into the common CRS defined for your project. Behind the scenes, QGIS transparently reprojects all layers contained within your project into the project’s CRS, so that they will all be rendered in the correct position with respect to each other!

É importante fazer uma escolha apropriada de SRC para seus projetos QGIS. Escolher um SRC inadequado pode fazer com que seus mapas pareçam distorcidos e reflitam mal os tamanhos relativos do mundo real e as posições das feições. Normalmente, ao trabalhar em áreas geográficas menores, haverá vários SRCs padrão usados em um determinado país ou área administrativa. É importante pesquisar quais SRCs são escolhas apropriadas ou padrão para a área que você está mapeando e garantir que seu projeto QGIS siga esses padrões.

Por padrão, o QGIS inicia cada novo projeto usando uma projeção padrão global. Este SRC padrão é `EPSG:4326` (também conhecido como “WGS 84”), e é um sistema de referência global baseado na latitude/longitude. Este SRC padrão pode ser alterado através da configuração *SRC para Novos Projetos* na guia *SRC* em *Configurações* ► *Opções...* (veja `figura_opções_de_projeção`). Há uma opção para definir automaticamente o SRC do projeto para corresponder ao SRC da primeira camada carregada em um novo projeto ou, alternativamente, você pode selecionar um SRC padrão diferente para utilizar em todos os projetos recém-criados. Esta escolha será salva para uso em sessões subseqüentes do QGIS.

O projeto SRC também pode ser definido através da guia *SRC* do :menuselection:`Projeto –> Propriedades…`janela. Também será mostrado na parte inferior direita da barra de status QGIS.

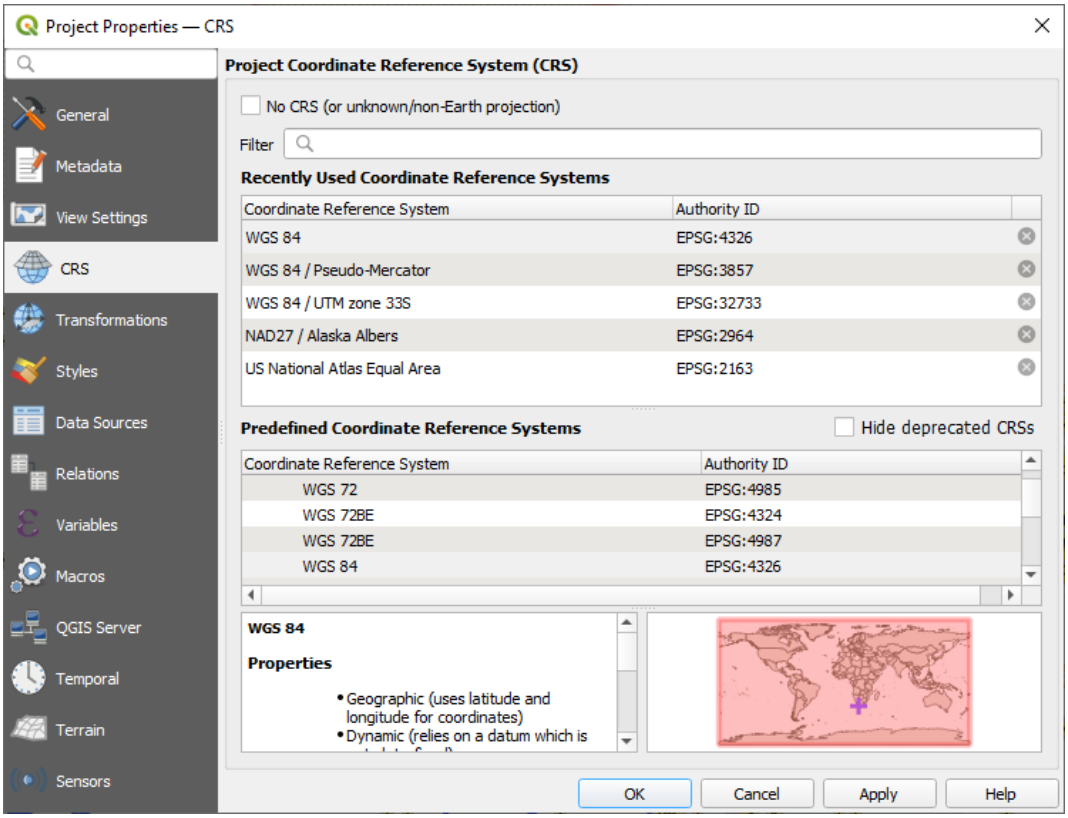


Figura6.2: Caixa de Diálogo de Propriedades de Projeto

As opções disponíveis são:

- *No CRS (or unknown/non-Earth projection)*: Checking this setting will disable ALL projection handling within the QGIS project, causing all layers and map coordinates to be treated as simple 2D Cartesian coordinates, with no relation to positions on the Earth's surface. It can be used to guess a layer CRS (based on its raw coordinates or when using QGIS for non earth uses like role-playing game maps, building mapping or microscopic stuff. In this case:
  - Nenhuma reprojeção é feita durante a renderização das camadas: as feições são apenas desenhadas usando suas coordenadas brutas.
  - A elipsóide é bloqueada e forçada a "Nenhum/Planimétrico".
  - As unidades de distância e área, e a exibição de coordenadas são bloqueadas e forçadas a "unidades desconhecidas"; todas as medições são feitas em unidades de mapa desconhecidas, e nenhuma conversão é possível.
- or an existing coordinate reference system that can be *geographic*, *projected* or *user-defined*. A preview of the CRS extent on earth is displayed to help you select the appropriate one. Layers added to the project are translated on-the-fly to this CRS in order to overlay them regardless of their original CRS. Use of units and ellipsoid setting are available and make sense and you can perform calculations accordingly.

If you change your project's CRS and want the units (for distance and area) to match the units defined by the CRS, make sure you have selected *Map units* in the relevant settings, in the *General* tab of the *Project* ► *Properties…* dialog.

Dica

Configurando a projeção SRC a partir de uma camada

Você pode atribuir um SRC ao projeto usando um SRC de camada:

1. No painel *Camadas*, clique com o botão direito do mouse sobre a camada que você quer escolher o SRC

2. Selecione *Configurar SRC do projeto a partir da Camada*.

The project's CRS is redefined using the layer's CRS. Map canvas extent, coordinates display are updated accordingly and all the layers in the project are on-the-fly translated to the new project CRS.

6.4 Seletor do Sistemas de Referência de Coordenadas

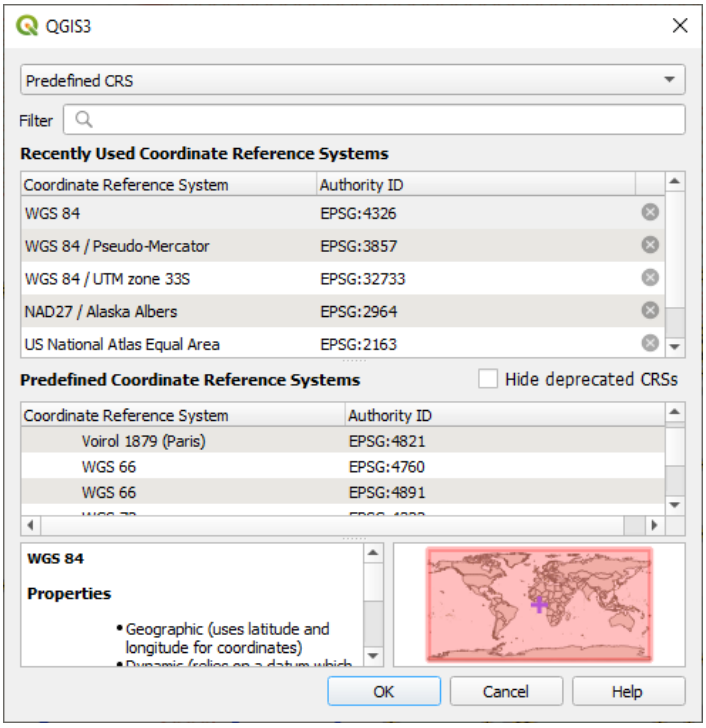


Figura6.3: CRS Selector

Este diálogo ajuda a atribuir um Sistema de Referência Coordenada a um projeto ou uma camada, desde que um conjunto de bancos de dados de projeção. Os itens na janela são:

- **Filtro**: Se você conhece o código EPSG, o identificador ou o nome de um Sistema de Referência de Coordenadas, você pode usar o recurso de busca para encontrá-lo. Digite o código EPSG, o identificador ou o nome.
- ** Sistemas de referência de coordenadas recentemente utilizados**: Se você tiver certos SRCs que você usa freqüentemente em seu trabalho diário de SIG, estes serão exibidos nesta lista. Clique em um destes itens para selecionar o SRC associado.
- **Sistemas de referência coordenados do mundo**: Esta é uma lista de todos os SRCs suportados pelo QGIS, incluindo os sistemas de referência de coordenadas geográficas, projetadas e personalizadas. Para definir um SRC, selecione-o da lista expandindo o nó apropriado e selecionando o SRC. O SRC ativo é pré-selecionado.
- **PROJ text**: This is the CRS string used by the PROJ projection engine. This text is read-only and provided for informational purposes.

If you want to remove a recently used CRS from the list that can be done in several ways:

- Press Remove from recently used CRS button on the right side.
- Press `Del` after selecting CRS that you want to remove.

- Right-click on used CRS and choose *Remove selected CRS from recently used CRS* to remove selected CRS or *Clear all recently used CRS* to remove all used CRS.

The CRS selector also shows a rough preview of the geographic area for which a selected CRS is valid for use. Many CRSs are designed only for use in small geographic areas, and you should not use these outside of the area they were designed for. The preview map shades an approximate area of use whenever a CRS is selected from the list. In addition, this preview map also shows an indicator of the current main canvas map extent.

6.5 Sistema de Referência de Coordenadas personalizado

Se o QGIS não fornece o sistema de referência de coordenadas(SRC) que você precisa, você pode definir um SRC personalizado. Para definir um SRC, selecione *Customizar SRC...* no menu :menuselection:`Configurações'. Os SRC personalizados são armazenados em seu banco de dados de usuários QGIS. Além de seus SRCs personalizados, este banco de dados também contém seus marcadores espaciais e outros dados personalizados.

Defining a custom CRS in QGIS requires a good understanding of the PROJ projection library. To begin, refer to "Cartographic Projection Procedures for the UNIX Environment - A User's Manual" by Gerald I. Evenden, U.S. Geological Survey Open-File Report 90-284, 1990 (available at https://pubs.usgs.gov/of/1990/of90-284/ofr90-284.pdf).

Este manual descreve o uso de `proj` e utilitários de linha de comando relacionados. Os parâmetros cartográficos utilizados com o `proj` estão descritos no manual do usuário e são os mesmos que os utilizados pelo QGIS.

A janela *Definição de Sistema de Referência de Coordenadas personalizado* exige apenas dois parâmetros para definir um SRC ao usuário:

1. Um nome descritivo
2. Os parâmetros cartográficos em formato PROJ ou WKT

Para criar um novo SRC:

1. Clique no botão Adicionar novo SRC.
2. Insira um nome descritivo
3. Select the format: it can be *Proj String* or *WKT*
4. Adicionar ao SRC *Parâmetros*.

> **Nota**
>
> **Preferir o armazenamento da definição de SRC no formato WKT**
>
> Although both `Proj String` and `WKT` formats are supported, it's highly recommended to store projection definitions in the WKT format. Therefore, if the available definition is in the proj format, select that format, enter the parameters and then switch to WKT format. QGIS will convert the definition to the WKT format that you can later save.

5. Click *Validate* to test whether the CRS definition is an acceptable projection definition.

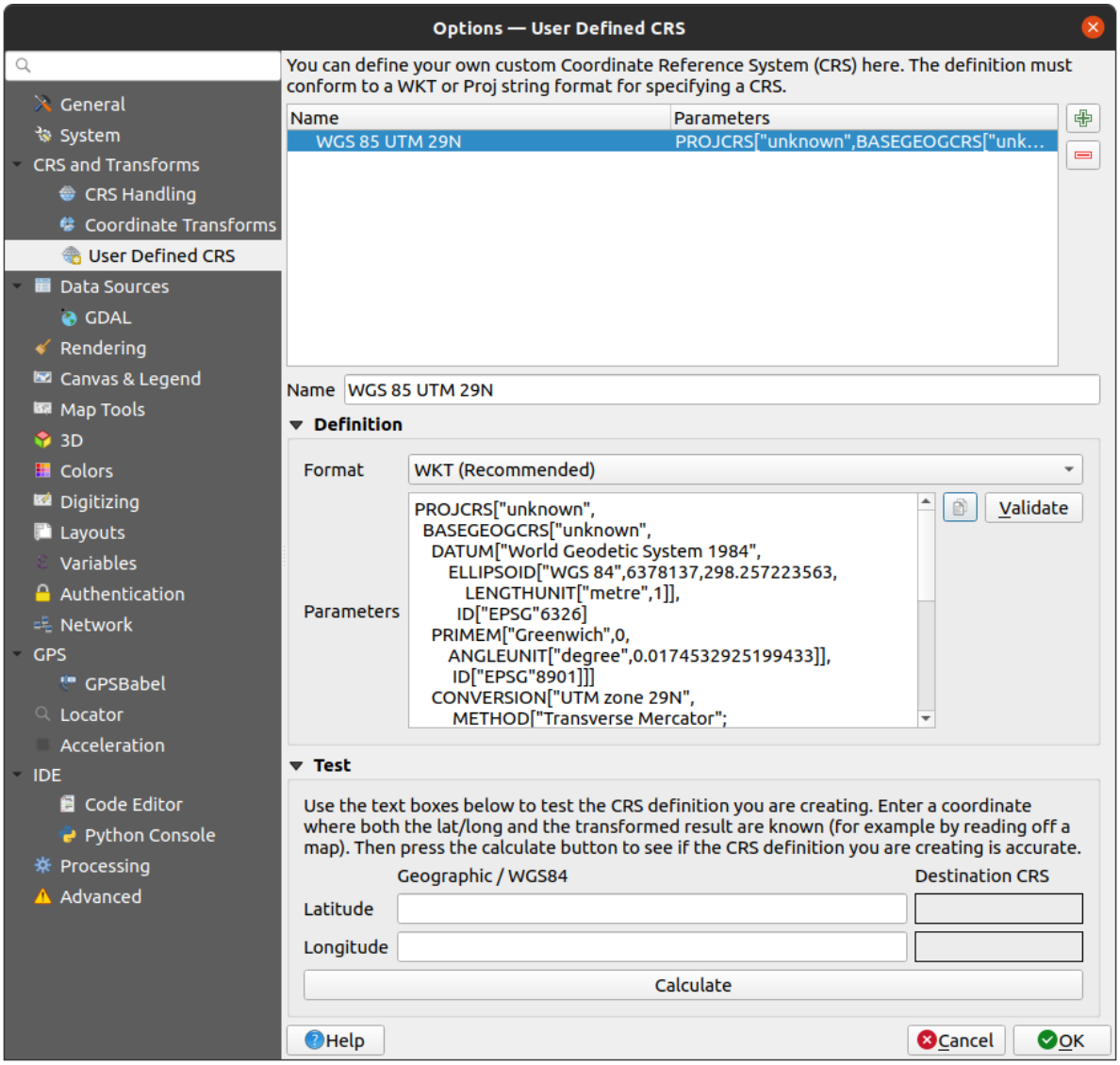


Figura6.4: Custom CRS Dialog

Você pode testar seus parâmetros SRC para ver se eles dão resultados coerentes. Para fazer isso, digite os valores conhecidos de latitude e longitude WGS 84 nos campos *Norte* e *Leste*, respectivamente. Clique em *Calcular*, e compare os resultados com os valores conhecidos em seu sistema de referência de coordenadas.

6.5.1 Integrate an NTv2-transformation in QGIS

To integrate an NTv2 transformation file in QGIS you need one more step:

1. Place the NTv2 file (.gsb) in the CRS/Proj folder that QGIS uses (e.g. `C:\OSGeo4W64\share\proj` for windows users)
2. Add **nadgrids** (`+nadgrids=nameofthefile.gsb`) to the Proj definition in the *Parameters* field of the *Custom Coordinate Reference System Definition* (*Settings* ► *Custom Projections…*).

Figura6.5: Configurar uma transformação NTv2

6.6 Transformações de Datum

In QGIS, 'on-the-fly' CRS transformation is enabled by default, meaning that whenever you use layers with different coordinate systems QGIS transparently reprojects them to the project CRS. For some CRS, there are a number of possible transforms available to reproject to the project's CRS!

By default, QGIS will attempt to use the most accurate transformation available. However, in some cases this may not be possible, e.g. whenever additional support files are required to use a transformation. Whenever a more accurate transformation is available, but is not currently usable, QGIS will show an informative warning message advising you of the more accurate transformation and how to enable it on your system. Usually, this requires download of an external package of transformation support files, and extracting these to the `proj` folder under your QGIS *user profile* folder.

Se desejar, o QGIS também pode lhe solicitar sempre que várias transformações possíveis possam ser feitas entre dois CRSs, e permitir que você faça uma seleção informada de qual é a transformação mais apropriada para usar para seus dados.

This customization is done in the *Settings* ► *Options* ► *Transformations* tab menu under the *Default datum transformations* group:

- using *Ask for datum transformation if several are available*: when more than one appropriate datum transformation exist for a source/destination CRS combination, a dialog will automatically be opened prompting users to choose which of these datum transformations to use for the project. If the *Make default* checkbox is ticked when selecting a transformation from this dialog, then the choice is remembered and automatically applied to any newly created QGIS projects.
- ou definir uma lista de transformações de dados apropriadas para usar como padrão ao carregar uma camada em um projeto ou ao reprojetar uma camada.

  Use the button to open the *Select Datum Transformations* dialog. Then:

  1. Choose the *Source CRS* of the layer, using the drop-down menu or the Select CRS widget.
  2. Fornecer o *Destino SRC* da mesma forma.
  3. Uma lista das transformações disponíveis da origem ao destino será mostrada na tabela. Clicando em uma linha serão mostrados detalhes sobre as configurações aplicadas e a correspondente precisão e área de uso da transformação.

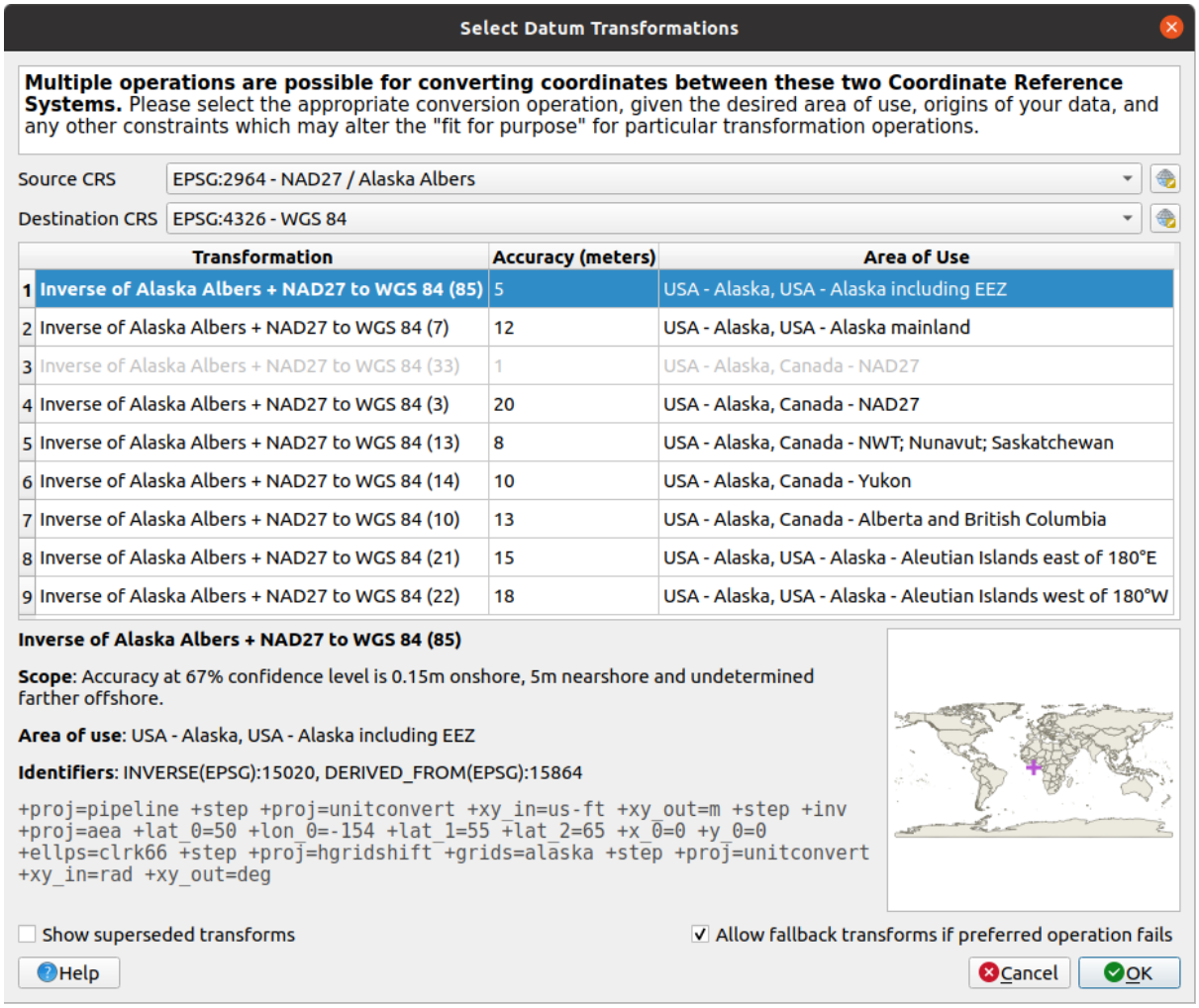


Figura6.6: Selecionando uma transformação padrão de datum preferida

In some cases a transformation may not be available for use on your system. In this case, the transformation will still be shown (greyed) in this list but can not be picked until you install the required package of transformation support. Usually, a button is provided to download and install the corresponding grid, which is then stored under the `proj` folder in the active *user profile* directory.

4. Encontre a transformação de sua preferência e selecione
5. Defina se você *Permitir transformações em caso de falha na operação preferida*.
6. Haz clic en *Aceptar*.

   A row is added to the table under *Default Datum Transformations* with information about the *Source CRS*, the *Destination CRS*, the *Operation* applied for the transformation and whether *Allow fallback Transforms* is enabled.

From now, QGIS automatically uses the selected datum transformations for further transformation between these two CRSs until you remove it ( ) from the list or change the entry ( ) in the list.

Datum transformations set in the *Settings* ► *Options* ► *Transformations* tab will be inherited by all new QGIS projects created on the system. Additionally, a particular project may have its own specific set of transformations specified via the *CRS* tab of the *Project properties* dialog (*Project* ► *Properties…*). These settings apply to the current project only.

CAPÍTULO 7

Visualizing Maps

Creating maps is one of the “business ends” of QGIS. Map views are canvases to visualize spatial layers loaded in the project, with the various rendering settings applied to them. They can be of *2D* or *3D* type or plotted along a *profile line*, show different scale or extent, or display a different set of the loaded layers thanks to *map themes*.

7.1 2D Map View

The 2D map view (also called **Map canvas**) is the central place where maps are displayed. QGIS opens by default with a single map view (called `main map`), showing layers in 2D, and tightly bound to the *Layers* panel. That window reflects the rendering (symbology, labeling, visibilities…) you applied to the loaded layers.

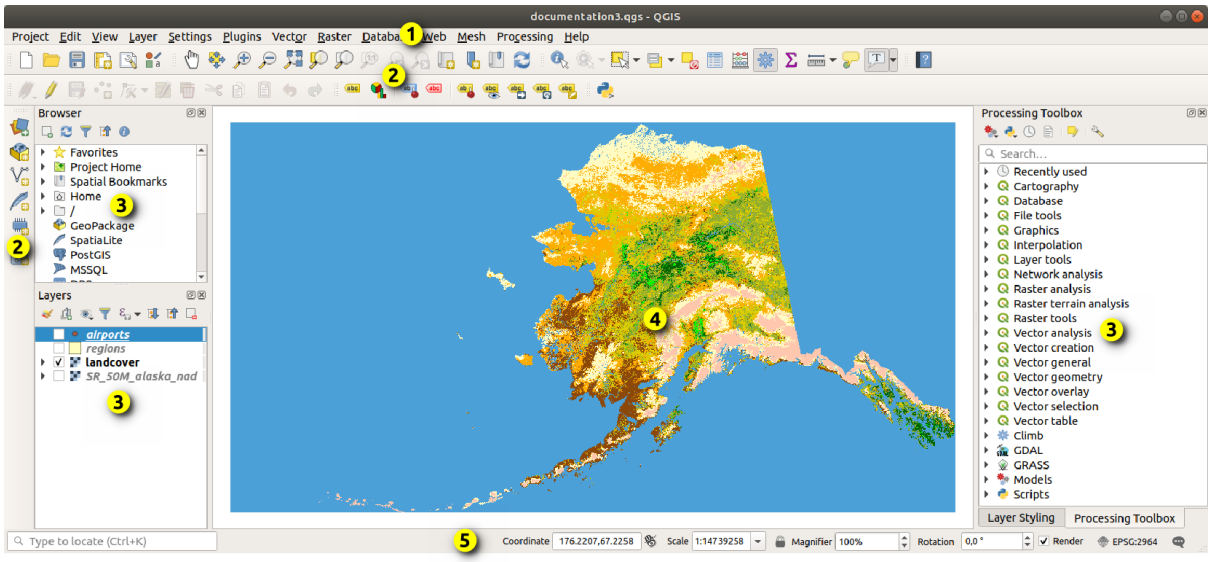


Figura7.1: Interface do QGIS

7.1.1 Explorando a visualização do mapa

When you add a layer (see e.g. *Abrir Dados*), QGIS automatically looks for its CRS. If a different CRS is set by default for the project (see *Sistemas referência de coordenadas projetadas*) then the layer extent is "on-the-fly" translated to that CRS, and the map view is zoomed to that extent if you start with a blank QGIS project. If there are already layers in the project, no map canvas resize is performed, so only features falling within the current map canvas extent will be visible.

Click on the map view and you should be able to interact with it, panning or zooming to different areas of the map. Dedicated tools are provided in the *Navigation Toolbar* and in the *View* menu, with handful shortcuts from the keyboard or the mouse buttons.

Tabela7.1: Map canvas navigation tools

Ferramenta	Utilização
Pan Map	• Single left click: the map is centered on the clicked point, at the same scale • Hold down the left mouse button and drag the map canvas.
Zoom In	• Single left click: the map is centered on the clicked point, while the scale gets doubled • Drag a rectangle on the map canvas with the left mouse button to zoom in to an area. • Hold the `Alt` key to switch to the Zoom Out tool.
Zoom Out	• Single left click: the map is centered on the clicked point, while the scale gets halved • Drag a rectangle on the map canvas with the left mouse button to zoom out from an area. • Hold the `Alt` key to switch to the Zoom In tool.
Pan Map to Selection	Pan the map to the selected features of all the selected layers in the *Layers* panel.
Zoom To Selection	Zoom to the selected features of all the selected layers in the *Layers* panel. *Also available in the layer contextual menu*
Zoom To Layer(s)	Zoom to the extent of all the selected layers in the *Layers* panel. *Also available in the layer contextual menu*
Zoom Full	Zoom to the extent of all the layers in the project or to the *project full extent*.
Zoom Last	Zoom the map to the previous extent in history.
Zoom Next	Zoom the map to the next extent in history.
Zoom to Native Resolution	Zoom the map to a level where one pixel of the active raster layer covers one screen pixel. *Also available in the layer contextual menu*
Mouse wheel	• Pan map: Hold and drag the mouse wheel. • Zoom: Roll the mouse wheel to zoom in or zoom out. With `Ctrl` key pressed while rolling the mouse wheel results in a finer zoom. • Press the back or forward button to browse the map canvas zoom history.

continua na próxima página

Tabela 7.1 – continuação da página anterior

Ferramenta	Utilização
Keyboard	• Pan map: Hold down the `Space` key and move the mouse. Press the arrow keys to pan up, down, left and right. • Zoom in: Press `PgUp` or `Ctrl++` • Zoom out: Press `PgDown` or `Ctrl+-` • Zoom to area: When certain map tools are active (Identify, Measure…), hold down `Shift` and drag a rectangle on the map to zoom to that area. Not compatible with active selection or edit tools.

Right-click over the map and you should be able to *Copy coordinates* of the clicked point in the map CRS, in WGS84 or in a custom CRS. The copied information can then be pasted in an expression, a script, text editor or spreadsheet…

7.1.2 Controlling map rendering

By default, QGIS renders all visible layers whenever the map canvas is refreshed. The events that trigger a refresh of the map canvas include:

- changing the visibility of a layer
- modifying symbology of a visible layer
- adicionando uma camada
- panning or zooming
- redimensionando a janela QGIS

QGIS permite controlar o processo de renderização de várias formas.

- at the *global level*
- per layer, using e.g. the *scale dependent rendering*
- or with dedicated tools in the GUI.

To stop the map drawing, press the `Esc` key. This will halt the refresh of the map canvas and leave the map partially drawn. It may however take a bit of time after pressing `Esc` for the map drawing to halt.

To suspend rendering, click the *Render* checkbox in the bottom-right corner of the status bar. When *Render* is unchecked, QGIS does not redraw the canvas in response to any of the usual triggers mentioned earlier. Examples of when you might want to suspend rendering include:

- adding many layers and symbolizing them prior to drawing
- adding one or more large layers and setting scale dependency before drawing
- adding one or more large layers and zooming to a specific view before drawing
- qualquer combinação dos itens acima

Caixa de seleção :guilabel:caixa de seleção *Renderizar* permite renderização e causa uma atualização imediata na tela do mapa.

7.1.3 Controle baseado no tempo na tela do mapa

O QGIS pode manipular o controle temporal em camadas carregadas, ou seja, modificar a renderização da tela de mapa com base em uma variação de tempo. Para conseguir isso, você precisa:

1. Camadas que têm propriedades temporais dinâmicas definidas. O QGIS suporta controle temporal para diferentes provedores de dados, com configurações personalizadas. Trata-se principalmente de definir o intervalo de tempo no qual a camada seria exibida:
   - *raster layers*: controls whether to display or not the layer.
   - *WMTS layers*: controls whether the data should be rendered based on a static time range or following a dynamic temporal range
   - *vector layers*: features are filtered based on time values associated to their attributes
   - *mesh layers*: displays dynamically the active dataset groups values

   When dynamic temporal options are enabled for a layer, an icon is displayed next to the layer in the *Layers* panel to remind you that the layer is temporally controlled. Click the icon to update the temporal settings.
2. Enable the temporal navigation of the map canvas using the *Temporal controller panel*. The panel is activated:
   - using the Temporal controller panel icon located in the *Map Navigation* toolbar
   - or from the *View* ► *Panels* ► *Temporal controller panel* menu
   - or from the *View* ► *Data Filtering* ► *Temporal controller panel* menu

O painel de controle temporal

The *Temporal controller* panel has the following modes:

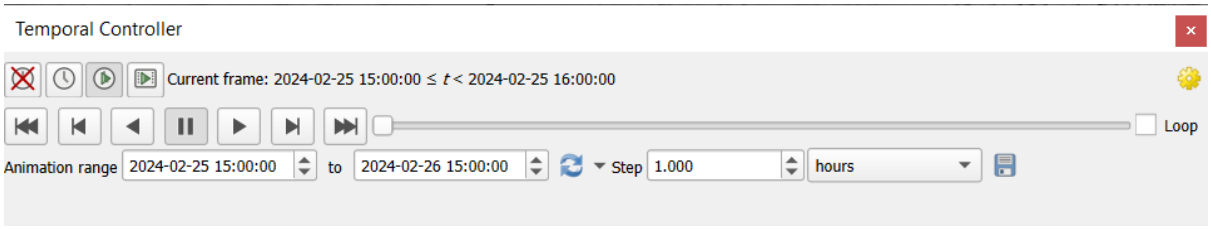


Figura7.2: Painel Controlador Temporal no modo de navegação

- Desligar a navegação temporal: todas as configurações temporais são desativadas e as camadas visíveis são renderizadas como de costume
- Fixed range temporal navigation: a time range is set and only layers (or features) whose temporal range overlaps with this range are displayed on the map.
- Animated temporal navigation: a time range is set, split into steps, and only layers (or features) whose temporal range overlaps with each frame are displayed on the map
- Animated movie: a total number of frames is set for the animation, with the map settings advancing at each step, displaying the progression frame by frame without any time-based filtering of data.
- Definições para controle geral da animação
  - *Frames rate*: number of steps that are shown per second
  - *Cumulative range*: all animation frames will have the same start date-time but different end dates and times. This is useful if you wish to accumulate data in your temporal visualization instead of showing a 'moving time window' across your data.

Animando uma navegação temporal

Uma animação é baseada em um conjunto variável de camadas visíveis em momentos específicos dentro de um intervalo de tempo. Para criar uma animação temporal:

1. Toggle on the Animated temporal navigation, displaying the animation player widget
2. Digite o *Faixa de tempo* a considerar. Usando o botão , isto pode ser definido como:
   - *Set to full range* of all the time enabled layers
   - *Set to preset project range* as defined in the *project properties*
   - *Set to single layer's range* taken from a time-enabled layer
3. Fill in the time *Step* to split the time range. Different units are supported, from `seconds` to `centuries`. A `source timestamps` option is also available as step: when selected, this causes the temporal navigation to step between all available time ranges from layers in the project. It's useful when a project contains layers with non-contiguous available times, such as a WMS-T service which provides images that are available at irregular dates. This option will allow you to only step between time ranges where the next available image is shown.
4. Clique no botão para visualizar a animação. QGIS irá gerar cenas usando as camadas de renderização nos horários definidos. A exibição das camadas depende de se elas se sobrepõem a qualquer período de tempo individual.

   The animation can also be previewed by moving the time slider. Checking the *Loop* checkbox will repeatedly run the animation while clicking stops a running animation. A full set of video player buttons is available.

   Horizontal scrolling using the mouse wheel (where supported) with the cursor on the map canvas will also allow you to navigate, or "scrub", the temporal navigation slider backwards and forwards.
5. Click the Export animation button if you want to generate a series of images representing the scene. They can be later combined in a video editor software:

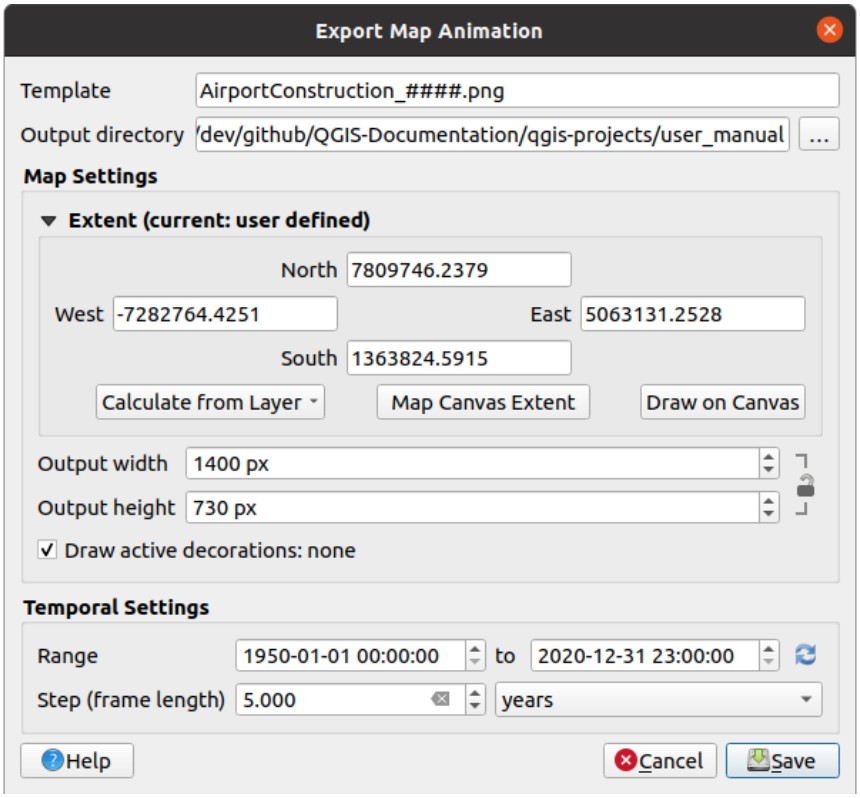


Figura7.3: Exporting map canvas animation scenes to images

- O nome do arquivo *Modelo*: o #### é substituído pelo número da seqüência de quadros
- O *Diretório de saída*
- Sob :guilabel:`Configurações do mapa', você pode:
  - redefinir a :ref:`extensão espacial <extent_selector>`a ser utilizada
  - control the *Resolution* of the image (*Output width* and *Output height*)
  - *Draw active decorations*: whether active *decorations* should be kept in the output
- Sob :guilabel:`Configurações temporais', você pode redefinir:
  - the time *Range* for the animation
  - the *Step (frame length)* in the unit of your choice

7.1.4 The Elevation Controller panel

Elevation Controller allows you to handle elevation z-range data in 2D maps. This option currently supports point cloud layers and raster DEMs. Use the panel to visualize and interact with elevation data in the map canvas.

Activate the *Elevation Controller* panel in the map canvas by clicking on the *Data Filtering* in the *View* menu. The *Elevation Controller* appears as a range slider on the left side of the map canvas. At the top of the slider, there is a Settings menu where you can:

- Set the *Fixed Range Size* value to lock the elevation range to a specific size. This setting is saved within the project and retained across sessions, ensuring consistency for your elevation filter. Use the *Invert Direction* option to reverse the elevation filter slider, so that it moves from high values at the bottom to low values at the top.

- *Set Elevation Range* to open the project elevation tab, where you can set dedicated lower and upper bounds for the elevation range
- *Disable Elevation Filter* to remove the elevation filter from the map canvas

7.1.5 Bookmarking extents on the map

Spatial Bookmarks allow you to "bookmark" a geographic location and return to it later. By default, bookmarks are saved in the user's profile (as *User Bookmarks*), meaning that they are available from any project the user opens. They can also be saved for a single project (named *Project Bookmarks*) and stored within the project file, which can be helpful if the project is to be shared with other users.

Novo Favorito

Para criar um novo favorito:

1. Zoom and pan to the area of interest.
2. Select the menu option *View* ► *New Spatial Bookmark…*, press `Ctrl+B` or right-click the *Spatial Bookmarks* entry in the *Browser* panel and select *New Spatial Bookmark*. The *Bookmark Editor* dialog opens.

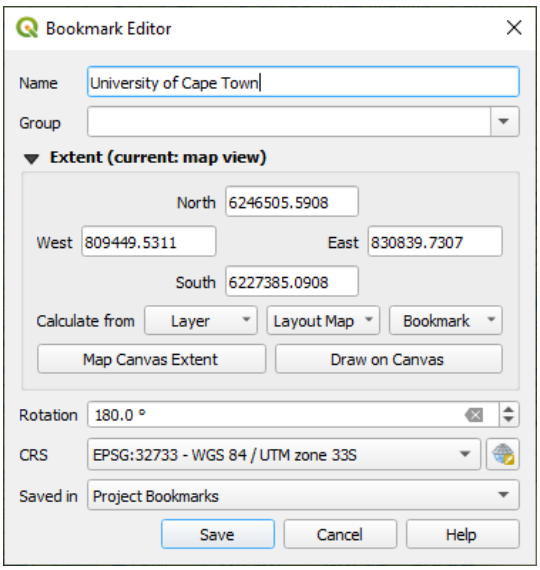


Figura7.4: The Bookmark Editor Dialog

3. Enter a descriptive name for the bookmark
4. Enter or select a group name in which to store related bookmarks
5. Select the extent of the area you wish to save, using the *extent selector* widget
6. Change the map *Rotation*
7. Indicate the *CRS* to use for the extent
8. Select whether the bookmark will be *Saved in User Bookmarks* or *Project Bookmarks* (by default, this drop-down list is set to *User Bookmarks*)
9. Press *Save* to add the bookmark to the list

Note que você pode ter vários favoritos com o mesmo nome.

Trabalhando com favoritos

To use and manage bookmarks, you can either use the *Spatial Bookmarks* panel or *Browser*:

- Select *View* ► *Show Spatial Bookmark Manager* or press `Ctrl+7` to open the *Spatial Bookmark Manager* panel.
- Select *View* ► *Show Spatial Bookmarks* or press `Ctrl+Shift+B` to show the *Spatial Bookmarks* entry in the *Browser* panel.

Você pode executar as seguintes tarefas:

Tabela7.2: Managing bookmark actions

Tarefa	Spatial Bookmark Manager	Navegador
Zoom to a Bookmark	Double-click on it, or select the bookmark and press the *Zoom to bookmark* button.	Double-click on it, drag and drop it to the map canvas, or right-click the bookmark and select *Zoom to Bookmark*.
Delete a bookmark	Select the bookmark and click the *Delete bookmark* button. Confirm your choice.	Right-click the bookmark and select *Delete Spatial Bookmark*. Confirm your choice.
Export bookmarks to XML	Click the *Import/Export Bookmarks* button and select *Export*. All the bookmarks (user or project) are saved in an xml file.	Select one or more folders (user or project) or subfolders (groups), then right-click and select *Export Spatial Bookmarks…*. The selected bookmark subset is saved.
Import bookmarks from XML	Click the *Import/Export Bookmarks* button and select *Import*. All bookmarks in the XML file are imported as user bookmarks.	Right-click the *Spatial Bookmarks* entry or one of its folders (user or project) or subfolders (groups) to determine where to import the bookmarks, then select *Import Spatial Bookmarks*. If performed on the *Spatial Bookmarks* entry, the bookmarks are added to *User Bookmarks*.
Edit bookmark	You can change a bookmark by changing the values in the table. You can edit the name, the group, the extent and if it is stored in the project or not.	Right-click the desired bookmark and select *Edit Spatial Bookmark…*. The *Bookmark Editor* will open, allowing you to redefine every aspect of the bookmark as if you were creating it for the first time. You can also drag and drop the bookmark between folders (user and project) and subfolders (groups).

You can manage bookmark actions by right-click on the desired bookmark in the *Spatial Bookmarks Manager*. You can also zoom to bookmarks by typing the bookmark name in the *locator*.

7.1.6 Decorating the map

Decorations include Grid, Title Label, Copyright Label, Image, North Arrow, Scale Bar and Layout Extents. They are used to 'decorate' the map by adding cartographic elements.

Malha

Grid allows you to add a coordinate grid and coordinate annotations to the map canvas.

1. Select menu option *View* ► *Decorations* ► *Grid…* to open the dialog.

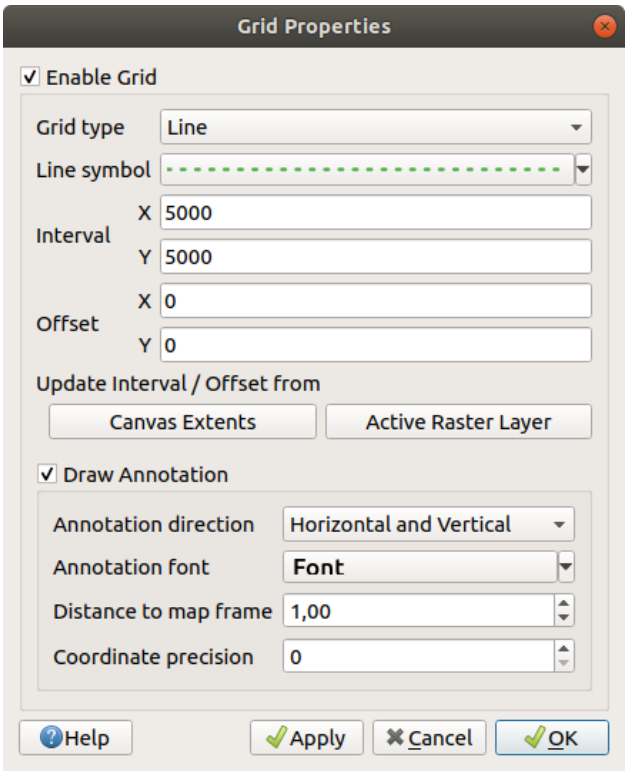


Figura7.5: The Grid Dialog

2. Tick *Enable grid* and set grid definitions according to the layers loaded in the map canvas:
   - The *Grid type*: it can be *Line* or *Marker*
   - The associated *Line symbol* or *marker symbol* used to represent the grid marks
   - The *Interval X* and *Interval Y* between the grid marks, in map units
   - An *Offset X* and *Offset Y* distance of the grid marks from the bottom left corner of the map canvas, in map units
   - The interval and offset parameters can be set based on the:
     - *Canvas Extents*: generates a grid with an interval that is approximatively 1/5 of the canvas width
     - *Active Raster Layer* resolution
3. Tick *Draw annotations* to display the coordinates of the grid marks and set:
   - The *Annotation direction*, ie how the labels would be placed relative to their grid line. It can be:
     - *Horizontal* or *Vertical* for all the labels

  - *Horizontal and Vertical*, ie each label is parallel to the grid mark it refers to
  - *Boundary direction*, ie each label follows the canvas boundary, and is perpendicular to the grid mark it refers to
- The *Annotation font* (text formatting, buffer, shadow…) using the *font selector widget*
- The *Distance to map frame*, margin between annotations and map canvas limits. Convenient when *exporting the map canvas* eg to an image format or PDF, and avoid annotations to be on the "paper" limits.
- The *Coordinate precision*

4. Click *Apply* to verify that it looks as expected or *OK* if you're satisfied.

Title Label

Title Label allows you to decorate your map with a **Title**.

To add a Title Label decoration:

1. Select menu option *View* ► *Decorations* ► *Title Label…* to open the dialog.

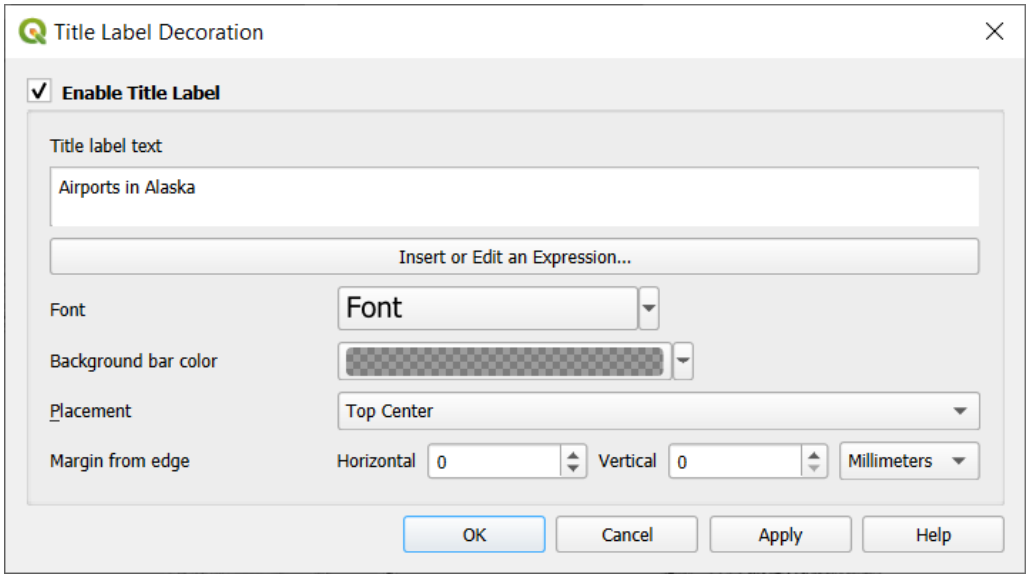


Figura7.6: The Title Decoration Dialog

2. Make sure *Enable Title Label* is checked
3. Enter the title text you want to place on the map. You can make it dynamic using the *Insert or Edit an Expression…* button.
4. Choose the *Font* for the label using the *font selector widget* with full access to QGIS *text formatting* options. Quickly set the font color and opacity by clicking the black arrow to the right of the font combo box.
5. Select the *color* to apply to the title's *Background bar color*.
6. Choose the *Placement* of the label in the canvas: options are *Top left*, *Top Center* (default), *Top Right*, *Bottom left*, *Bottom Center* and *Bottom Right*.
7. Refine the placement of the item by setting a horizontal and/or vertical *Margin from Edge*. These values can be in **Millimeters** or **Pixels** or set as a **Percentage** of the width or height of the map canvas.
8. Click *Apply* to verify that it looks as expected or *OK* if you're satisfied.

Copyright Label

Copyright Label can be used to decorate your map with a **Copyright** label.

Para adicionar esta decoração:

1. Select menu option *View* ► *Decorations* ► *Copyright Label…* to open the dialog.

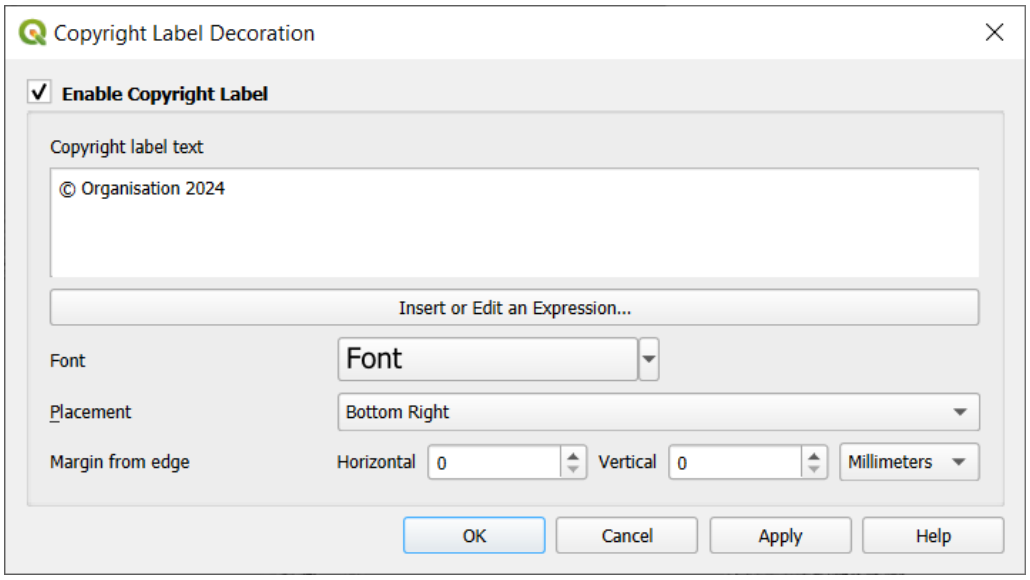


Figura7.7: The Copyright Decoration Dialog

2. Make sure *Enable Copyright Label* is checked
3. Enter the copyright text you want to place on the map. You can make it dynamic using the *Insert or Edit an Expression…* button.
4. Choose the *Font* for the label using the *font selector widget* with full access to QGIS *text formatting* options. Quickly set the font color and opacity by clicking the black arrow to the right of the font combo box.
5. Choose the *Placement* of the label in the canvas: options are *Top left*, *Top Center*, *Top Right*, *Bottom left*, *Bottom Center*, and *Bottom Right* (default for Copyright decoration)
6. Refine the placement of the item by setting a horizontal and/or vertical *Margin from Edge*. These values can be in **Millimeters** or **Pixels** or set as a **Percentage** of the width or height of the map canvas.
7. Click *Apply* to verify that it looks as expected or *OK* if you're satisfied.

Image Decoration

Image allows you to add an image (logo, legend, ..) on the map canvas.

Para adicionar uma imagem:

1. Select menu option *View* ► *Decorations* ► *Image…* to open the dialog.

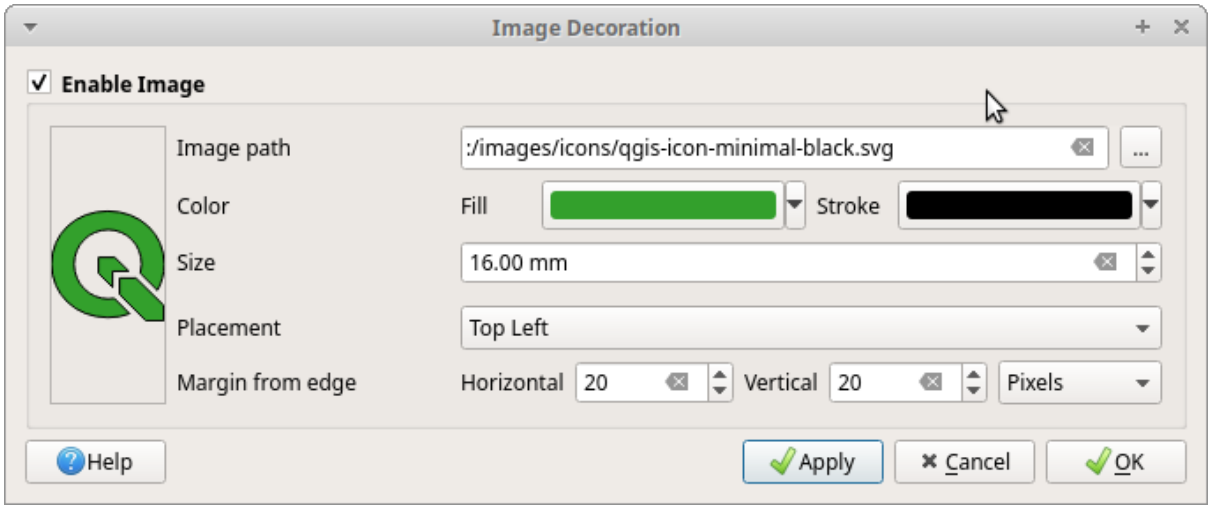


Figura7.8: The Image Decoration Dialog

2. Make sure *Enable Image* is checked
3. Select a bitmap (e.g. png or jpg) or SVG image using the … Browse button
4. If you have chosen a parameter enabled SVG then you can also set a *Fill* or *Stroke* (outline) color. For bitmap images, the color settings are disabled.
5. Set a *Size* of the image in mm. The width of selected image is used to resize it to given *Size*.
6. Choose where you want to place the image on the map canvas with the *Placement* combo box. The default position is *Top Left*.
7. Set the *Horizontal* and *Vertical Margin from (Canvas) Edge*. These values can be set in **Millimeters**, **Pixels** or as a **Percentage** of the width or height of the map canvas.
8. Click *Apply* to verify that it looks as expected and *OK* if you're satisfied.

Seta Norte

North Arrow allows you to add a north arrow on the map canvas.

Para adicionar uma seta norte:

1. Select menu option *View* ► *Decorations* ► *North Arrow*… to open the dialog.

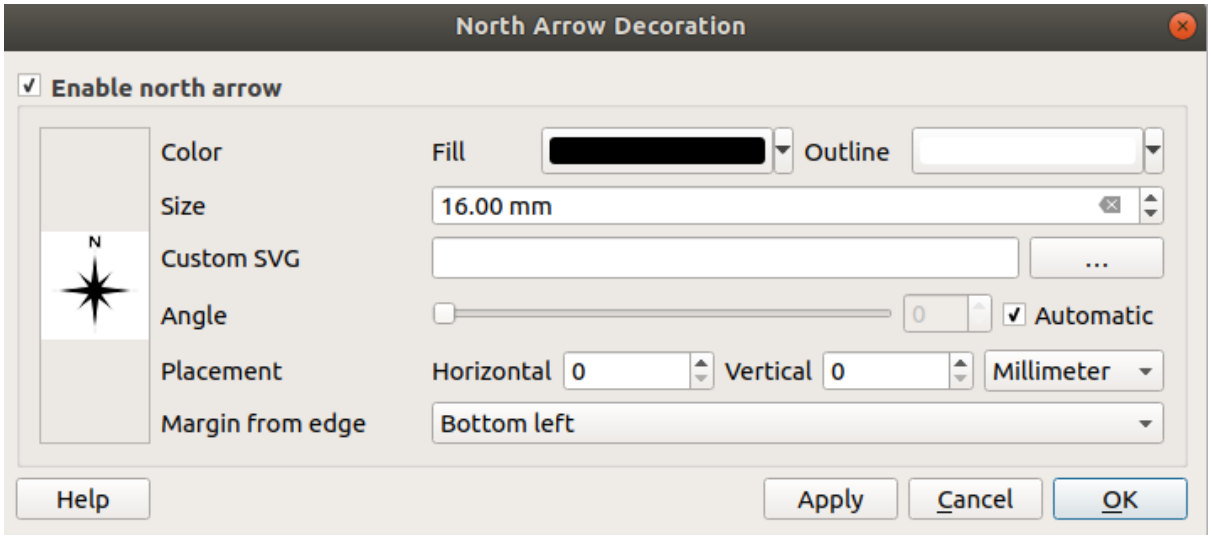


Figura7.9: The North Arrow Dialog

2. Make sure *Enable north arrow* is checked
3. Optionally change the color and size, or choose a custom SVG
4. Opcionalmente, altere o ângulo ou escolha **Automático** para permitir que o QGIS determine a direção
5. Optionally choose the placement from the Placement combo box
6. Optionally refine the placement of the arrow by setting a horizontal and/or vertical *Margin from (Canvas) Edge*. These values can be in **Millimeters** or **Pixels** or set as a **Percentage** of the width or height of the map canvas.
7. Click *Apply* to verify that it looks as expected and *OK* if you're satisfied.

Barra de Escala

Scale Bar adds a simple scale bar to the map canvas. You can control the style and placement, as well as the labelling of the bar. The scale bar respects the active *project's distance unit* as defined in *Project properties* ► *General* ► *Units for distance measurement*.

Para adicionar uma barra de escala:

1. Select menu option *View* ► *Decorations* ► *Scale Bar…* to open the dialog

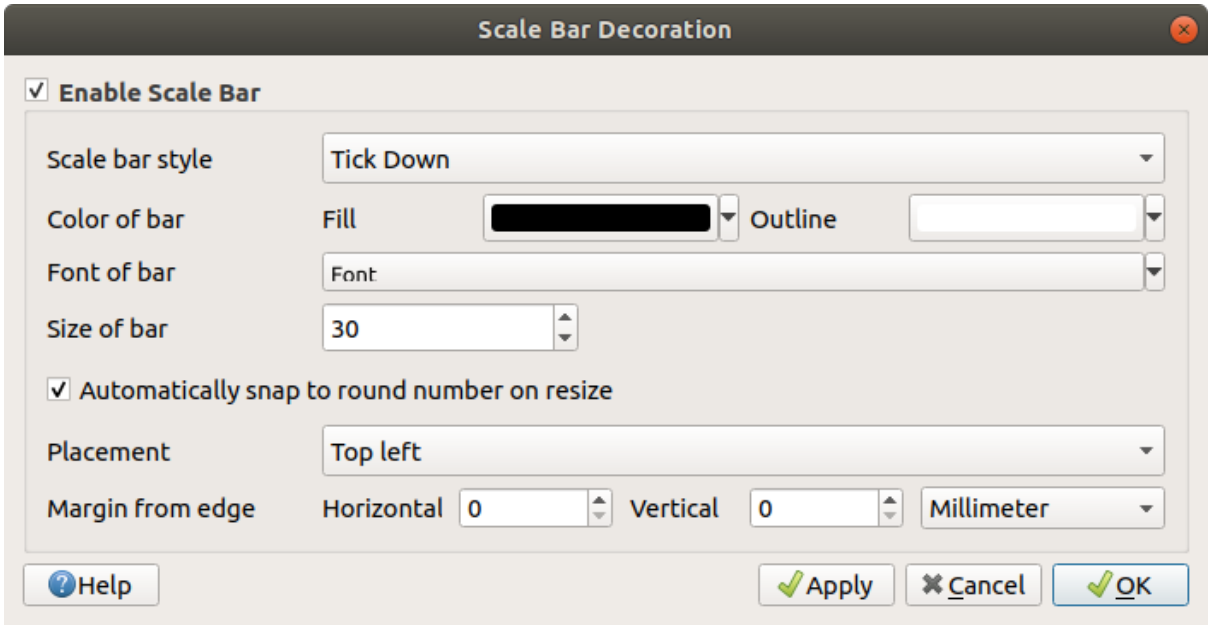


Figura7.10: The Scale Bar Dialog

2. Make sure *Enable scale bar* is checked
3. Choose a style from the *Scale bar style* combo box
4. Select the *Color of bar* by choosing a fill color (default: black) and an outline color (default: white). The scale bar fill and outline can be made opaque by clicking on the down arrow to the right of the color input.
5. Selecione a fonte para a barra de escala na caixa *Fonte da barra*
6. Set the *Size of bar* in the active unit
7. Optionally check *Automatically snap to round number on resize* to display easy-to-read values
8. Choose the placement from the *Placement* combo box
9. You can refine the placement of the item by setting a horizontal and/or vertical *Margin from (Canvas) Edge*. These values can be in **Millimeters** or **Pixels** or set as a **Percentage** of the width or height of the map canvas.
10. Click *Apply* to verify that it looks as expected or *OK* if you're satisfied.

Extensões de Layout

Layout Extents adds the extents of *map item(s)* in print layout(s) to the canvas. When enabled, the extents of all map items within all print layouts are shown using a lightly dotted border labeled with the name of the print layout and map item. You can control the style and labeling of the displayed layout extents. This decoration is useful when you are tweaking the positioning of map elements such as labels, and need to know the actual visible region of print layouts.

Figura7.11: Example of layout extents displayed in a QGIS project with two print layouts. The print layout named 'Sights' contains two map items, while the other print layout contains one map item.

Para adicionar extensão(ões) de layout:

1. Select *View* ► *Decorations* ► *Layout Extents* to open the dialog

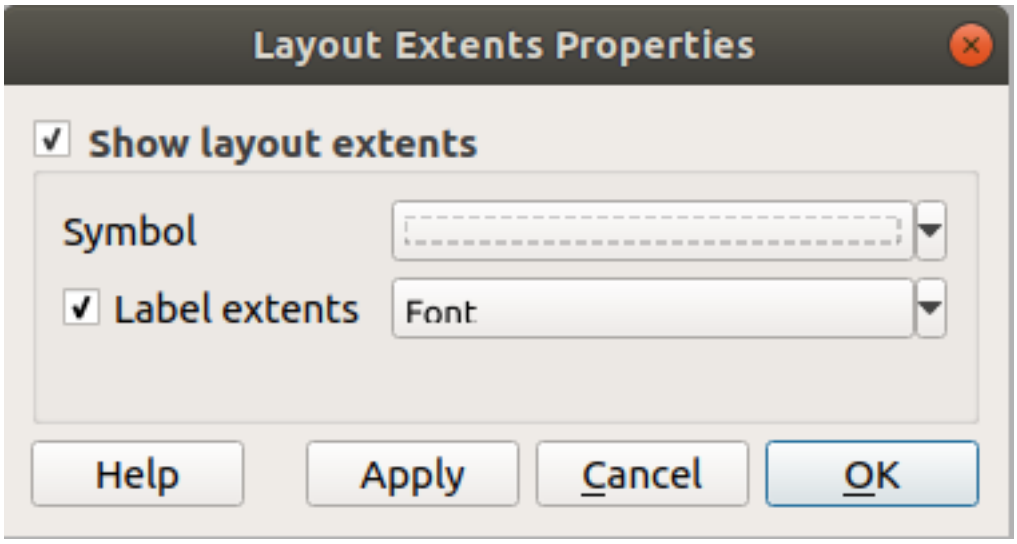


Figura7.12: The Layout Extents Dialog

2. Make sure *Show layout extents* is checked.

3. Opcionalmente, altere o símbolo e a rotulagem das extensões.
4. Click *Apply* to verify that it looks as expected and *OK* if you're satisfied.

Dica

Decorations Settings

When you save a QGIS project file, any changes you have made to Grid, North Arrow, Scale Bar, Copyright and Layout Extents will be saved in the project and restored the next time you load the project.

7.1.7 Ferramentas de anotação

Annotations are another type of elements added onto the map canvas to provide additional information that can not be depicted by the rendered layers. Unlike *labels* that rely on attribute values stored in vector layers, annotations are independent details, stored within the project file itself.

Two families of annotations are available in QGIS:

- **Feature annotations**: they are actual georeferenced features of text, marker, line or polygon type stored within a special layer type called "annotation layer". They are tied to a particular geographic location, meaning that moving your map, changing the scale or changing projection won't cause your annotations to jump around the map. Rather, they'll be locked in place to the location you've drawn them.
- **Balloon annotations**: these are individual HTML or form annotations inside a bubble. They can be associated to any layer for their visibility and drawn on top of the map canvas. The size is dependent from the map canvas scale, and its position can be anchored.

Dica

Layout the map with annotations

You can print or export annotations with your map to various formats using:

- map canvas export tools available in the *Project* menu
- *print layout*, in which case you need to check *Draw map canvas items* in the corresponding map item properties

The *Annotations Toolbar* provides a set of tools to create and interact with both families of annotations.

Tabela7.3: The Annotations Toolbar actions

Ferramenta	Utilização	Scope
New Annotation Layer	Create a new layer to store annotations	*Feature annotations*
Main Annotation Layer Properties	Control settings of the Main Annotation Layer	
Modify Annotations	Select, move, resize and modify symbology properties of annotations	
Create Polygon Annotation	Create an annotation as a polygon feature	
Create Line Annotation	Create an annotation as a polyline feature	
Create Marker Annotation	Create an annotation as a point feature	
Create Text Annotation at Point	Create an annotation as a text label anchored at a point	
Create Text Annotation along Line	Create an annotation as a curved text along a line feature	

continua na próxima página

Tabela 7.3 – continuação da página anterior

Ferramenta	Utilização	Scope
Create Text Annotation in Rectangle	Create an annotation as a text label in a rectangular text box	*Balloon annotations*
Create Picture Annotation	Create an annotation representing an `.SVG` or raster image	
HTML Annotation	Select and create annotation with a `HTML` formatted text	
Form Annotation	Select and create annotation showing attributes of a vector layer in a custom form file	

Nota

Starting with QGIS 3.40, it is no longer possible to create balloon annotations of SVG or text type. When loading old projects, any of these annotations will automatically be converted to the newer picture and rectangular text annotation item types.

Feature Annotations

Feature annotations are stored in **annotation layers**. Unlike conventional layers, an annotation layer is available only in the current project and can contain features of different types (text, marker, line, polygon, picture). The layer has no attributes and no symbology associated, but instead each feature can be symbolized on an item-by-item basis, through *Layer Styling* panel.

Two types of annotation layer are available in QGIS:

- A common *Annotation Layer*: you can create one using the New Annotation Layer tool. It is listed in the *Layers* panel, allowing you to control its features' visibility, move it to show above or below particular layers in your map, like any common layer. Double-click the layer and you can access its properties.
- The *Main Annotation Layer*: By default, this is where annotations are stored when no annotation layer is available in the project or is selected at creation time. This layer is always drawn on the very top of your map and you won't see it listed in the *Layers* panel alongside the other layers in your project, meaning that its features are always visible. The *Main Annotation Layer Properties* entry on the *Annotations* toolbar helps you open its properties dialog.

Propriedades da Camada

The properties dialog of an annotation layer provides the following tabs:

- *Information*: a read-only dialog representing an interesting place to quickly grab summarized information and metadata on the current layer. This may include the layer extent, count of items per annotation type and total count, CRS details, …
- *Source*: defines general settings for the annotation layer. You can:
  - Set a *Layer name* that will be used to identify the layer in the project (in the *Layers Panel*, with expressions, …)
  - Display the layer's *Assigned Coordinate Reference System (CRS)*: you can change the layer's CRS, selecting a recently used one in the drop-down list or clicking on Select CRS button (see *Seletor do Sistemas de Referência de Coordenadas*). Use this process only if the CRS applied to the layer is a wrong one or if none was applied.
- *Rendering*:

- You can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale, defining a range of scale in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Visibility Scale Selector* for more information.
- *Linked layer*: allows you to optionally set a linked visibility layer for the annotation layer. If set, then the annotations will only be drawn when the linked layer is visible in the map.
- *Opacity*: You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your vector layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Blending mode* at the *Layer* level: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlying layers are mixed through the settings described in *Modos de Mistura*.
- Aplique:ref:*efeitos de pintura<draw_effects>* em todos os recursos da camada com o botão:guilabel: *Efeitos de desenho*.

Some of these options are accessible from the feature annotation *Symbology* properties.

Interagindo com feições

The Feature annotations have dedicated tools for creation depending on their type.

Tabela7.4: Creating feature annotation

Family / based on	Annotation tool	Procedimento
Geometry	Create Polygon Annotation	This is like digitizing a point, line or polygon vector feature. All the usual shortcuts for creating features apply when creating annotation items. A line or polygon annotation is drawn by left-clicking once for each vertex, with a final right mouse click to complete the shape. Snapping can be enabled while you draw, you can use the *Advanced Digitizing Tools* to precisely place vertices, and even switch the *drawing tools* to for example the streaming mode for completely free-form shapes.
	Create Line Annotation	
	Create Marker Annotation	
Text	Create Text Annotation at Point	Left-click on the map canvas.
	Create Text Annotation along Line	Left-click to add vertices to the text basement line geometry, and right-click to end. As above, digitizing features capabilities are also available.
	Create Text Annotation In Rectangle	Draw the rectangle box of the text, by left-clicking to add a first corner and left-clicking again to add the opposite one.
Picture	Create Picture Annotation	Draw the rectangle box of the picture, by left-clicking to add a first corner and left-clicking again to add the opposite one. Then pick an image or SVG file from the file explorer.

Unlike common layers, an annotation layer does not need to be active before you select its features. Simply grab the Modify Annotations tool and you will be able to interact with any feature annotation:

- **Selection**: left-click on the annotation. By default, annotations are rendered in the order of creation, meaning that recent annotations will be placed on top of older. You may need to play with the Z-index property of features in order to select ones they would sit above.

- **Moving**: Left click on a selected annotation item to start moving it. A right-click or pressing `Esc` key cancels the move, while a second left click will confirm the move. The displacement can also be controlled pressing the cursor keys:
  - `Shift+key` for big movement
  - `Alt+key` for `1 px` movement
- **Geometry modification**: move over an annotation and purple squares are displayed on nodes of its underlying geometry. Left-click on the square, move and click again. For linear or polygonal annotations, double-clicking a segment will add a new vertex.
- **Delete**: Pressing the `Del` or `Backspace` key while an annotation is selected will delete that annotation
- *Change feature symbology*

Feature symbology

A selected annotation will display its *Symbology* properties in the *Layer styling* panel.

- For polygon, polyline and marker annotations, you can modify the appearance using full capabilities of the *symbol properties*
- For text-based annotations, an area with text editing tools helps you construct the string to display. It can be multiline, use HTML formatting and rely on QGIS expression functions. You can moreover modify the appearance using full capabilities of the *text format properties*.

  Depending on the type of annotation, additional options are available.
  - For text annotation at point, it is also possible to:
    - set the text *Alignment* (left, center or right of the anchor point)
    - configure whether the text should *Ignore map rotation* or *Rotate with map*. In both cases, a custom *Angle* can be set for the feature orientation.
  - For text annotation along a line, configure an *Offset from line* in the unit of your choice
  - For text annotation in rectangle, it is also possible to:
    - set the text *Horizontal alignment* (left, center, right or justify) and *Vertical alignment* (top, vertical center or bottom) in the rectangle box
    - configure for the rectangle, the *Margins* distance in the units of your choice, the *Frame* and *Background* colors.
- Picture annotations allow you to display a *Raster image* or an *SVG image*, using the *Seletor de arquivos remoto ou incorporado*. The placement and size of the picture can be set as:
  - *Scale dependent size*: the picture scales with the map, appearing larger when zooming in and smaller when zooming out. The size is defined by a single reference size in map units (e.g. meters or degrees, depending on the layer's CRS). With *Lock aspect ratio*, you can keep the annotation size proportional to the embedded picture size, or stretch the picture to fill the annotation extent.
  - *Fixed size*: the picture annotation maintains a constant size in screen or output units (e.g. pixels or millimeters), regardless of the map scale. The *Width* and *Height* of the annotation can be filled separatey or adjusted proportionally to the embedded picture size.
  - *Relative to map*: the picture annotation will always be rendered at the same position relative to the map canvas bounds, regardless of the map scale. The *Width* and *Height* of the annotation can be filled separatey or adjusted proportionally to the embedded picture size.

  Moreover, you can enable display of the *Frame* and *Background* of the rectangle box, and configure them using *fill symbols*.
- For text at point, text inside rectangle and picture annotations, you can *Show callout* when the picture or text anchor point is offset from its default placement. To create a callout for an annotation:

1. Select the annotation item
2. Move the anchor point, i.e., click the central and green `X` node and click at the new placement
3. In the annotation properties, press the … button to configure the *callout properties*.

> **Dica**
>
> Applying a balloon callout to a text or picture annotation will make you get their old-style balloon annotation display.

- Configure a *Reference scale*: indicates the map scale at which symbol or text sizes which use paper-based units (such as millimeters or points) relate to. The sizes will be scaled accordingly whenever the map is viewed at a different scale. For instance, a line feature wide of 2mm at 1:2000 *Reference scale* will be rendered using 4mm when the map is viewed at 1:1000.
- Set a *Z-index*: a feature with a higher index is placed on top of features with lower index. A convenient setting for both feature display and selection.
- Modify some of the *Layer rendering* settings

Balloon annotations

You can add balloon annotations through *Edit* ► *Add Annotation* ► menu or from the *Annotations Toolbar*:

- HTML Annotation to display a HTML-formatted string or a whole `HTML` file as an annotation.
- Form Annotation: useful to display attributes of a vector layer in a customized `.ui` file. This is similar to the *custom attribute forms*, but displayed in an annotation item.

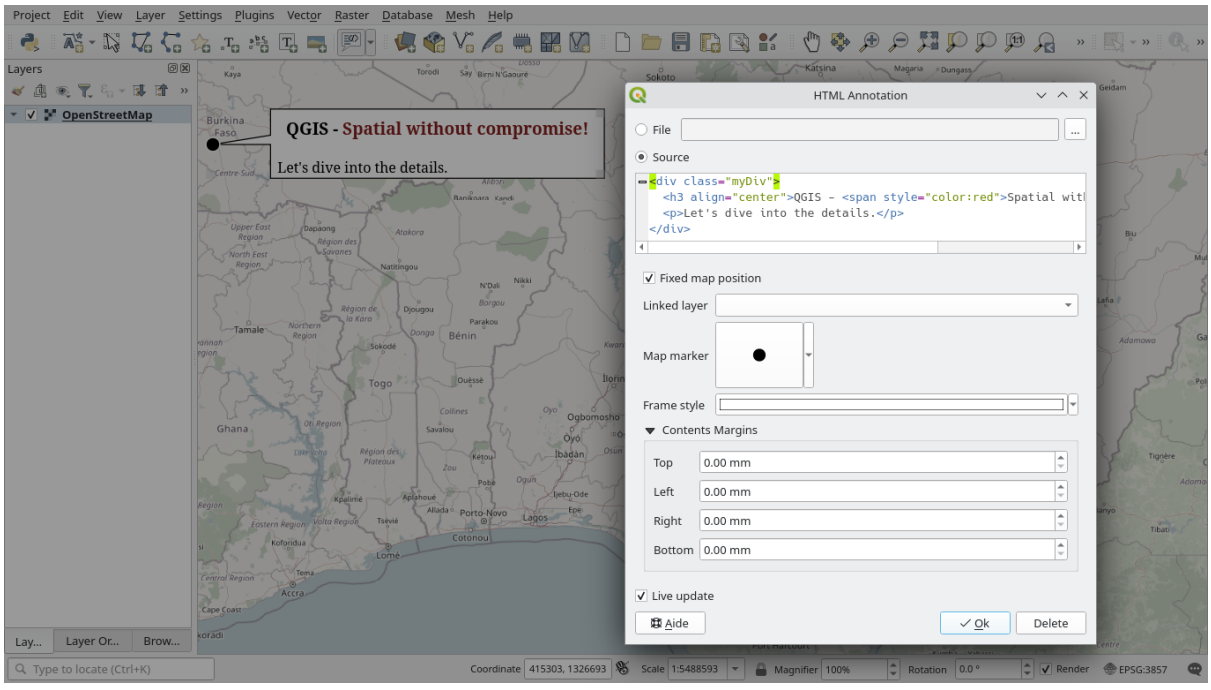


Figura7.13: Example of a HTML balloon annotation

To add a balloon annotation, select the corresponding tool and click on the map canvas. An empty balloon is added. Double-click on it and a dialog opens with various options. This dialog is almost the same for all the annotation types:

- *File*: Fill it with the path to a `.html` or `.ui` file, depending on the type of annotation, to serve as contents of the annotation.
- *Source*: Only available for HTML annotations, it allows to serve the annotation with a plain or HTML-formatted text instead of a file.
- *Fixed map position*: when unchecked, the balloon placement is based on a screen position (instead of the map), meaning that it's always shown regardless of the map canvas extent.
- *Linked layer*: associates the annotation with a map layer, making it visible only when that layer is visible.
- *Map marker*: using *QGIS symbols*, sets the symbol to display at the balloon anchor position (shown only when *Fixed map position* is checked).
- *Frame style*: sets the frame background color, transparency, stroke color or width of the balloon using QGIS symbols.
- *Contents margins*: sets interior margins of the annotation frame.
- *Live update* allows you to live preview your changes.

When a balloon annotation tool is active, you can also:

- Select an annotation
- Resize an annotation
- Move an annotation by map position (by dragging the map marker) or by moving only the balloon.
- Delete an annotation: select it and either press the `Del` or `Backspace` button, or double-click it and press the *Delete* button in the properties dialog.
- Right-click and in the contextual menu:
  - *Copy coordinate* of the annotation's map marker in various CRS
  - *Edit* properties of the annotation. Same as double-clicking the annotation.
  - *Delete* the annotation

7.1.8 Medição

Informação Geral

O QGIS fornece quatro meios de medição de geometrias:

- interactive measurement tools
- measuring in the Field Calculator
- derived measurements in the *Identificando Feições* tool
- the vector analysis tool: *Vector* ► *Geometry Tools* ► *Export/Add Geometry Columns*

Measuring works within projected coordinate systems (e.g., UTM) and unprojected data. The first three measuring tools behave equally to global project settings:

- Diferentemente da maioria dos outros GIS, a métrica de medição padrão é elipsoidal, usando o elipsóide definido em *Projeto* ► *Propriedades...* ► *Geral*. Isso ocorre quando os sistemas de coordenadas geográficas e projetadas são definidos para o projeto.
- If you want to calculate the projected/planimetric area or distance using cartesian maths, the measurement ellipsoid has to be set to "None/Planimetric" (*Project* ► *Properties...* ► *General*). However, with a geographic (ie unprojected) CRS defined for the data and project, area and distance measurement will be ellipsoidal.

However, neither the identify tool nor the field calculator will transform your data to the project CRS before measuring. If you want to achieve this, you have to use the vector analysis tool: *Vector* ► *Geometry Tools* ► *Add Geometry Attributes*.... Here, measurement is planimetric, unless you choose the ellipsoidal measurement.

Measure length, areas, bearings and angles interactively

Click the icon in the Attribute toolbar to begin measurements. The down arrow near the icon switches between length, area, bearing or angle. The default unit used in the dialog is the one set in *Project* ► *Properties*... ► *General* menu.

For the *Measure Line* and the *Measure Area* the measurements can be done in *Cartesian* or *Ellipsoidal* measure.

Nota

Configurando a ferramenta de medição

While measuring length or area, clicking the *Configuration* button at the bottom of the widget opens the *Settings* ► *Options* ► *Map Tools* menu, where you can select the rubberband color, the precision of the measurements and the unit behavior. You can also choose your preferred measurement or angle units, but keep in mind that those values are overridden in the current project by the selection made in the *Project* ► *Properties*... ► *General* menu, and by the selection made in the measurement widget.

All measuring modules use the snapping settings from the digitizing module (see section *Setting the snapping tolerance and search radius*). So, if you want to measure exactly along a line feature, or around a polygon feature, first set its layer snapping tolerance. Now, when using the measuring tools, each mouse click (within the tolerance setting) will snap to that layer.

The Measure Line measures distances between given points. The tool then allows you to click points on the map. Each segment length, as well as the total, shows up in the measure window. In the measure window, you will see coordinates for all your points and distances. Keep in mind that the first row will contain only coordinates, as it represents your starting point. Now it is possible to copy all your line measurements at once to the clipboard using the *Copy* button. Clicking the *Configuration* button you will access to *Measure Tool Copy Settings* where you can set up *copy options*. To stop measuring, click the right mouse button.

Note that you can use the drop-down list near the total to change the *measurement units* interactively while working with the measure tool. This unit is retained for the widget until a new project is created or another project is opened.

The *Info* section in the dialog explains how calculations are made according to the CRS settings available.

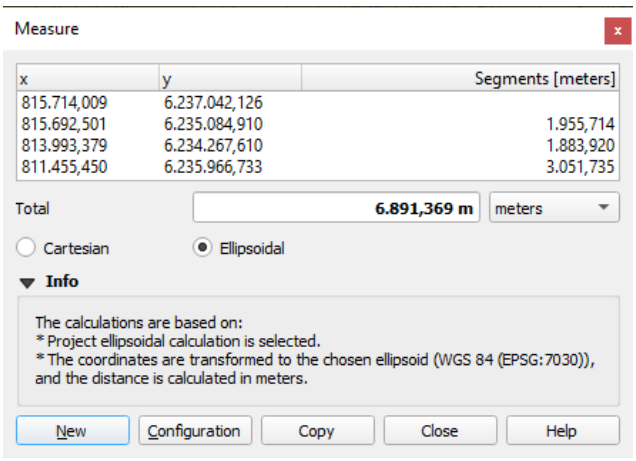


Figura7.14: Medir distância

Measure Area: Areas can also be measured. In the measure window, the accumulated area size appears. Right-click to stop drawing. The Info section is also available as well as the ability to switch between different *area units*.

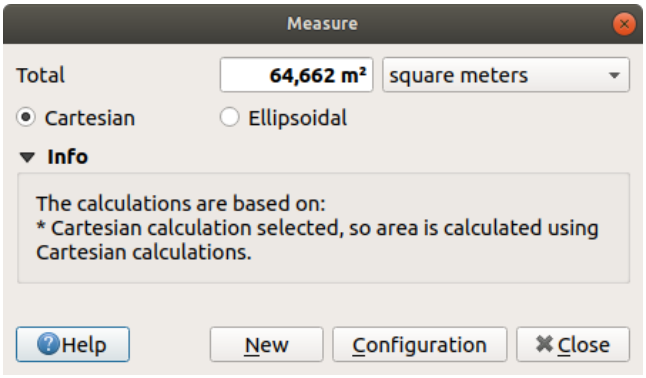


Figura7.15: Medir Área

Measure Bearing: You can also measure bearings. The cursor becomes cross-shaped. Click to draw the first point of the bearing, then move the cursor to draw the second point. The measurement is displayed in a pop-up dialog.

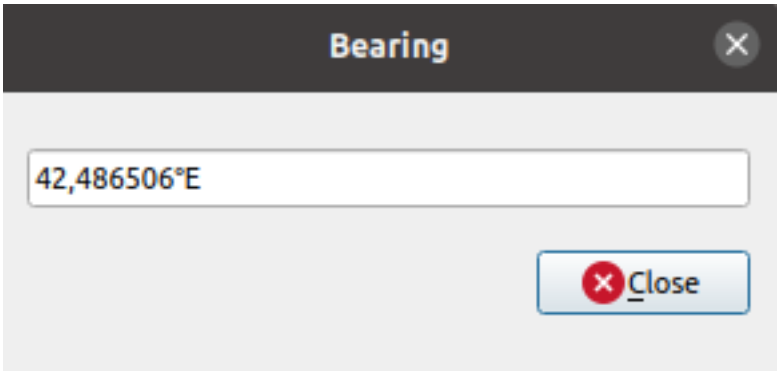


Figura7.16: Measure Bearing

Measure Angle: You can also measure angles. The cursor becomes cross-shaped. Click to draw the first segment of the angle you wish to measure, then move the cursor to draw the desired angle. The measurement is displayed in a pop-up dialog.

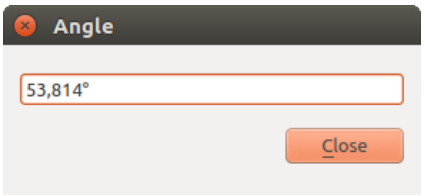


Figura7.17: Medir Ângulo

7.1.9 Definição de visualizações adicionais do mapa

It is also possible to open additional map views whose content could diverge from the *Layers* panel current state. To add a new map view, go to *View* ► *New Map View*. A new floating widget, mimicking the main map view's rendering, is added to QGIS. You can add as many map views as you need. They can be kept floating, placed side by side or stacked on top of each other.

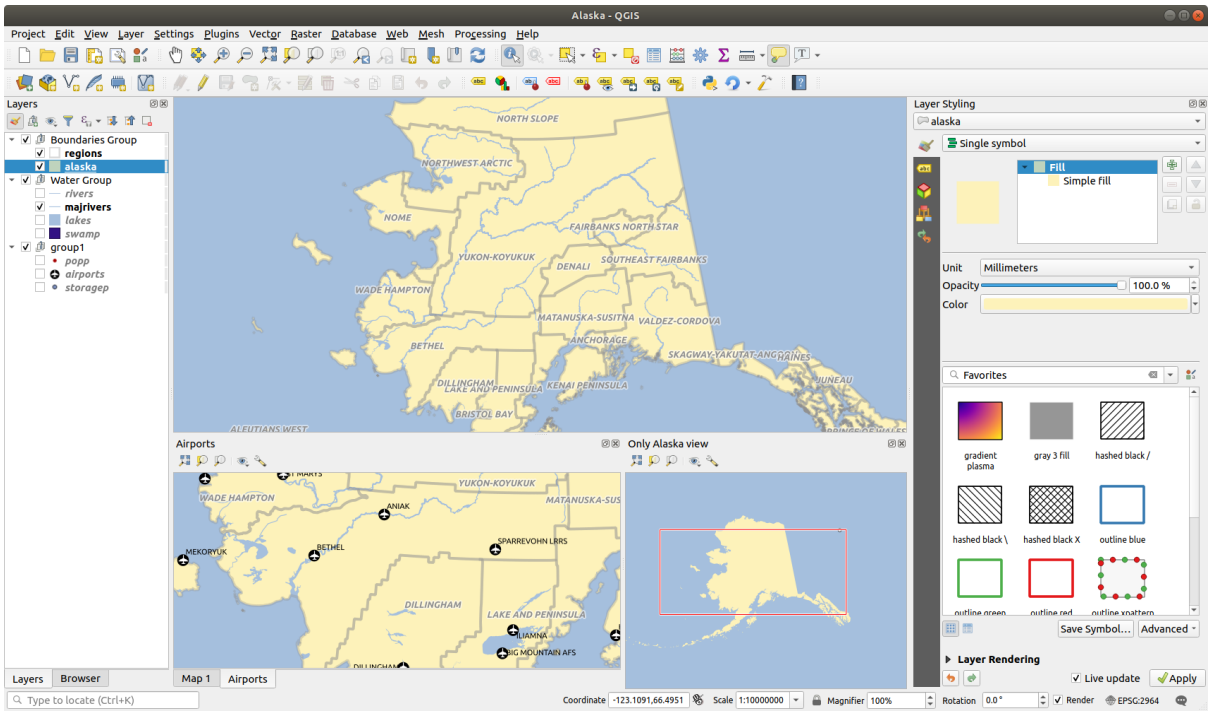


Figura7.18: Múltiplas visualizações de mapas com diferentes configurações

No topo de uma tela de mapa adicional, há uma barra de ferramentas com as seguintes capacidades:

- Zoom Full, Zoom to Selection and Zoom to Layer(s) to navigate within the view
- Set View Theme to select the *map theme* to display in the map view. If set to `(none)`, the view will follow the *Layers* panel changes.
- Configurações de visualização para configura a visualização do mapa:
  - *Synchronize view center with main map*: syncs the center of the map views without changing the scale. This allows you to have an overview style or magnified map which follows the main canvas center.
  - *Synchronize view to selection*: same as zoom to selection
  - *Escala*
  - *Rotação*
  - guilabel:`Magnificação'
  - *Synchronize scale* with the main map scale. A *Scale factor* can then be applied, allowing you to have a view which is e.g. always 2x the scale of the main canvas.
  - *Show annotations*
  - *Mostrar a posição do cursor*
  - *Show main canvas extent*

- *Show labels*: allows to hide labels regardless they are set in the displayed layers' properties
- *Mudar SRC do mapa…*
- *Renomear visualização…*

7.1.10 Exportando a visualização do mapa

Maps you make can be layout and exported to various formats using the advanced capabilities of the *print layout or report*. It's also possible to directly export the current rendering, without a layout. This quick "screenshot" of the map view has some convenient features.

To export the map canvas with the current rendering:

1. Vá para *Projeto* ► *Importar/Exportar*
2. Dependendo do seu formato de saída, selecione
   - *Export Map to Image…*
   - ou *Export Map to PDF…*

The two tools provide you with a common set of options. In the dialog that opens:

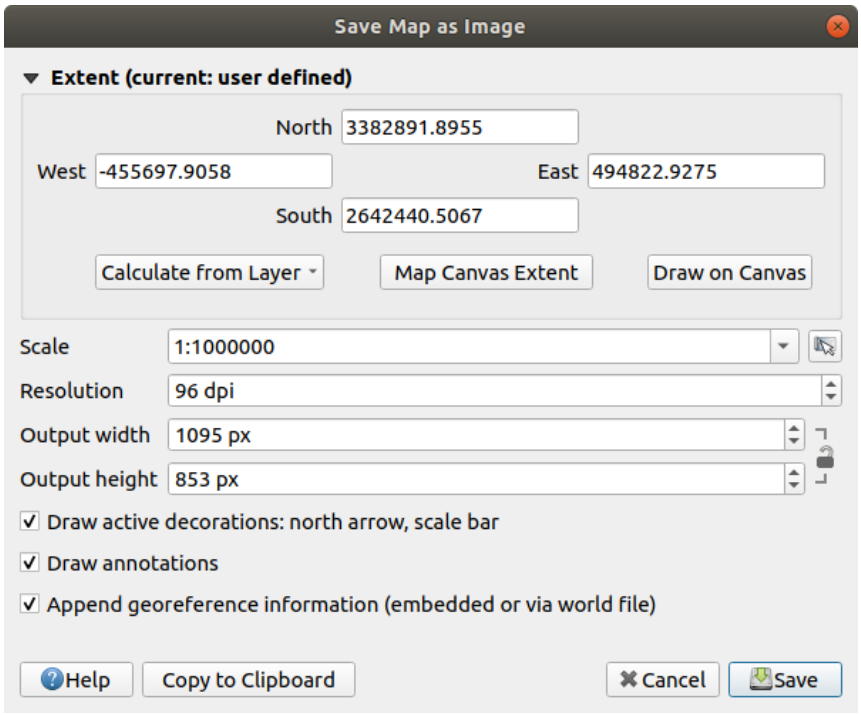


Figura7.19: The Save Map as Image dialog

1. Choose the *Extent* to export: it can be the current view extent (the default), the extent of a layer or a custom extent drawn over the map canvas. Coordinates of the selected area are displayed and manually editable.
2. Enter the *Scale* of the map or select it from the *predefined scales*: changing the scale will resize the extent to export (from the center).
3. Defina a *Resolução* da saída
4. Control the *Output width* and *Output height* in pixels of the image: based by default on the current resolution and extent, they can be customized and will resize the map extent (from the center). The size ratio can be locked, which may be particularly convenient when drawing the extent on the canvas.

5. *Draw active decorations*: in use *decorations* (barra de escala, título, malha, Rosa dos ventos…) são exportados com o mapa
6. *Desenhar anotações* para exportar qualquer *annotation*
7. *Append georeference information (embedded or via world file)*: depending on the output format, a world file of the same name (with extension `PNGW` for `PNG` images, `JPGW` for `JPG`, …) is saved in the same folder as your image. The `PDF` format embeds the information in the PDF file.
8. Ao exportar para PDF, mais opções estão disponíveis na caixa de diálogo *Salvar mapa como PDF*…:

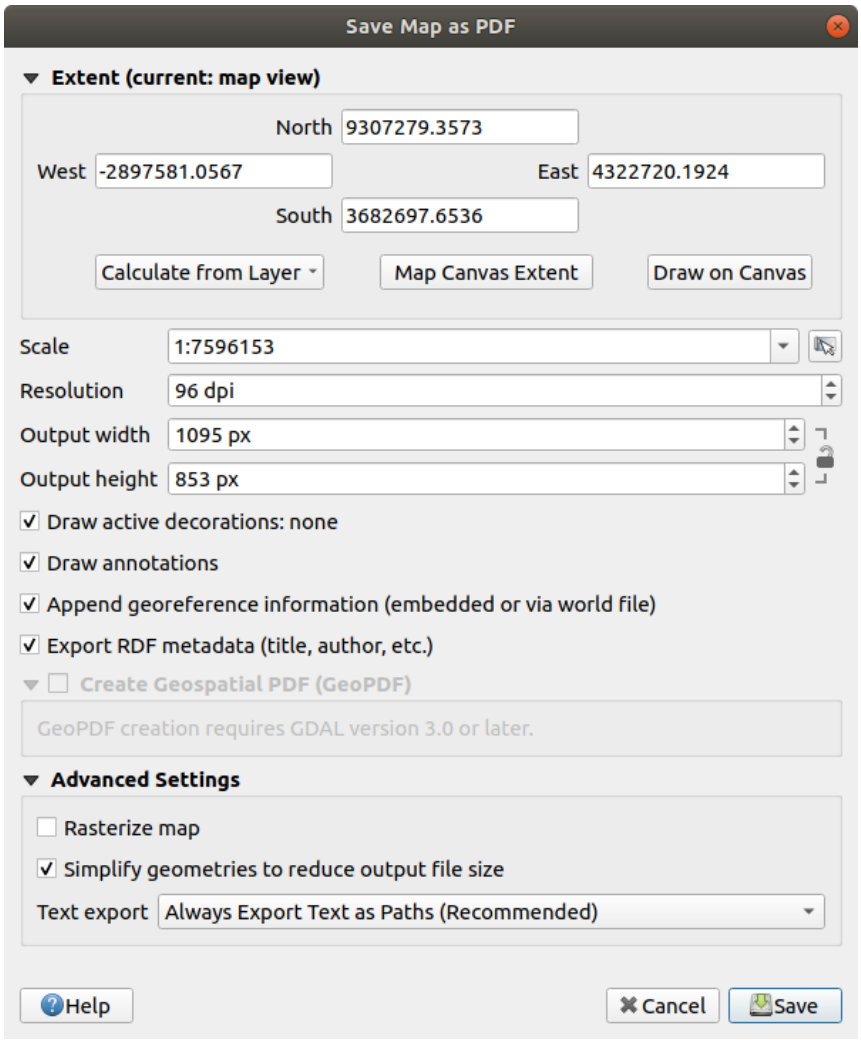


Figura7.20: A caixa de diálogo Salvar Mapa como PDF

- *Exportar metadados RDF* do documento, como título, autor, data, descrição…
- *Create Geospatial PDF*: Generate a georeferenced PDF file. You can:
  - Choose the Geospatial PDF *Format*
  - *Include vector feature information* in the Geospatial PDF file: will include all the geometry and attribute information from features visible within the map in the output Geospatial PDF file.

Nota

A Geospatial PDF file can also be used as a data source. For more on Geospatial PDF support in QGIS, see https://north-road.com/2019/09/03/qgis-3-10-loves-geospatialpdf/.

- *Rasterize map*
- *Simplify geometries to reduce output file size*: Geometries will be simplified while exporting the map by removing vertices that are not discernibly different at the export resolution (e.g. if the export resolution is `300 dpi`, vertices that are less than `1/600 inch` apart will be removed). This can reduce the size and complexity of the export file (very large files can fail to load in other applications).
- Set the *Text export*: controls whether text labels are always or preferably exported as *text or outline objects*.

9. Clique :guilabel:`Salvar' para selecionar o local, nome e formato do arquivo.

   When exporting to image, it's also possible to *Copy to clipboard* the expected result of the above settings and paste the map in another application such as LibreOffice, GIMP…

7.2 Visualização de Mapa 3D

3D visualization support is offered through the 3D map view. You can create, manage and open 3D map views via *View* ► *3D Map Views* ► menu:

1. By clicking on *New 3D Map View* you can create a new 3D map view. A floating and dockable QGIS panel will appear (see *The 3D Map View dialog*). It has the same extent and view as the 2D main map canvas and provides a set of navigation tools to turn the view into 3D.
2. By clicking on *Manage 3D Map Views* you get in the 3D Map Views Manager. Here you get the ability to open, duplicate, remove and rename 3D map views.
3. If you created one or more 3D map views, you see them listed in *3D Map Views*. You can turn them on and off by clicking on. They will be saved by saving the project, even if they are turned off.

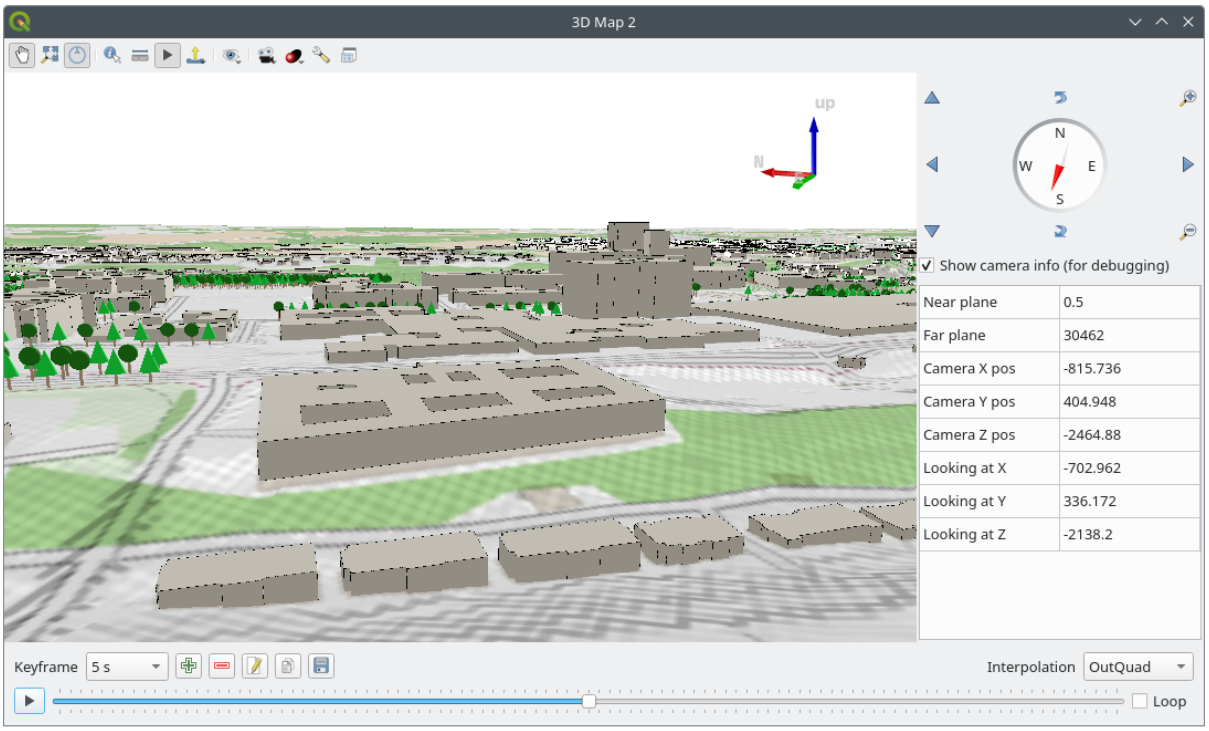


Figura7.21: The 3D Map View dialog

The following tools are provided at the top of the 3D map view panel:

- Camera Control: moves the view, keeping the same angle and direction of the camera
- Zoom Full: resizes the view to the whole layers' extent, or the *reference extent* if set
- Toggle On-Screen Notification: shows/hides the navigation widget (that is meant to ease controlling of the map view)
- Identify: returns information on the clicked point of the terrain or the clicked 3D feature(s) – More details at *Identificando Feições*
- Measurement Line: measures the horizontal, vertical, and total 3D distance between points
  - Click to add points
  - Right-click to finish the measurement
  - Right-clicking twice or pressing *New* will reset the table
- Animations: shows/hides the *animation player* widget
- Export menu allows to export the scene to various formats:
  - Save as Image…: exports the current view to an image file format
  - Export 3D Scene: exports the current view as a 3D scene (`.obj` file), allowing post-processing in applications like Blender… The terrain and vector features are exported as 3D objects. The export settings, overriding the layers *properties* or map view *configuration*, include:
    - *Scene name* and destination *Folder*
    - *Terrain resolution*
    - *Terrain texture resolution*

    - *Model scale*
    - *Suavizar bordas*
    - *Export normals*
    - *Exportar texturas*
- Set View Theme: Allows you to select the set of layers to display in the map view from predefined *map themes*.
- The Camera menu helps you control relation between the 2D and 3D views:
  - Synchronize the views (*2D map view follows 3D camera* and/or *3D camera follows 2D Map view*)
  - *Show visible camera area in 2D map view*
  - *Set 3D scene on 2D map view*: allows to clip the 3D scene and display only the terrain and features intersecting an extent drawn on the 2D map canvas. More options are available in the *General configuration* tab.
  - Cross Section Tool creates a cross section in 3D scene by defining an area on the terrain in the 2D map canvas. Click to set the start point, move the mouse to define the line direction and click again to set the end point. You can adjust the size of the section by moving the mouse after selecting the start and end points. Press `Esc` or right-click to reset the selection.
  - Use Disable Cross Section to remove the cross section.
- Effects adds visual effects to the 3D rendering, such as showing *shadows*, *eye dome lighting* or *ambient occlusion*.
- The Configure… button opens the dialog to configure the 3D map view *settings*.
- Dock 3D Map View: switch from docked widget to top level window

7.2.1 Scene Configuration

The 3D map view opens with some default settings you can customize. To do so, press the Configure… button at the top of the 3D canvas panel to open the *3D configuration* window.

In the 3D Configuration window there are various options to fine-tune the 3D scene:

General

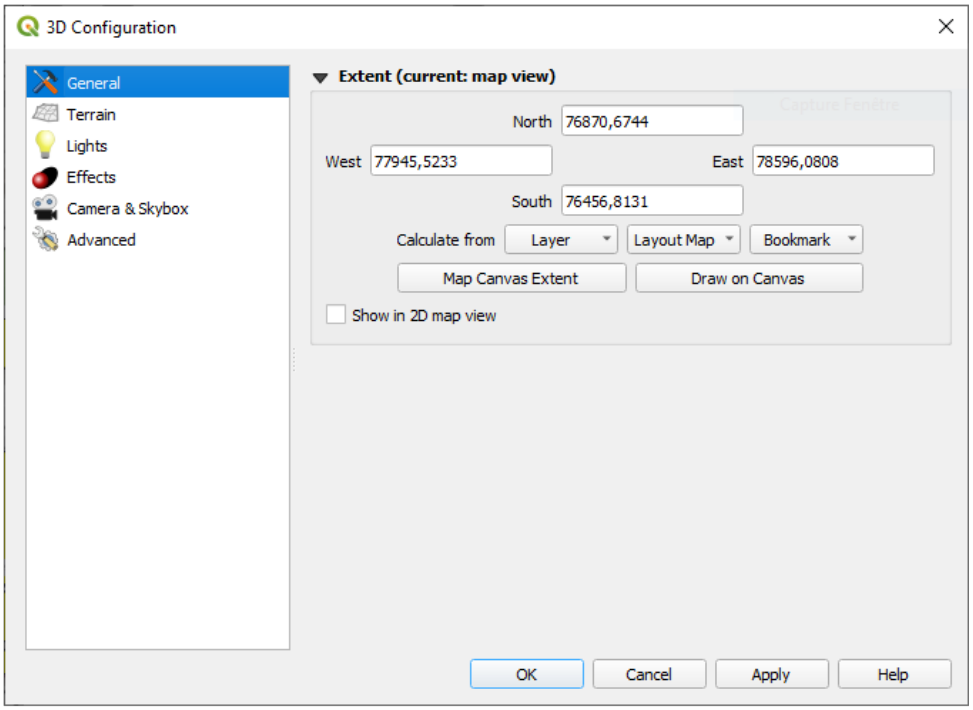


Figura7.22: The 3D Map General Configuration dialog

Under the *General* tab, you can:

- Limit the 3D scene's contents to a 2D map extent, using the *spatial extent selector*: terrain and features are clipped at the specified extent and only the parts falling within the extent are loaded in the scene. Elevation range of the terrain, mesh and pointcloud layers is also taken into account so that the camera is not positioned below the scene's contents, particularly when using the terrain's vertical scale setting to exaggerate the elevation differences.

  The set extent is also used as reference extent when pressing the Zoom full button in 3D map view.

- Check *Show in 2D map view* to display in the main map canvas a rubberband corresponding to the current extent of the 3D scene.

Terreno

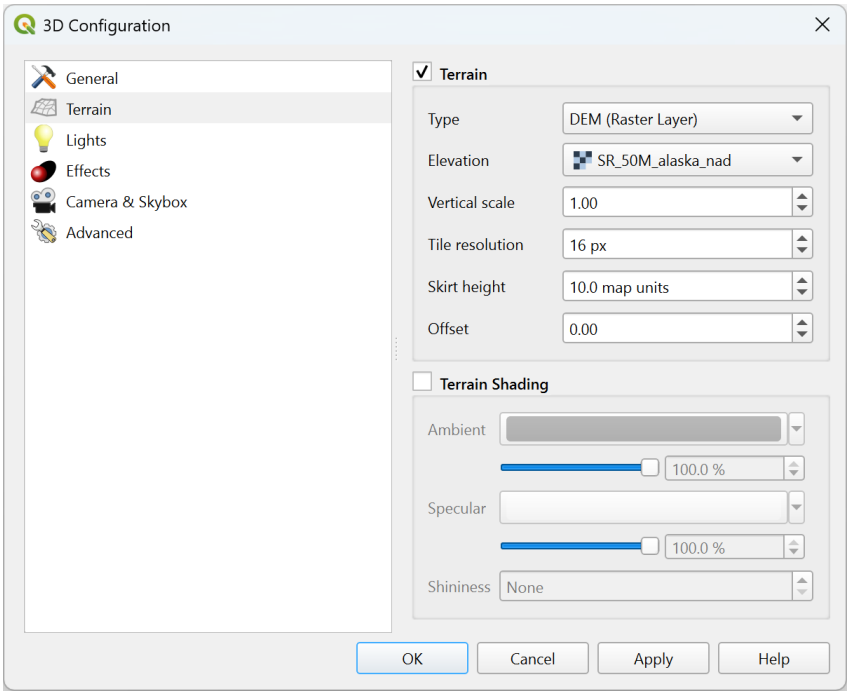


Figura7.23: The 3D Map Terrain Configuration dialog

- *Terrain*: Before diving into the details, it is worth noting that the terrain in a 3D view is represented by a hierarchy of terrain tiles and as the camera moves closer to the terrain, existing tiles that do not have sufficient details are replaced by smaller tiles with more details. Each tile has mesh geometry derived from the elevation raster layer and texture from 2D map layers.
  - The elevation terrain *Type* can be:
    - a *Flat terrain*
    - a loaded *DEM (Raster Layer)*
    - an *Online* service, loading elevation tiles produced by Mapzen tools – more details at https://registry.opendata.aws/terrain-tiles/
    - a loaded *Mesh* dataset
    - a *Quantized Mesh* layer from the project
  - *Elevation*: Raster or mesh layer to be used for generation of the terrain. The raster layer must contain a band that represents elevation. For a mesh layer, the Z values of the vertices are used.
  - *Vertical scale*: Scale factor for vertical axis. Increasing the scale will exaggerate the height of the landforms.
  - *Tile resolution*: How many samples from the terrain raster layer to use for each tile. A value of 16px means that the geometry of each tile will consist of 16x16 elevation samples. Higher numbers create more detailed terrain tiles at the expense of increased rendering complexity.
  - *Skirt height*: Sometimes it is possible to see small cracks between tiles of the terrain. Raising this value will add vertical walls ("skirts") around terrain tiles to hide the cracks.
  - *Offset*: moves the terrain up or down, e.g. to adjust its elevation with respect to the ground level of other objects in the scene.

    Isso pode ser útil quando há uma discrepância entre a altura do terreno e a altura das camadas em sua cena (por exemplo, nuvens de pontos que usam apenas uma altura vertical relativa). Nesse caso, ajustar a

elevação do terreno manualmente para coincidir com a elevação dos objetos em sua cena pode melhorar a experiência de navegação.

- When a mesh layer is used as terrain, you can configure the *Triangles settings* (wireframe display, smooth triangles, level of detail) and the *Rendering colors settings* (as a uniform color or *color ramp based*). More details in the *Mesh layer 3D properties* section.
- *Terrain shading*: Permite a você escolher como o terreno deveria ser renderizado:
  - Shading disabled - terrain color is determined only from map texture
  - Shading enabled - terrain color is determined using Phong's shading model, taking into account map texture, the terrain normal vector, scene light(s) and the terrain material's *Ambient* and *Specular* colors and *Shininess*

Luzes

From the *Lights* tab, press the menu to add

- up to eight *Point lights*: emits light in all directions, like a sphere of light filling an area. Objects closer to the light will be brighter, and objects further away will be darker. A point light has a set position (*X*, *Y* and *Z*), a *Color*, an *Intensity* and an *Attenuation*
- up to four *Directional lights*: mimics the lighting that you would get from a giant flash light very far away from your objects, always centered and that never dies off (e.g. the sun). It emits parallel light rays in a single direction but the light reaches out into infinity. A directional light can be rotated given an *Azimuth*, have an *Altitude*, a *Color* and an *Intensity*.

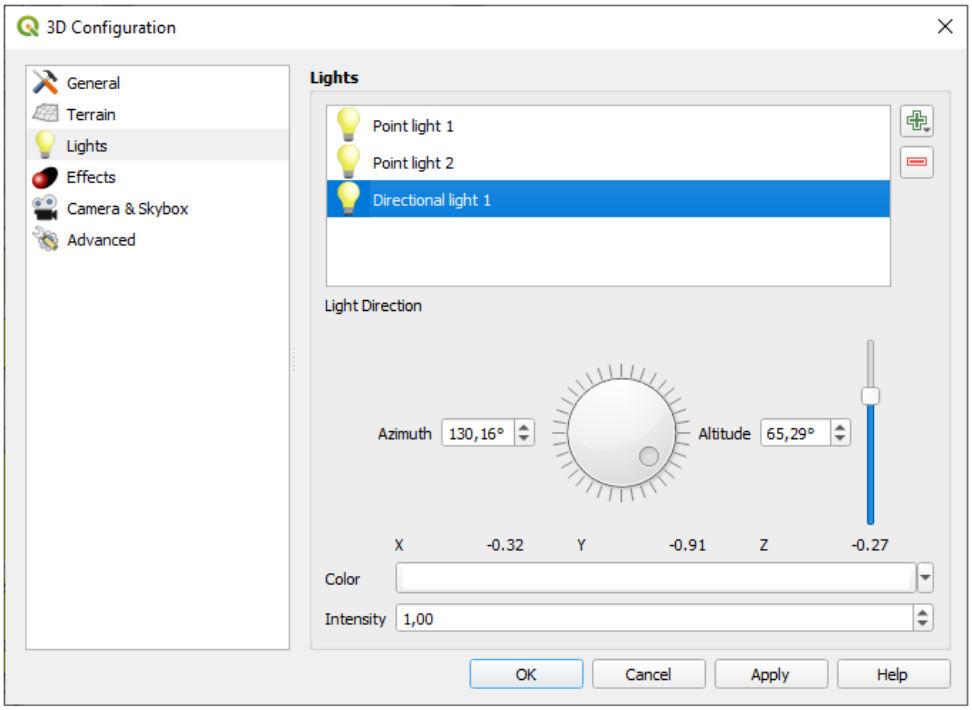


Figura7.24: The 3D Map Lights Configuration dialog

Effects

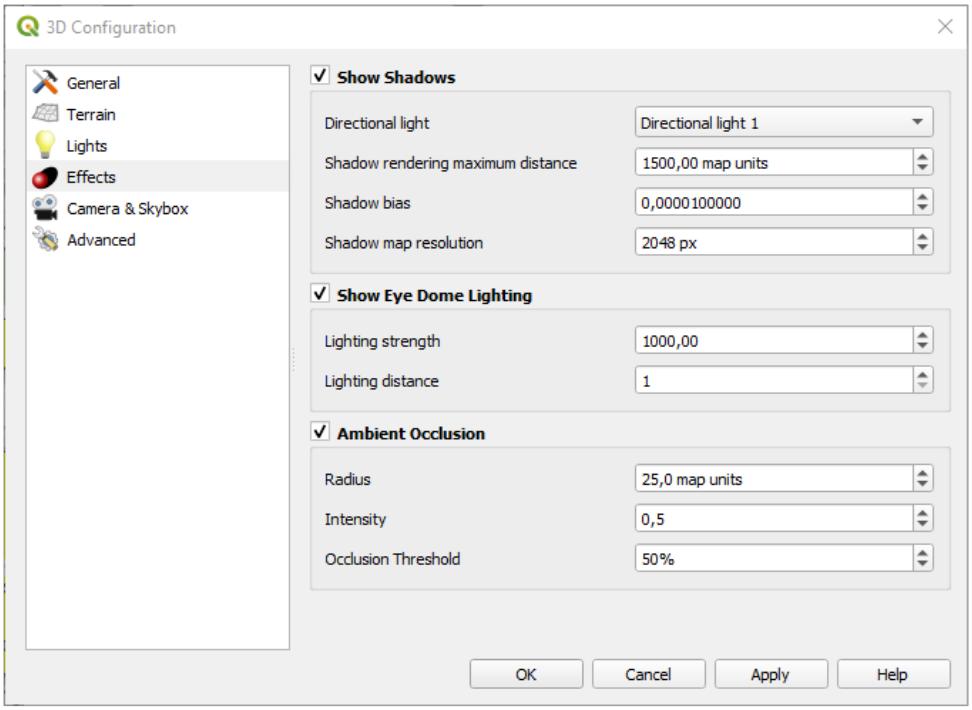


Figura7.25: The 3D Map Effects Configuration dialog

- Check *Show shadows* to display shadows within your scene, given:
  - a *Directional light*
  - a *Shadow rendering maximum distance*: to avoid rendering shadow of too distant objects, particularly when the camera looks up along the horizon
  - a *Shadow bias*: to avoid self-shadowing effects that could make some areas darker than others, due to differences between map sizes. The lower the better
  - a *Shadow map resolution*: to make shadows look sharper. It may result in less performance if the resolution parameter is too high.
- *Show Eye Dome Lighting* (EDL): a post processing effect which enhances depth perception. Each pixel's depth (distance off the camera) is compared to its neighboring pixels' depth and gets highlighted according to that depth difference, making the edges stand out. Affects the whole scene and can be combined with *Screen Space Ambient Occlusion*. Following parameters can be controlled:
  - *Lighting strength*: increases the contrast, allowing for better depth perception
  - *Lighting distance*: represents the distance of the used pixels off the center pixel and has the effect of making edges thicker.
- Add screen-space *Ambient Occlusion* (SSAO): a post processing effect which also enhances depth perception by applying a darker shading to areas which are less exposed to ambient lighting. Affects the whole scene and can be combined with *Eye dome Lighting*. Following parameters can be controlled:
  - *Radius*: how far we will reach to calculate ambient occlusion
  - *Intensity*: how strong the effect should be (higher values make things darker)
  - *Occlusion threshold*: how many neighboring points need to be occluded for the effect to appear (lower values than 50% will make the output darker, but possibly providing greater range of occlusion)

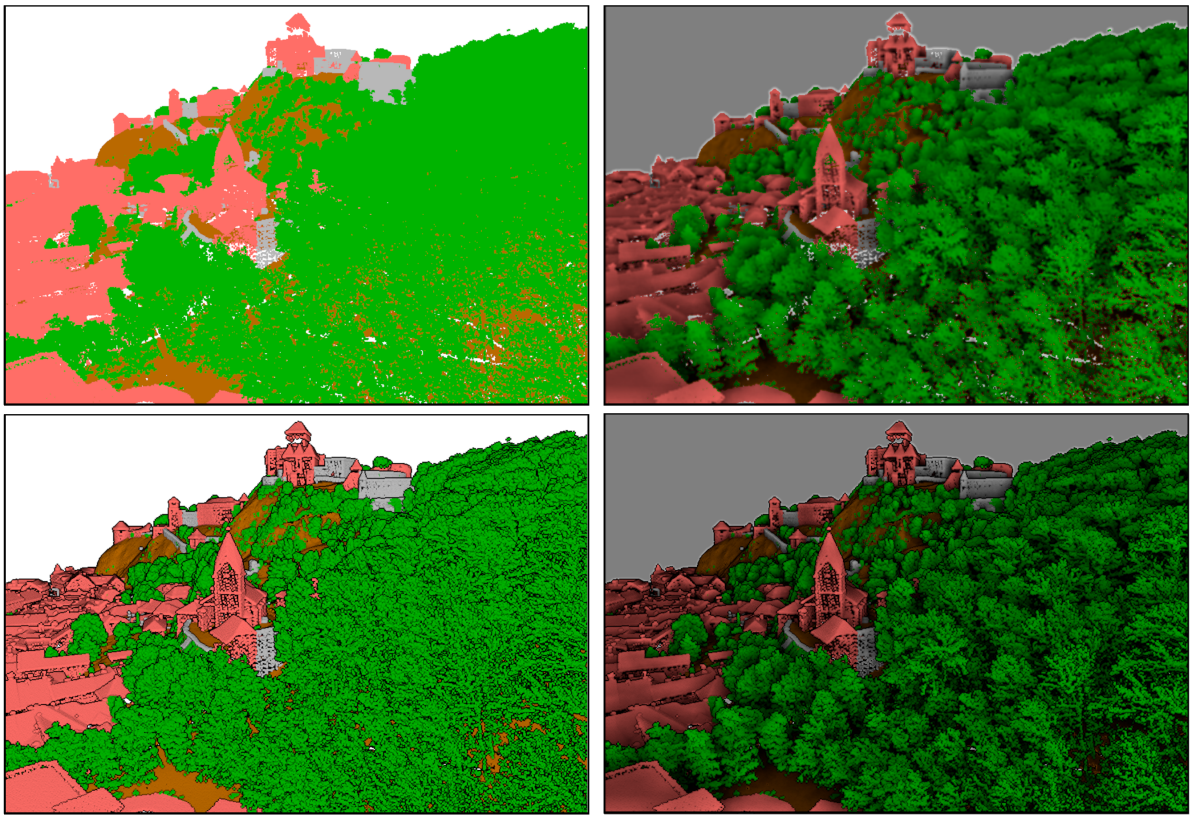


Figura7.26: Rendering Point clouds in 3D map using Eye Dome Lighting (EDL) and/or Screen-Space Ambient Occlusion (SSAO)

From top, left to right: No effect – SSAO only – EDL only – SSAO and EDL

Camera & Skybox

In this tab, you can control different parameters like camera, 3D axis, navigation synchronization and skybox.

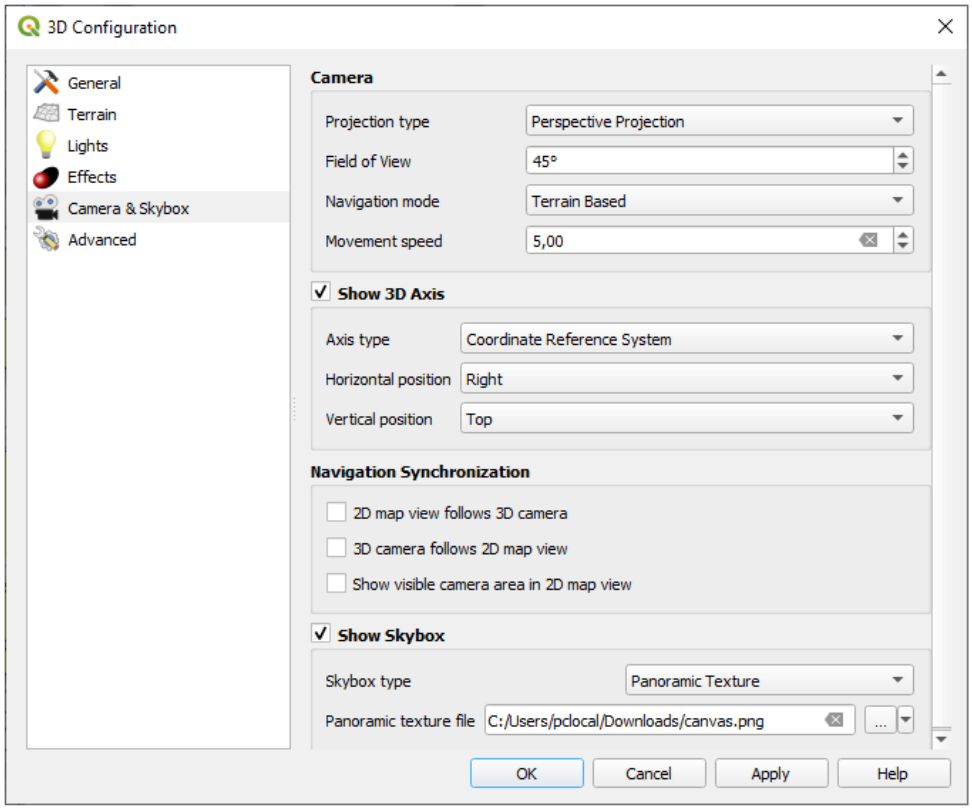


Figura7.27: The 3D Map Camera Configuration dialog

- The *Camera* parameter group overrides some *default camera settings* made in the *Settings* ► *Options* ► *3D* dialog.
- Check *Show 3D Axis* to enable 3D axis tool. This parameter group allows to set the axis type and its position.
  - With the *Coordinate Reference System* type an orthogonal axis will be represented.
  - With the *Cube* type, a 3D cube will be represented. The cube faces can be used to change the camera view: for example, click on the *north* face to set the camera to see from the north.

Dica

Right-click the 3D axis to quickly set its position and type, and the camera view.

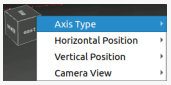


Figura7.28: The 3D Axis context menu

- The *Navigation Synchronization* parameter group adds options to synchronize 2D view with 3D camera position or 3D camera position with 2D view or bi directional synchronization. The last option displays the extent visible from the 3D camera over the 2D map view.
- Check *Show skybox* to enable skybox rendering in the scene. The skybox type can be:
  - *Panoramic texture*, with a single file providing sight on 360°
  - *Distinct faces*, with a texture file for each of the six sides of a box containing the scene

Texture image files of the skybox can be files on the disk, remote URLs or embedded in the project (*more details*).

Avançado

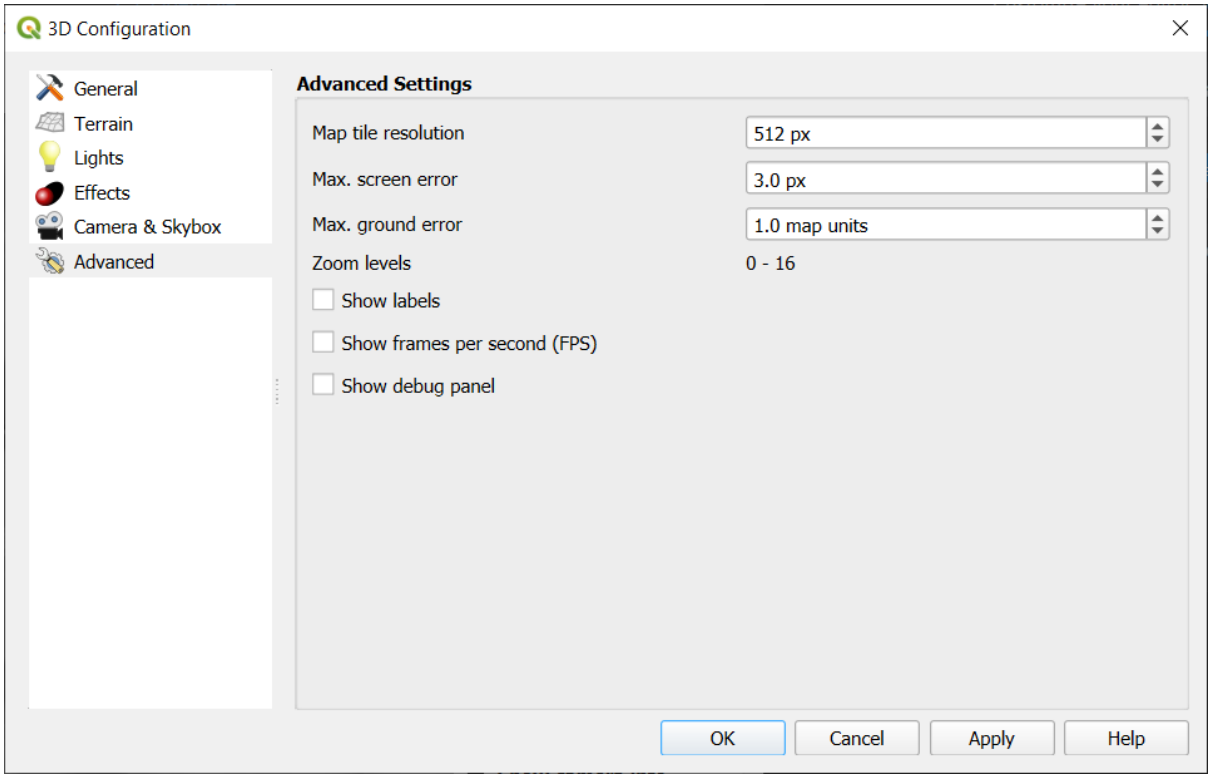


Figura7.29: The 3D Map Advanced Configuration dialog

- *Map tile resolution*: Width and height of the 2D map images used as textures for the terrain tiles. 256px means that each tile will be rendered into an image of 256x256 pixels. Higher numbers create more detailed terrain tiles at the expense of increased rendering complexity.
- *Max. screen error*: Determines the threshold for swapping terrain tiles with more detailed ones (and vice versa) - i.e. how soon the 3D view will use higher quality tiles. Lower numbers mean more details in the scene at the expense of increased rendering complexity.
- *Max. ground error*: The resolution of the terrain tiles at which dividing tiles into more detailed ones will stop (splitting them would not introduce any extra detail anyway). This value limits the depth of the hierarchy of tiles: lower values make the hierarchy deep, increasing rendering complexity.
- *Zoom levels*: Shows the number of zoom levels (depends on the map tile resolution and max. ground error).
- ☐ *Show labels*: Ativa/desativa os rótulos de mapa
- ☐ *Show frames per second (FPS)*
- ☐ *Show debug panel*: allows to show or hide a panel next to the 3D map view displaying options to better control map navigation and debug information:
  - ☐ *Show map tile info*: Include border and tile numbers for the terrain tiles (useful for troubleshooting terrain issues)
  - ☐ *Show bounding boxes*: Show 3D bounding boxes of the terrain tiles (useful for troubleshooting terrain issues)

  - ☐ *Show camera's view center*
  - ☐ *Show camera's rotation center*
  - ☐ *Show light sources*: shows a sphere at light source origins, allowing easier repositioning and placement of light sources relative to the scene contents
  - ☐ *Stop scene updates*
  - ☐ *Show debug overlay*: visual overlay which displays some useful debugging and profiling information. This allows in particular to quickly see the frame graph and the scene graph
  - ☐ *Debug Shadow Map*: renders the scene as a red-black image from the point of view of the light used for shadows (for troubleshooting). The widget is set with a proportional *Size* to the 3D map view's, and docked in a *Corner*.
  - ☐ *Debug Depth Map*: renders the scene's depth map as an image with nearer pixels being darker (for troubleshooting). The widget is set with a proportional *Size* to the 3D map view's, and docked in a *Corner*.
  - *Show camera info*:
    * ☐ *Far plane*: controls how far from the camera rendering stops
    * ☐ *Near plane*: controls how close to the camera rendering starts
    * ☐ *Camera X/Y/Z pos*: sets the camera's position in 3D space
    * ☐ *Looking at X/Y/Z*: sets the target point the camera is looking at

> **Nota**
>
> When your 3D map view is open in a standalone window, you can use `Ctrl+Shift+d` keyboard shortcut to access the debug panel.

7.2.2 Opções de navegação

Para explorar a visualização do mapa em 3D:

- Tilt the terrain (rotating it around a horizontal axis that goes through the center of the window)
  - Press the Tilt up and Tilt down tools
  - Press `Shift` and use the up/down keys
  - Arraste o mouse para frente/para trás com o botão central do mouse pressionado
  - Pressione `Shift` e arraste o mouse para frente/para trás com o botão esquerdo do mouse pressionado
- Rotacionar o terreno (em torno de um eixo vertical que atravessa o centro da janela)
  - Turn the compass of the navigation widget to the watching direction
  - Pressione `Shift` e use as teclas esquerda/direita
  - Arraste o mouse para a direita/esquerda com o botão central do mouse pressionado
  - Pressione `Shift` e arraste o mouse para a direita/esquerda com o botão esquerdo do mouse pressionado
- Mudar a posição da câmera (e o centro de visão), movendo-a em um plano horizontal
  - Drag the mouse with the left mouse button pressed, and the Camera control button enabled
  - Press the directional arrows of the navigation widget

  - Use as teclas para cima/para baixo/esquerda/direita para mover a câmera para frente, para trás, direita e esquerda, respectivamente
- Change the camera altitude: press the `Page Up`/`Page Down` keys
- Alterar a orientação da câmera (a câmera é mantida em sua posição, mas o ponto central da visão se move)
  - Press `Ctrl` and use the arrow keys to turn the camera up, down, left and right
  - Press `Ctrl` and drag the mouse with the left mouse button pressed
- Aumentar e diminuir o zoom
  - Press the corresponding Zoom In and Zoom Out tools of the navigation widget
  - Scroll the mouse wheel (keep `Ctrl` pressed results in finer zooms)
  - Drag the mouse with the right mouse button pressed to zoom in (drag down) and out (drag up)

Para redefinir a visualização da câmera, clique no botão :sup:`Zoom completo' na parte superior do painel de tela 3D.

7.2.3 Criar uma animação

Uma animação é baseada em um conjunto de quadros-chave - posições da câmera em momentos específicos. Para criar uma animação:

1. Toggle on the Animations tool, displaying the animation player widget
2. Click the Add keyframe button and enter a *Keyframe time* in seconds. The *Keyframe* combo box now displays the time set.
3. Usando as ferramentas de navegação, mova a câmera para a posição para associar ao tempo atual do quadro-chave.
4. Repita as etapas anteriores para adicionar quantos quadros-chave (com tempo e posição) forem necessários.
5. Click the button to preview the animation. QGIS will generate scenes using the camera positions/rotations at set times, and interpolating them in between these keyframes. Various *Interpolation* modes for animations are available (eg, linear, inQuad, outQuad, inCirc… – more details at https://doc.qt.io/archives/qt-5.15/qeasingcurve.html#EasingFunction-typedef).

   The animation can also be previewed by moving the time slider. Keeping the *Loop* box checked will repeatedly run the animation while clicking stops a running animation.

Click Export animation frames to generate a series of images representing the scene. Other than the filename *Template* and the *Output directory*, you can set the number of *Frames per second*, the *Output width* and *Output height*.

7.2.4 Camadas vetoriais 3D

Uma camada vetorial com valores de elevação pode ser mostrada no mapa 3D, marcando *Ativar renderizador 3D* na seção *Vista 3D* das propriedades da camada vetorial. Há várias opções disponíveis para controlar a renderização da camada vetorial 3D.

7.3 Elevation Profile View

The *Elevation Profile* panel is a plotting tool for side view, for visualizing elevation data along a line. It supports vector, raster, mesh and point cloud layers. Data can be of 2D or 3D type.

To add an elevation profile view, go to *View* ► *Elevation Profile* menu. You can add as many profile views as you want, and they can be docked, piled on top of each other, or floating.

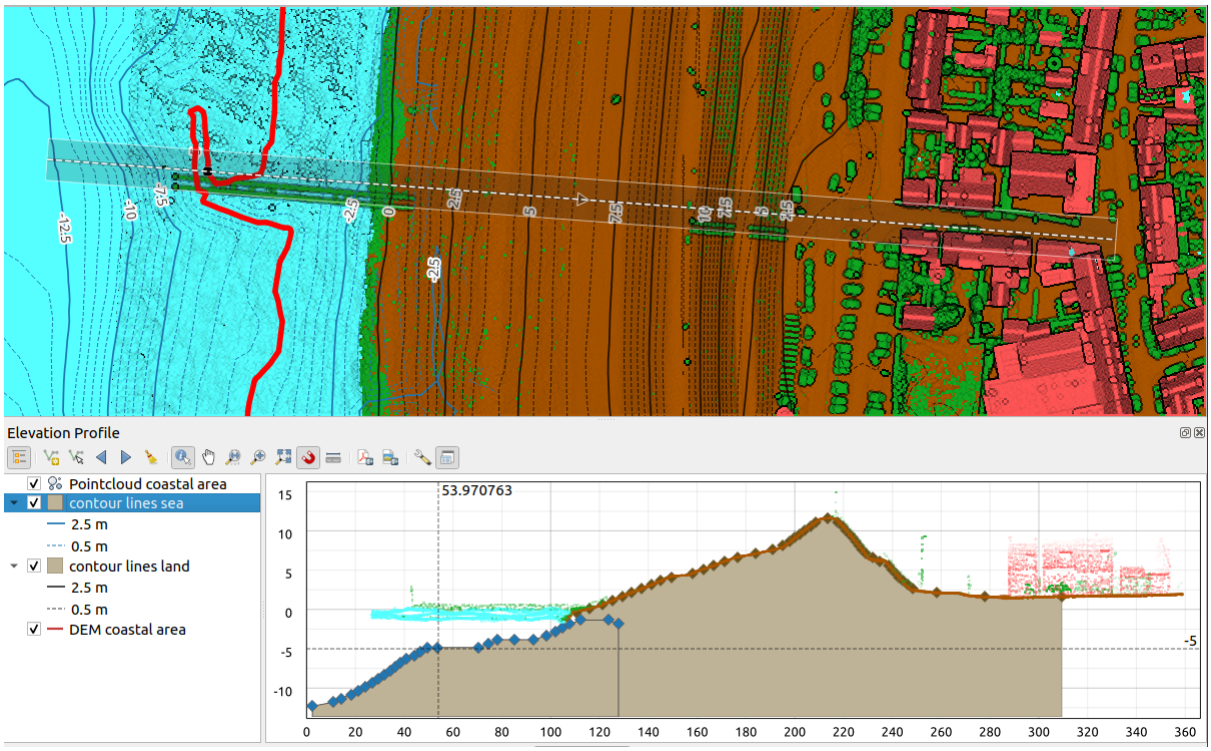


Figura7.30: Elevation Profile dialog embedded below main map canvas
Figure shows a coastline with a town behind a sea-dike. A larger tolerance applied to the elevation profile line returns more point cloud features.

7.3.1 The interface

At the top of the *Elevation Profile* panel, a toolbar provides you with the following tools:

Tabela7.5: Elevation Profile View toolbar

Ferramenta	Atalho	Descrição
Add Layers	hold `Ctrl` while drag-and-drop	Allows you to add new layers to a plot.
Show Layer Tree		Shows or hides a list of project layers to configure rendering in the profile view.
Capture Curve		Draws interactively a line over the map canvas to represent the profile curve.
Capture Curve From Feature		Picks an existing line feature on the map canvas and generate a profile curve along that line.

continua na próxima página

Tabela 7.5 – continuação da página anterior

Ferramenta	Atalho	Descrição
Nudge Left	`Ctrl+Alt+,`	Allows you to slowly move the capture line across the map to the left (e.g. to find the optimal profile line based on the elevation).
Nudge Right	`Ctrl+Alt+.`	Allows you to slowly move the capture line across the map to the right (e.g. to find the optimal profile line based on the elevation).
Clear		Removes the profile line and any plot displayed in the *Elevation Profile* view
Identify Features		Identifies features in the plot canvas via either a single click, or click-and-drag rectangle. Results are shown in the standard *Identify Results* dock.
Pan	`Space`	Click and drag to pan the plot canvas. Can also be done with the middle mouse button.
Zoom X Axis		Zooms in/out along the horizontal axis, keeping the vertical ratio
zoom	`Ctrl+Space`	Click or click-and-drag a rectangle over the plot to zoom in. Press `Alt` and click to instead zoom out.
Zoom Full		Zooms the *Elevation Profile* view to the extent of the capture line
Enable Snapping		Allows to snap to the edge or vertex of the plot features in the profile view. Convenient for accurate retrieval of coordinates or distance measurements.
Measure Distances		Measures horizontal and vertical distances
Export As PDF		Exports plots to PDF (as high quality vector objects)
Export As Image		Exports plots to several image formats
Export Results		Exports plots as 3D features, 2D Profile and Distance/Elevation Table. (The results can be saved as DXF files, CSV files or any of the standard writable vector spatial formats). More details at *Exporting from the elevation profile*.
Options ►		Provides access to configuration settings of the profile elevation line.
► *Lock distance/elevation scales*		Ensures that the horizontal and vertical scales are always kept equal (so that e.g. a 45° slope will appear as a 45° slope in the profile view).
► *Distance units*		Allows to render distances in the profile chart with units other than the map canvas units.
► *Tolerance*		Sets how far from the actual profile line a feature (vector point, line or polygon, point cloud) can reside to be included in the results. Uses the map units and is ignored by other layer or geometry types. For some layer types (e.g. points, lines, polygons), a custom tolerance can be set in the layer's Elevation properties tab, which replaces the global tolerance and allows more control over feature inclusion based on distance to the profile line.
► *Rename Profile…*		Allows to rename the profile view.

continua na próxima página

Tabela 7.5 – continuação da página anterior

Ferramenta	Atalho	Descrição
Dock Elevation Profile View		Switch between docked and floating status of the view

In the bottom left, a copy of the *Layers* panel can be displayed pushing the *Show Layer Tree* button. This is however an independent widget, with its own set of visible layers, in a custom stack order. It allows you to control layers rendering and behavior within the plot canvas:

- Tick the box next to the layer name to set whether it should be rendered in the plot canvas
- Drag-and-drop layers up or down to change the order of the layers
- style rendering of layers in the profile view: double-click a layer or right-click and select *Properties…* to open the layer's *Elevation* properties tab for configuration. A summary of elevation settings is displayed as tooltip when hovering over the layer.

On the right of the layer tree, the plot canvas is the main place you can preview the elevation profile of the enabled layers. It is based on a graduated grid in which the horizontal axis displays the length of the profile line and the vertical axis displays the Z elevation of the observed features. It also allows a set of interactions such as zooming, panning, measuring, identifying features, … using the tools at the top.

7.3.2 Creating an elevation profile

To create a profile view, you can:

1. Go to *View* ► *Elevation Profile* menu. The *Elevation profile* panel opens.
2. Create the profile line along which the terrain and the features will be rendered. Select a drawing tool:
   - Capture Curve: click left over the main map canvas to add vertices and click right to finalize a line that will be used as profile line
   - or Capture Curve From Feature: click a line feature on the map canvas to select it as the profile line. If multiple features are present at the clicked point then a popup menu will appear allowing you to select among them.

   All the line digitizing capabilities such as the *snapping options*, *tracing*, *digitizing techniques* or the advanced *digitizing panel* are available for use.

   The plot canvas may start rendering some features.
3. The next step is to configure the elevation properties of the layers you want to visualize.
   1. Push on the Show Layer Tree button to display the list of layers. By default, not all the layers of the project are loaded and referenced in the profile tool; e.g., raster layers which are not marked as having elevation data in their properties are ignored. Clicking the Add layers button will show a filtered list of possible layers which can be added to the plot, but which currently aren't in the plot. Applying the dialog with selected layers will automatically mark them as having elevation data and immediately add them to the plot.

      Dica

      Holding `Ctrl` key, it is possible to drag and drop additional layers from the main layer tree to one in the elevation profile tool.

   2. Toggle visibility of the layers you are interested in. These are the only ones rendered in the profile view and selected layers can be different from the main *Layers* panel's.

3. Double-click a layer name or right-click and select *Properties*. The *Elevation* properties tab of the layer opens. This is the place you configure how each feature or terrain should render on the profile view. Available properties depend on the layer type:

   - *Raster Elevation Properties*
   - *Vector Elevation Properties*
   - *Point Cloud Elevation Properties*
   - *Mesh Elevation Properties*

   The profile view starts rendering terrain or features of active layers crossing the given profile curve, as soon as they have elevation properties configured.

4. Under Options drop-down menu, you can:

   - set the *Tolerance* value. This value is used to create a flat buffer around the elevation profile line, visible in the main map canvas. Any visible feature (point, line, polygon, etc.) overlapping that buffer will be captured in the plot canvas.
   - select the *Show Subsections Indicator* to display vertical lines at the distance of the vertices of the profile curve. This highlights the position of the vertices of the curve on the Elevation Profile.
   - select the *Subsections Symbology* to set the color, opacity, width and style of the Subsections Indicator lines.

> **Nota**
>
> **Limitations with polygon extrusion**
>
> Geometry extrusion can be set in the *Elevation* properties of a layer, and rendered in the profile view. When tolerance is enabled, it is however not trivial to render extruded polygons, thus, for now, polygon extrusion is ignored.

7.3.3 Interacting with the profile Views

When an elevation profile line is created, the plot canvas zooms to its full extent. On the X-axis you can see the length of the profile and, on the Y-axis the height range between minimum and maximum height captured, both in map units.

When you move the mouse pointer in the elevation profile view, you will see two crossing dot lines:

- the vertical line shows the height information
- the horizontal line shows the distance from the beginning of the elevation profile Line

When you move the mouse pointer in the elevation profile canvas, you can also see a black dot move along the elevation profile line on the main map canvas. At the middle of the profile line, an arrow indicates its direction.

As for the main map canvas, QGIS provides means to navigate on the plot canvas:

- Pan is used to move the elevation profile extent in any direction you want. Holding `Space` key while moving the mouse also shifts the plot canvas extent.
- Zoom X Axis is used to zoom in along the horizontal axis, keeping the ratio of the vertical axis (the elevation) the same. Left click to stretch out the plot along the X axis, with the clicked point at the center of the axis, or drag a rectangle to stretch out the plot along the X axis to the rectangle width. Hold `Alt` while using Zoom X Axis to zoom out along that axis.

- Zoom is used to zoom in on a certain point (using left click), or to a certain extent (dragging a rectangle on the area). Hold `Alt` while using Zoom to instead zoom out. In combination with the `Ctrl` key you can zoom in or out more smoothly.
- Zoom Full is the default zoom level used at the beginning and shows the full extent of the profile line, with all returned features. Use it to reset the zoom level.

It is also possible to interact with the elements displayed in the plot canvas:

- Press Enable snapping button to accurately catch points, vertices or edges of the features, for an accurate measurement or coordinates report.
- Identify Features is used to identify features of the visible layers in the layer tree. You can drag a rectangle across several features in the profile view to query all of them. When compatible with the format (e.g. vector, point cloud), these features will be highlighted in the main map canvas.
- Measurement Distances: click or select two points in the plot canvas to report the horizontal *Distance*, the *Elevation* and the *Total distance* between them, in map units.
- Nudge Left and Nudge right are used to shift the position of the elevation profile line in the map canvas to its left or right. The plot canvas will be redrawn, showing features and terrain overlapping the profile line buffer. The line is moved sideways using the *Tolerance* value in Options menu.

> **Aviso**
>
> Closing an elevation profile view or the project currently removes the view from the project.

For more details, give a look to QGIS elevation profile/cross section tool – a deep dive!, a presentation done by Nyall Dawson.

7.3.4 Exporting from the elevation profile

The top toolbar of the *Elevation profile* dialog provides a variety of tools to export an elevation profile, allowing you to reuse the plots in another software:

- Export As PDF: Exports plots to PDF (as high quality vector objects). Following options are requested for generating the output file:
  - The *Page size*, selecting a page from a predefined list or setting custom dimensions, and the page orientation
  - The *Chart ranges* to control the extent of the chart to export:
    - on the X axis, the *Minimum distance* and *Maximum distance* from the profile curve starting point
    - On the Y axis, the *Minimum elevation* and *Maximum elevation*
  - The *Distance axis settings*: helps you configure the horizontal graduation of the grid over which the elevation profile is rendered. You can customize the *Label interval*, the *Major grid line interval* and the *Minor grid line interval*.
  - The *Elevation axis settings*, same as above, for vertical graduation
- Export As Image: Exports plots to an image file format. Specific *Export options* such as the *Image width* and *Image height* are requested, along with the *Chart ranges*, *Distance axis settings* and *Elevation axis settings* previously mentioned.

- Export results: Exports the profile results as a set of features of a vector layer (`.DXF`, `.CSV`, `.SHP`, `.GPKG`, …). Depending on the output format, multiple layers may be returned when the geometry types of exported features differ.
  - *Export 3D features*: Exports profile lines as 3D features, with elevation values taken from the elevation slices and stored in exported geometry Z values.
  - *Export 2D profile*: Exports profiles as 2D profile lines, with elevation stored in exported geometry Y dimension and distance in X dimension (as shown in the elevation profile widget).
  - *Export distance/elevation table*: Exports profiles as a table of sampled distance vs elevation values.

CAPÍTULO 8

Ferramentas Gerais

8.1 Conteúdo da ajuda

Sempre que precisar de ajuda sobre um tópico específico, você pode acessar a página correspondente no Manual do Usuário atual, através do botão *Ajuda*, disponível na maioria das caixas de diálogo — por favor, observe que plugins de terceiros podem apontar para páginas da web dedicadas.

8.2 Painéis

Por padrão, o QGIS apresenta muitos painéis para o usuário trabalhar. Alguns desses painéis estão descritos abaixo, enquanto outros podem ser encontrados em diferentes partes do documento. Uma lista completa dos painéis aprentados pelo QGIS está disponível em *View* ► *Panels* ► menu and mentioned at *Painéis*.

8.2.1 Painel Camadas

O painel *Layers* (também chamado de ``legenda do mapa`) lista todas as camadas no projeto e ajuda a gerenciar a visibilidade e forma do mapa. Você pode mostrar ou esconder o painel presssionando `Ctrl+1`.

QGIS provides a variety of ways to add layers to a project:

- using the *Add* button from the dedicated data provider tab in the *Data source manager* dialog
- from QGIS *Browser panel* or *DB Manager*: double-click, drag-and-drop files and layers onto QGIS or use the contextual menu
- drag-and-drop files from the Operating System files explorer onto QGIS

In all these scenarios, you can open one or many layers at a time. New layers are added to the *Layers* panel:

1. if dropped over the *Layers* panel, at the exact location they are released
2. in the other cases, at a location that respects the global *behavior used when adding new layers* setting
3. and in case of multiple layers, they are sorted in a way that increases the chance of their stacking being logical and features being visible as most as possible, using the following logic (top to bottom):
    - vector point layers
    - vector line layers

- vector polygon layers
- point cloud layers
- mesh layers
- raster layers

No topo do painel *Layers* , uma barra de ferramentas permite:

- Open the layer styling dock (F7): alterna o painel *Layer Styling* entre ligado e desligado.
- Add new group: veja *Interagir com grupos e camadas*
- Manage Map Themes: controla a visibilidade das camadas e as arranja em diferentes *map themes*.
- filtra camadas na árvore da legenda:
  - *Filter Legend by Map Content*: only the layers that are set visible and whose features intersect the current map canvas have their style rendered in the layers panel. Otherwise, a generic NULL symbol is applied to the layer. Based on the layer symbology, this is a convenient way to identify which kind of features from which layers cover your area of interest.
  - *Show Private Layers*: a convenient shortcut to display and interact with *private layers* in the *Layers* panel without modifying the project settings.
  - *Show Broken Layers Only*: only layers with broken data sources are displayed.
- Filter Legend by Expression: apply an expression to remove styles from the selected layer tree that have no feature satisfying the condition. This can be used to highlight features that are within a given area/feature of another layer. From the drop-down list, you can edit and clear the expression currently applied.
- Expand All or Collapse All layers and groups in the layers panel.
- Remove Layer/Group corrente selecionado.

Figura8.1: Barra de Ferramentas de Camadas no Painel de Camadas

Nota

Tools to manage the layers panel are also available for map and legend items in print layouts

Configurando temas de mapas

The Manage Map Themes drop-down button provides access to convenient shortcuts to manipulate visibility of the layers in the *Layers* panel:

- *Mostrar Todas as Camadas*
- *Ocultar Todas as Camadas*
- *Mostrar Camadas Selecionadas*
- *Ocultar Camadas Selecionadas*

- *Toggle Selected Layers*: changes the visibility of the first selected layer in the panel, and applies that state to the other selected layers. Also accesible through `Space` shortcut.
- *Toggle Selected Layers Independently*: changes the visibility status of each selected layer
- *Ocultar Camadas Não Selecionadas*

Beyond the simple control of layer visibility, the Manage Map Themes menu allows you to configure **Map Themes** in the legend and switch from one map theme to another. A map theme is a **snapshot** of the current map legend that records:

- the layers set as visible in the *Layers* panel
- **e** para cada camada visível:
  - a referência ao *estilo* aplicado à camada
  - the visible classes of the style, ie the layer checked node items in the *Layers panel*. This applies to *symbologies* other than the single symbol rendering
  - the collapsed/expanded state of the layer node(s) and the group(s) it's placed inside

Para criar um tema de mapa:

1. Marcar uma camada que você deseja mostrar
2. Configurar as propriedades da camada (simbologia, diagrama, etiquetas…) como de costume
3. Expand the *Style* ► menu at the bottom and click on *Add…* to store the settings as *a new style embedded in the project*

> **Nota**
>
> Um tema de mapa não lembra os detalhes atuais das propriedades: apenas uma referência ao nome do estilo é salva, portanto, sempre que você aplicar modificações na camada enquanto este estilo estiver ativado (por exemplo, alterar a renderização da simbologia), o tema do mapa será atualizado com nova informação.

4. Repetir as etapas anteriores conforme necessário para as outras camadas
5. If applicable, expand or collapse groups or visible layer nodes in the *Layers* panel
6. Click on the Manage Map Themes button on top of the panel, and *Add Theme…*
7. Digite o nome do tema do mapa e clique em *OK*.

The new theme is listed in the lower part of the drop-down menu.

You can create as many map themes as you need: whenever the current combination in the map legend (visible layers, their active style, the map legend nodes) does not match any existing map theme contents as defined above, click on *Add Theme…* to create a new map theme, or use *Replace Theme* ► to update a map theme. You can rename the active map theme with *Rename Current Theme…* or use the *Remove Current Theme* button to delete it.

Map themes are helpful to switch quickly between different preconfigured combinations: select a map theme in the list to restore its combination. All configured themes are also accessible in the print layout, allowing you to create different map items based on specific themes and independent of the current main canvas rendering (see *Map item layers*).

Visão geral do menu de contexto do painel Camadas

At the bottom of the toolbar, the main component of the Layers panel is the frame listing all the layers added to the project, optionally organized in groups. A layer with a checked box next to it displays its contents overlapping the map canvas extent, unless a *scale-based visibility* is set. A layer can be selected and dragged up or down in the legend to change the Z-ordering. Z-ordering means that layers listed nearer the top of the legend are drawn over layers listed lower down in the legend. Also a layer or a group of layers can be dragged across several QGIS instances.

> **Nota**
>
> The Z-ordering behavior can be overridden by the *Layer Order* panel.

Depending on the item selected in the panel, a right-click shows a dedicated set of options presented below. Some of these entries are not available when multiple selections are made (e.g., *Filter*, *Rename*, *Properties*).

Tabela8.1: Contextual menus from *Layers* panel items

Opção	Grupo	Camada Vetorial	Camada Raster	Mesh Layer	Point Cloud Layer	3D Layer
Zoom to Layer(s)/Group	✓	✓	✓	✓	✓	✓
Zoom Para Seleção		✓				
Mostrar em Visão Geral		✓	✓	✓	✓	✓
Show Feature Count		✓				
Motrar rótulo		✓		✓		
Copiar Camada/Grupo	✓	✓	✓	✓	✓	✓
Renomear Camada/Grupo	✓	✓	✓	✓	✓	✓
Zoom para Resolução Nativa (100%)			✓			
Stretch Using Current Extent			✓			
Create Raster Attribute Table			✓			
Open Raster Attribute Table			✓			
Load Raster Attribute Table from VAT.DBF			✓			
Atualizar Camada SQL…		✓				
Execute SQL…		✓				
Edit Virtual Layer…		✓				
Add Group	✓					
Duplicate Layer		✓	✓	✓	✓	✓
Remove Layer/Group…	✓	✓	✓	✓	✓	✓
Move Out of Group		✓	✓	✓	✓	
Move to Top	✓	✓	✓	✓	✓	
Move to Bottom	✓	✓	✓	✓	✓	
Check and all its Parents		✓	✓	✓	✓	
Group Selected		✓	✓	✓	✓	

continua na próxima página

Tabela 8.1 – continuação da página anterior

Opção	Grupo	Camada Vetorial	Camada Raster	Mesh Layer	Point Cloud Layer	3D Layer
Abrir Tabela de Atributos		☑				
Alternar Edição		☑		☑		
Edições Atuais ►		☑		☑		
Filtrar…		☑	☑		☑	
Change Data Source…		☑	☑	☑	☑	
Repair Data Source…		☑	☑	☑	☑	
Actions on selections ► (in edit mode)		☑				
► *Duplicate Feature*		☑				
► *Duplicate Feature and Digitize*		☑				
Set Layer Scale Visibility…		☑	☑	☑	☑	☑
Zoom para escala visível		☑	☑	☑	☑	
Layer CRS ►		☑	☑	☑	☑	☑
► *Set Project CRS from Layer*		☑	☑	☑	☑	☑
► *Set to..* (recent CRSs)				☑	☑	☑
► *Set Layer CRS…*		☑	☑	☑	☑	☑
Set Group CRS…	☑					
Set Group WMS Data…	☑					
Mutually Exclusive Group	☑					
Check and all its children (Ctrl-click)	☑					
Uncheck and all its children (Ctrl-click)	☑					
Make Permanent		☑				
Exportar ►	☑	☑	☑	☑	☑	☑
► *Save As…*			☑			
► *Save Features As…*		☑				
► *Save Selected Features As…*		☑				
► *Save As Layer Definition File…*	☑	☑	☑	☑	☑	☑
► *Save As QGIS Layer Style File…*		☑	☑	☑	☑	☑
Estilos ►		☑	☑	☑	☑	☑
► *Copy Style*		☑	☑	☑	☑	☑
► *Paste Style*	☑	☑	☑	☑	☑	
► *Add…*		☑	☑	☑	☑	☑
► *Rename Current…*		☑	☑	☑	☑	☑
► *Edit symbol…*		☑				
► *Copy Symbol*		☑				
► *Paste Symbol*		☑				
Add Layer Notes…		☑	☑	☑	☑	☑

continua na próxima página

Tabela 8.1 – continuação da página anterior

Opção	Grupo	Camada Vetorial	Camada Raster	Mesh Layer	Point Cloud Layer	3D Layer
Edit Layer Notes…		☑	☑	☑	☑	
Remove Layer Notes		☑	☑	☑	☑	
Propriedades…		☑	☑	☑	☑	☑

For GRASS vector layers, Toggle editing is not available. See section *Digitalizando e editando uma camada vetorial GRASS* for information on editing GRASS vector layers.

Interagir com grupos e camadas

Layers in the legend window can be organized into groups. There are different ways to do this:

1. Press the icon to add a new group. Type in a name for the group and press `Enter`. Now click on an existing layer and drag it onto the group.
2. Select more than one layer, then press the icon. The selected layers are automatically inserted into the new group.
3. Select some layers, right-click in the legend window and choose *Group Selected*. The selected layers will automatically be placed in a new group.

To move a layer out of a group, drag it out, or right-click on it and choose *Move Out of Group*: the layer is moved from the group and placed above it. Groups can also be nested inside other groups. If a layer is placed in a nested group, *Move Out of Group* will move the layer out of all nested groups.

To move a group or layer to the top of the layer panel, either drag it to the top, or choose *Move to Top*. If you use this option on a layer nested in a group, the layer is moved to the top in its current group. The *Move to Bottom* option follows the same logic to move layers and groups down.

The checkbox for a group will show or hide the checked layers in the group with one click. With `Ctrl` pressed, the checkbox will also turn on or off all the layers in the group and its sub-groups.

`Ctrl`-click on a checked / unchecked layer will uncheck / check the layer and all its parents.

Enabling the **Mutually Exclusive Group** option means you can make a group have only one layer visible at the same time. Whenever a layer within the group is set visible the others will be toggled not visible.

É possível selecionar mais de uma camada ou grupo ao mesmo tempo mantendo pressionada a tecla `Ctrl` enquanto clica em camadas adicionais. Você pode mover todas as camadas selecionadas para um novo grupo ao mesmo tempo.

You may also delete more than one layer or group at once by selecting several items with the `Ctrl` key and then pressing `Ctrl+D`: all selected layers or groups will be removed from the layers list.

Mais informações sobre camadas e grupos usando o ícone do indicador

In some circumstances, formatting or icons next to the layer or group in the *Layers* panel change to give more information about the layer/group. These elements are:

- to indicate that the layer is in edit mode and you can modify the data
- to indicate that the layer being edited has some unsaved changes
- to indicate *a filter* applied to the layer. Hover over the icon to see the filter expression and double-click to update the query

- to identify layers that are *required* in the project, hence non removable
- to identify an *embedded group or layer* and the path to their original project file
- to identify a layer whose data source was not available at the project file opening (see *Manipulação de caminhos de arquivos quebrados*). Click the icon to update the source path or select *Repair Data Source…* entry from the layer contextual menu.
- to remind you that the layer is a *temporary scratch layer* and its content will be discarded when you close this project. To avoid data loss and make the layer permanent, click the icon to store the layer in any of the GDAL vector formats supported by QGIS.
- to identify a layer used in *offline editing mode*.
- to identify a layer that has no/unknown CRS
- for layers with coordinates stored in a coordinate reference system which is inherently low accuracy (requires the *corresponding setting* to be enabled)
- to identify a temporal layer controlled by canvas animation
- to identify a layer that has *notes* associated
- A grayed name, when the map canvas current scale is outside the layer's visibility scale range (as set in its *Rendering* properties). Select the contextual menu *Zoom to Visible Scale* option to zoom the map to the layer's nearest visibility scale bound.

Control layers rendering through grouping

Groups are a means of structuring layers within a tree in the project but they can also impact how their component layers are rendered, namely as a single flattened object during map renders.

The option for such a rendering is available within the *Layer Styling* panel whenever a group is selected. Under the Symbology tab, check *Render Layers as a Group* to enable a set of options to control the appearance of the child layers as a whole, instead of individual layers:

- *Opacity*: Features from child layers which are obscured by other child layers remain obscured, and the opacity applies to the "whole of group" only.

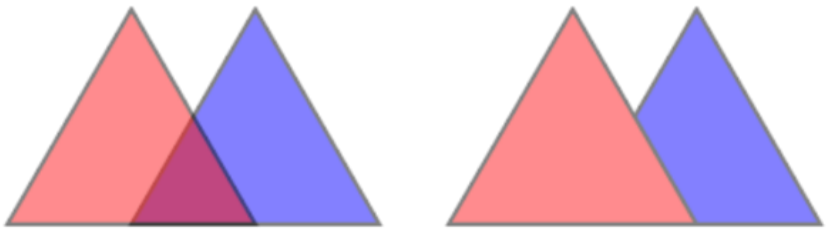


Figura8.2: Setting opacity on layers vs on a group
The image on the left shows two layers being rendered at 50% opacity (underlying features are visible, but semi-masked by the 50% red feature on top). The second image shows the result of setting the opacity on the group (parts of the blue underlying child layer is completely obscured by the red layer on top and then the result is rendered at 50% opacity).

- *Blend modes*: Just like opacity, setting a *blend mode* (like multiply, overlay, …) for an entire group results first in flattening features of child layers, with upper ones obscuring lower. The rendering is then obtained by blending the flat group and the layers sitting below the group.

  - When the child layers have blend modes assigned, it is applied before flattening but the scope is restricted to only affecting other child layers from that group, and not other layers sitting below the whole group.
  - Some more *blending modes* options are available for child layers in groups, in their *Symbology* tab which perform "clipping" style operations on other child layers during the render. You can e.g. clip the render of one layer's content by the content in a second "mask" layer.
- *Layer effects*: applies *effects* only to the flattened render of the child layers; So e.g. a drop shadow effect applied to the group would not be visible for obscured child layers.

When a group is set to *Render layers as a group*, then only the group will be shown in the *Layer Order* panel list. Group children will not be visible in this order list, as their ordering is determined by the placement of the group layer.

Editing layer style

From the *Layers* panel, you have shortcuts to change the layer rendering quickly and easily.

Right-click on a layer and select *Styles* ► in the list in order to:

- see the *styles* currently available for the layer. If you defined many styles for the layer, you can switch from one to another and your layer rendering will automatically be updated on the map canvas.
- copy part or all of the current style, and when applicable, paste a copied style from another layer
- *Rename current…* style
- *Add* a new style (which is actually a copy of the current one)
- or *Remove current* style (only when multiple styles are available).

> **Dica**
>
> **Compartilhe rapidamente um estilo de camada**
>
> From the context menu, copy the style of a layer and paste it to a group or a selection of layers: the style is applied to all the layers that are of the same type (vector, raster, mesh, point cloud, …) as the original layer and, for vector layers, have the same geometry type (point, line or polygon).

When using a symbology based on features classification (e.g. *categorized*, *graduated* or *rule-based* for vector layers, or *classification* for point clouds), right-clicking a class entry in the *Layers* panels makes it possible to edit the visibility of the classes (and their features) and avoid (un)checking them one by one:

- *Toggle Items*
- *Show All Items*
- *Hide All Items*

With vector layer, the contextual menu of a class leaf entry also gives access to:

- *Select features*: selects in the layer all the features matching that class
- *Show in attribute table*: opens an attribute table filtered to only the features matching that class
- update the *symbol color* using a **Color Wheel**. For convenience, the recently used colors are also available at the bottom of the color wheel.
- *Edit Symbol…*: opens the *Symbol Selector* dialog to change feature symbol (symbol, size, color…).
- *Copy Symbol*
- *Paste Symbol*

> **Dica**
>
> Clicar duas vezes em uma entrada de folha de classe também abre a caixa de diálogo *Seletor de Símbolos*.

8.2.2 Painel de Estilo de Camada

O painel *Estilo de Camada* (também ativado com `Ctrl+3`) é um atalho para algumas das funcionalidades da caixa de diálogo *Pripriedades da Camada*. Ele fornece uma maneira rápida e fácil de definir a renderização e o comportamento de uma camada e visualizar seus efeitos sem precisar abrir a caixa de diálogo de propriedades da camada.

Além de evitar a caixa de diálogo de propriedades de camada de bloqueio (ou "modal"), o painel de estilo de camada também evita sobrecarregar a tela com caixas de diálogo e contém a maioria das funções de estilo (seletor de cores, propriedades de efeitos, edição de regras, substituição de rótulos…): Por exemplo, clicar nos botões de cor dentro do painel de estilo de camada faz com que a caixa de diálogo do seletor de cores seja aberta dentro do próprio painel de estilo de camada, e não como uma caixa de diálogo separada.

Em uma lista suspensa de camadas atuais no painel de camadas, selecione um item e:

- Depending on the active item, set:
  - *Symbology* for groups (see *Control layers rendering through grouping*)
  - *Symbology*, *Transparency*, and *Histogram* properties for raster layer. These options are the same as in the *Caixa de Diálogo de Propriedades do Raster*.
  - *Symbology*, *Labels*, *Mask* and *3D View* properties for vector layer. These options are the same as in the *Janela de Propriedades de Vetor* and can be extended by custom properties introduced by third-party plugins.
  - *Symbology*, *Labels* and *3D View* properties for mesh layer. These options are the same as in the *Mesh Dataset Properties*.
  - *Symbology*, *3D View* and *Elevation* properties for point cloud or 3D Tiles layer. These options are the same as in the *Point Clouds Properties* or *3D Tiles Properties*.
  - *Symbology* and *Labels* properties for vector tiles layer. These options are the same as in the *Vector Tiles Dataset Properties*.
  - *Symbology* properties for annotations layer. These options are the same as in the *feature annotation properties*.
- Enable and configure *global map shading* properties
- Manage the associated style(s) in the *Style Manager* (more details at *Gerenciar Estilos Personalizados*).
- See the *History* of changes you applied to the layer style in the current project: you can therefore cancel or restore to any state by selecting it in the list and clicking *Apply*.

Another powerful feature of this panel is the *Live update* checkbox. Tick it to render your changes immediately on the map canvas: you no longer need to click the *Apply* button.

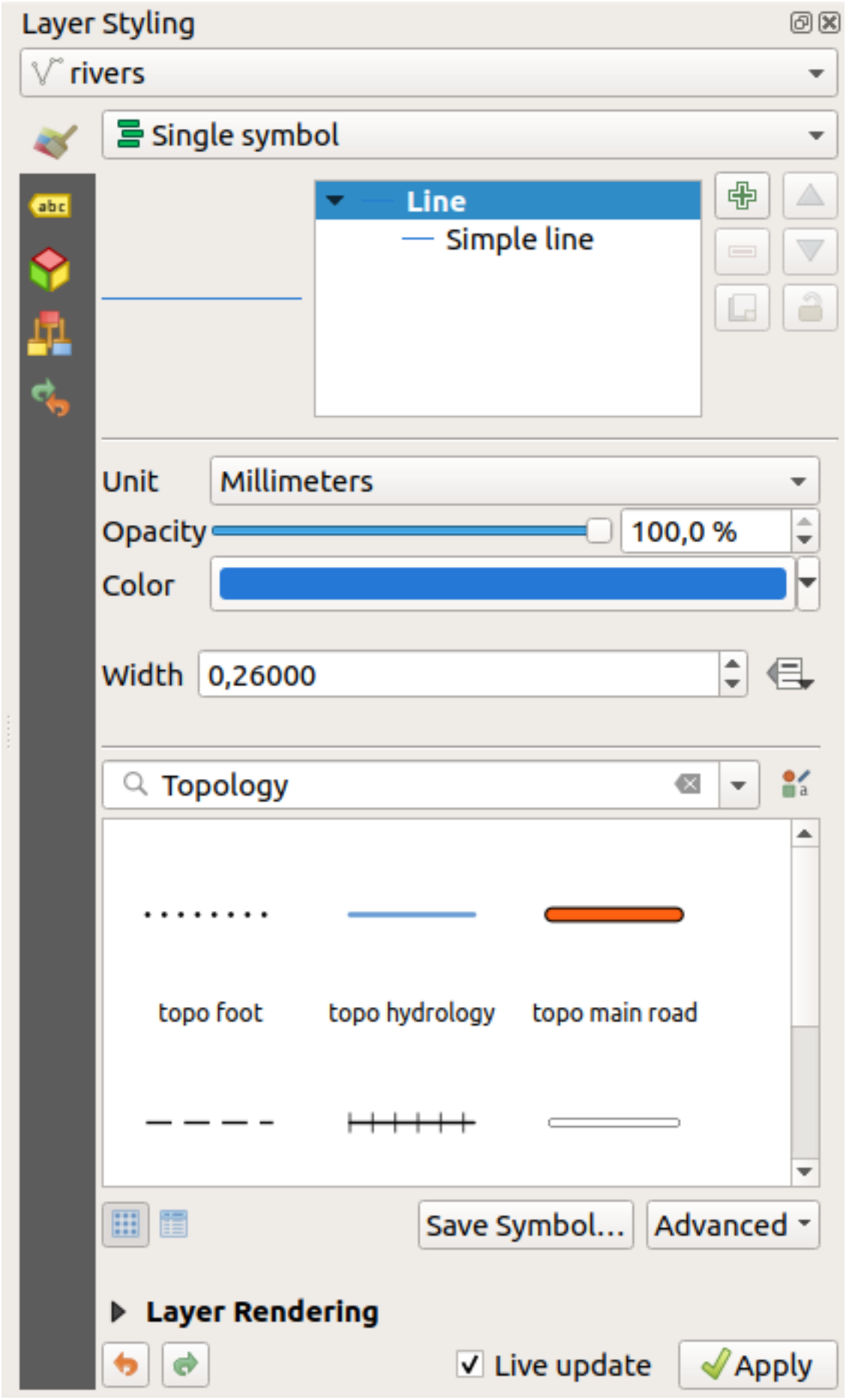


Figura8.3: Defining a layer's symbology from the layer styling panel

8.2.3 Layer Order Panel

By default, layers shown on the QGIS map canvas are drawn following their order in the *Layers* panel: the higher a layer is in the panel, the higher (hence, more visible) it'll be in the map view.

You can define a drawing order for the layers independent of the order in the layers panel with the *Layer Order* panel enabled in *View* ► *Panels* ► menu or with `Ctrl+9`. Check *Control rendering order* underneath the list of layers and reorganize the layers in the panel as you want. This order becomes the one applied to the map canvas. For example, in Fig.8.4, you can see that the `airports` features are displayed over the `alaska` polygon despite those layers' respective placement in the Layers panel.

Unchecking *Control rendering order* will revert to default behavior.

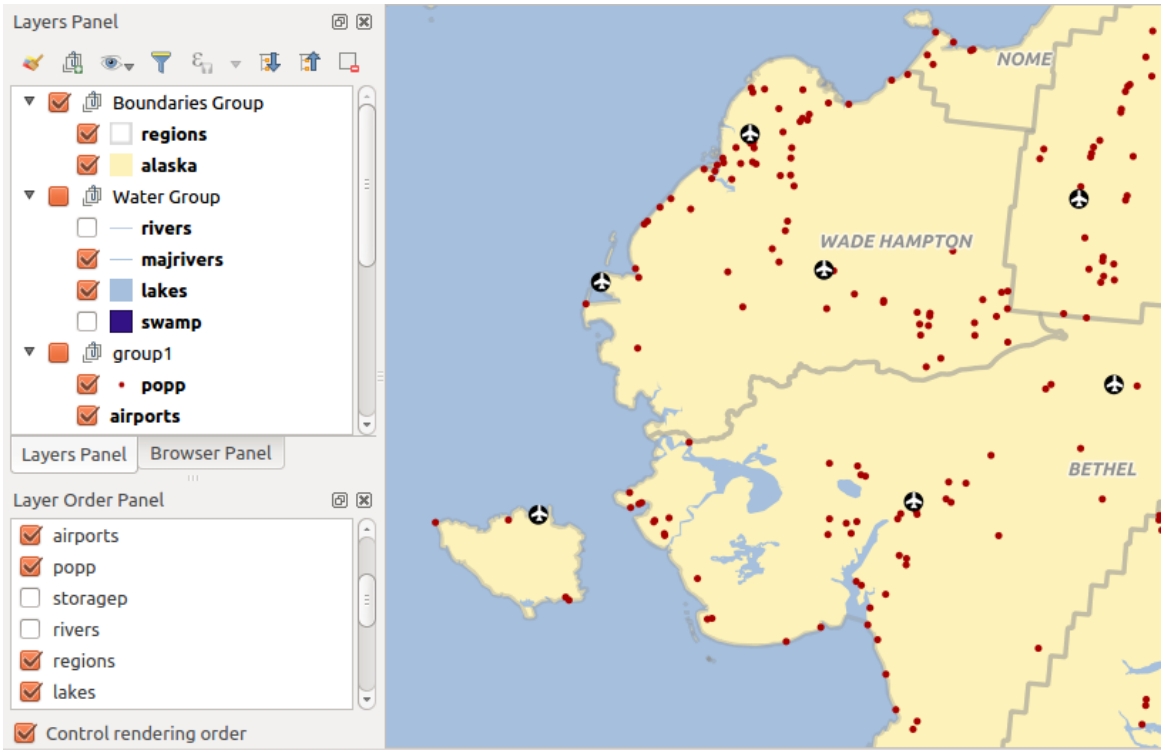


Figura8.4: Define a ordem das camadas independente da legenda

8.2.4 Painel geral

The *Overview* panel (`Ctrl+8`) displays a map with a full extent view of some of the layers. The Overview map is filled with layers using the *Show in Overview* option from the *Layer* menu or in the layer contextual menu. Within the view, a red rectangle shows the current map canvas extent, helping you quickly to determine which area of the whole map you are currently viewing. If you click-and-drag the red rectangle in the overview frame, the main map view extent will update accordingly.

Observe que as rótulos não são renderizados para a visão geral do mapa, mesmo que as camadas usadas na visão geral do mapa tenham sido criadas para rotulagem.

8.2.5 Log Messages Panel

When loading or processing some operations, you can track and follow messages that appear in different tabs using the *Log Messages Panel*. It can be activated using the most right icon in the bottom status bar.

8.2.6 Undo/Redo Panel

For each layer being edited, the *Undo/Redo* (`Ctrl+5`) panel shows the list of actions carried out, allowing you quickly to undo a set of actions by selecting the action listed above. More details at *Undo and Redo edits*.

8.2.7 Statistical Summary Panel

The *Statistics* panel (`Ctrl+6`) provides summarized information on any vector layer. This panel allows you to select:

- the vector layer to compute the statistics on: it can be selected from the top drop-down menu, or synced with the active layer in the *Layers* panel using the *Follow selected layer* checkbox at the bottom of the statistics drop-down list
- the field or *expression* to use: for each layer, the last entry is remembered and automatically computed upon layer reselection.
- the statistics to return using the drop-down button at the bottom-right of the dialog. Depending on the field's (or expression's values) type, available statistics are:

Tabela8.2: Statistics available for each field type

Estatísticas	String	Número inteiro	Float	Data
Contagem	☑	☑	☑	☑
Contagem de Valor Distinto	☑			☑
Contagem de Valor Ausente	☑	☑	☑	☑
Sum		☑	☑	
Média		☑	☑	☑
Desvio Padrão		☑	☑	
Desvio Padrão na Amostra		☑	☑	
Valor mínimo	☑	☑	☑	☑
Valor máximo	☑	☑	☑	☑
Intervalo		☑	☑	☑
Minoria	☑	☑	☑	
Maioria	☑	☑	☑	
Variedade		☑	☑	
Primeiro Quartil		☑	☑	
Terceiro Quartil		☑	☑	
Inter Quartile Range		☑	☑	
Comprimento Mínimo	☑			
Comprimento Máximo	☑			
Comprimento Médio	☑			

O resumo estatístico pode ser:

- returned for the whole layer or *Selected features only*
- copied to the clipboard and pasted as a table in another application
- recalculated using the button when the underlying data source changes (eg, new or removed features/fields, attribute modification)

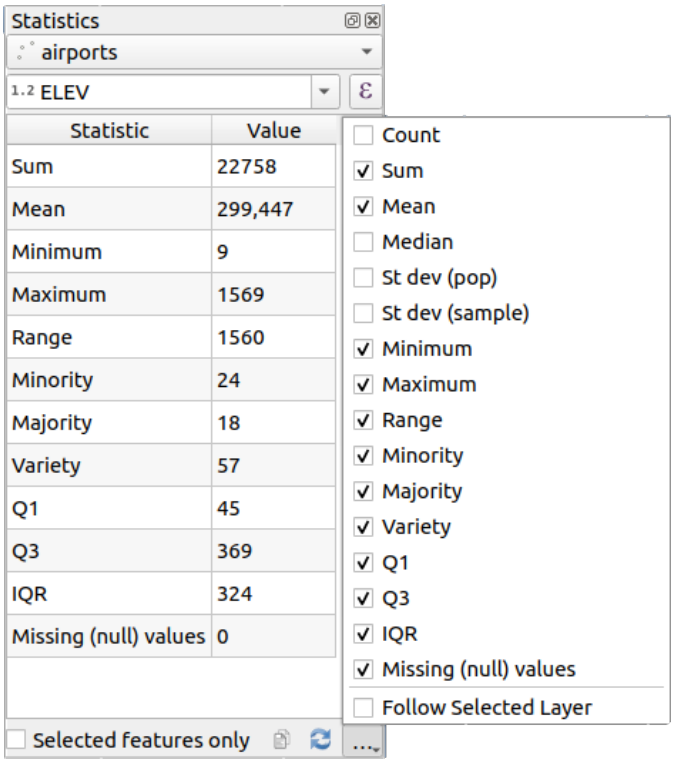


Figura8.5: Mostrar estatísticas em um campo

8.2.8 Debugging/Development Tools Panel

The *Debugging/Development Tools* panel (`F12`) provides a unified place for handling and debugging actions within QGIS. Available tools are organized under following tabs:

- *API Documentation*: embeds developer oriented documentation
- *Network Logger*: records and displays network requests and interactions
- *Query Logger*: logs SQL commands issued by data providers and duration of execution
- *Profiler*: provides load times for every actions in order to detect causes of slow down

Nota

Plugin authors can extend the panel with custom tabs for debugging and developping their own plugins. This is done using `registerDevToolWidgetFactory` method.

API Documentation

The *API Documentation* tab can display either the PyQGIS API documentation or the Qt API documentation. These are embedded in the widget, allowing you to browse and search the docs without leaving QGIS. It is also the place you can display specific help page for objects or classes in your code sample; from the Python interactive console or text editor dialog, select or hover a text and press `F1` or right-click and select Contextual help to get its API help. If you wish to read the active page in the system web browser, then press Open in web browser.

Read *contextual help* for details on the widget configuration.

Network Logger

The *Network Logger* tab helps you record and display network requests, along with a whole load of useful details like request and reply status, header, errors, SSL configuration errors, timeouts, cache status, etc.

From its top toolbar, you can:

- *Record Log*: which will start or stop the logging.
- *Clear Log*: will clear the log history.
- *Save Log…*: will first show a big warning that the log is sensitive and should be treated as confidential and then allow you to save the log.
- Press the *Settings* drop-down menu to select whether to *Show Successful Requests*, *Show Timeouts* and *Show Replies Served from Cache*.
- *Disable cache*: will disable the cache so that every request has to be performed.
- *Filter requests* based on URL string subsets or request status

Ao clicar com o botão direito do mouse em uma solicitação, você pode:

- *Open URL* which will open the URL in your default browser.
- *Copiar URL*
- *Copy As cURL* to use it in the terminal.
- *Copy as JSON*: copies the tree values as a json string to the clipboard, for easy pasting in bug reports or for remote assistance.

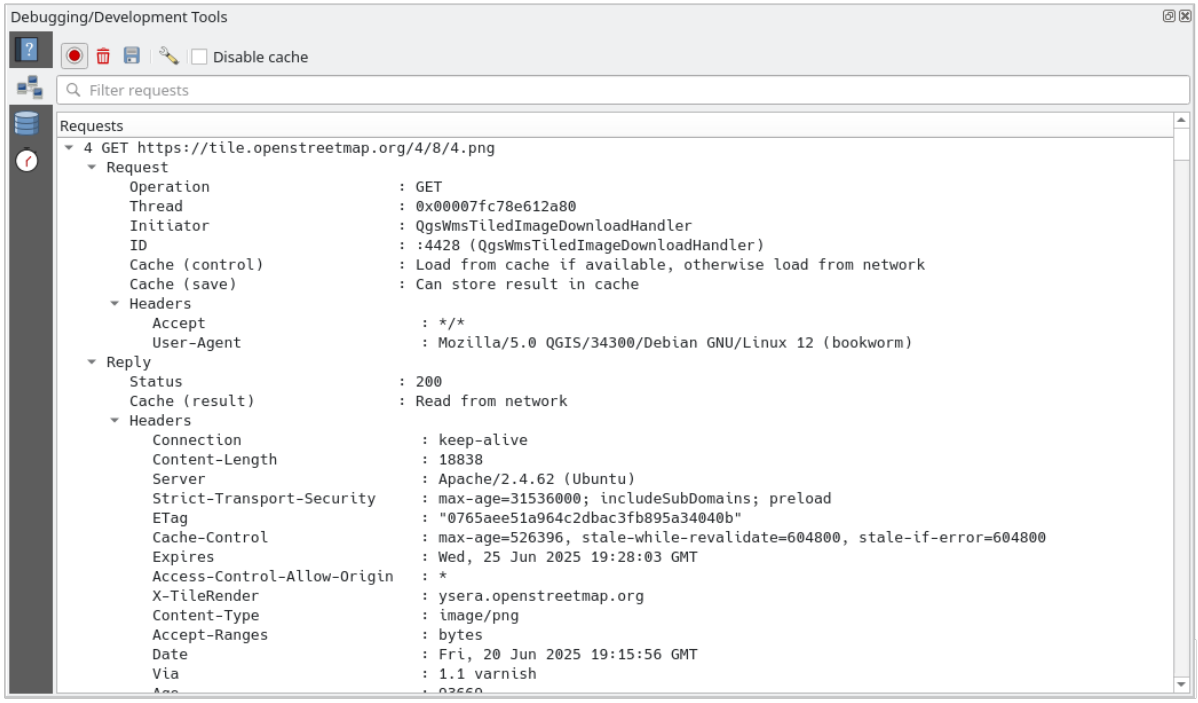


Figura8.6: Network Logger output for GET Request

Query Logger

The *Query Logger* is the place to log the SQL commands sent by the data provider and the connections API to the backend database, together with their execution time as measured by QGIS (i.e. in the client that sent the commands). This can be useful when investigating performances of a particular layer during debugging or development of a QGIS algorithm or plugin.

From its top toolbar, you can:

- *Record Log*: which will start or stop the logging.
- *Clear Log*: will clear the log history.
- *Save Log…*: will first show a big warning that the log is sensitive and should be treated as confidential and then allow you to save the log.
- *Filter queries* based on the query string subsets or details such as provider type, start time, initiator, …

Right-click on a reported query, and you can:

- *Copy SQL* command called by QGIS on the database
- *Copy as JSON*: copies the tree values as a json string to the clipboard, for easy pasting in bug reports or for remote assistance.

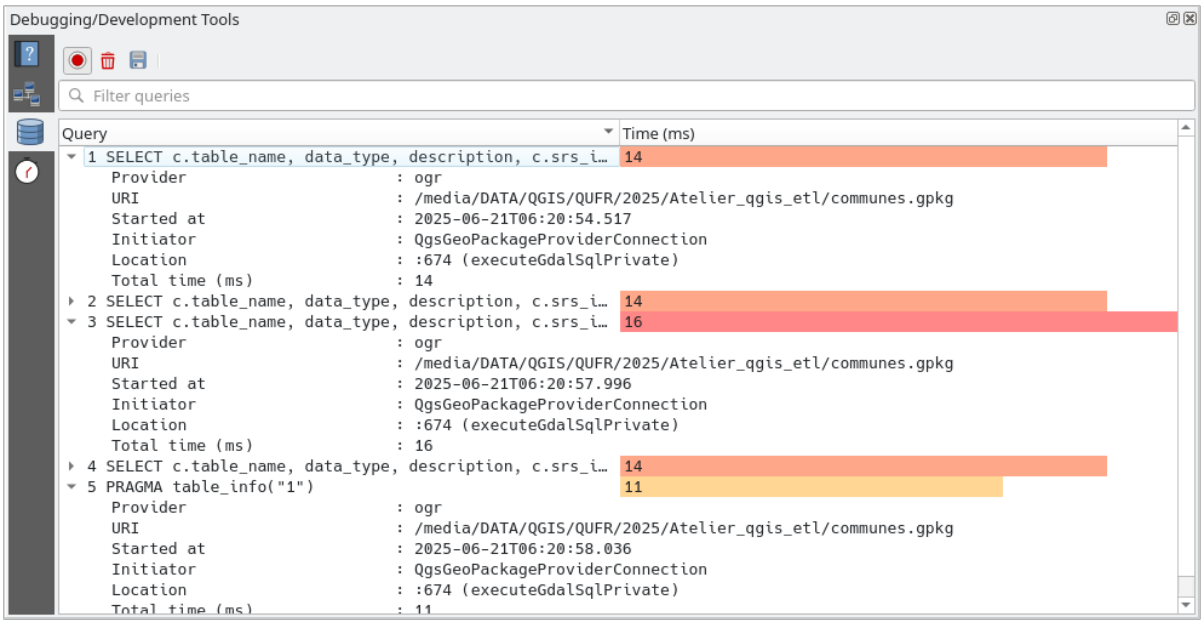


Figura8.7: Query Logger output

Profiler

The *Profiler* tab allows to get load times for every single operation involved in actions requested by the user. Depending on the context, these operations can be settings reading, menu, map canvas or 3D views creation, resolving map layers reference, bookmark or layout loading, … This helps identify causes of slow load times.

Default supported actions can be selected from the *Category* drop-down menu:

- *Startup* helps you diagnose (and fix) occasional long startup times of QGIS.
- *Project Load* allows you to get a breakdown of the various stages of project load, in order to identify the causes of slow project load times.
- *Map Render* tool allows you to identify pain points in your map rendering and track down exactly which layers are causing long map redraws.

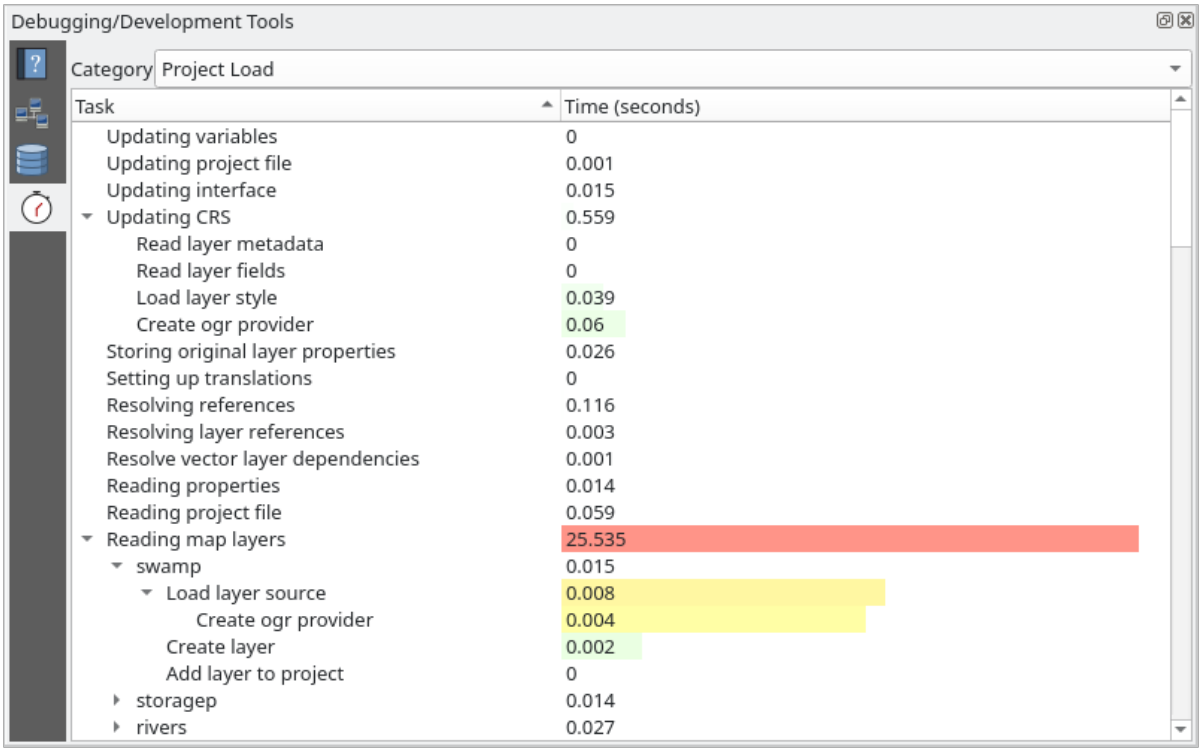


Figura8.8: Profiler for QGIS Startup

8.3 Incorporando camadas de projetos externos

Sometimes, you'd like to keep some layers in different projects, but with the same style. You can either create a *default style* for these layers or embed them from another project to save time and effort.

A incorporação de camadas e grupos de um projeto existente tem algumas vantagens sobre o estilo:

- Todos os tipos de camadas (vetor ou raster, local ou online…)
- Fetching groups and layers, you can keep the same tree structure of the "background" layers in your different projects
- While the embedded layers are editable, you can't change their properties such as symbology, labels, forms, default values and actions, ensuring consistency across projects
- Modifique os itens no projeto original e as alterações são propagadas para todos os outros projetos

If you want to embed content from other project files into your project, select *Layer* ► *Embed Layers and Groups*:

1. Click the … button to look for a project: you can see the content of the project (see Fig.8.9)
2. Hold down `Ctrl` (or `Cmd`) and click on the layers and groups you wish to retrieve
3. Clique *OK*.

The selected layers and groups are embedded in the *Layers* panel and displayed on the map canvas. An icon is added next to their name for recognition and hovering over displays a tooltip with the original project file path.

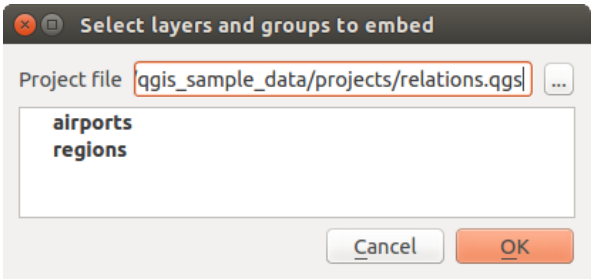


Figura8.9: Selecione camadas e grupos para incorporar

Like any other layer, an embedded layer can be removed from the project by right-clicking on the layer and clicking Remove.

Dica

Alterar a renderização de uma camada incorporada

It's not possible to change the rendering of an embedded layer, unless you make the changes in the original project file. However, right-clicking on a layer and selecting *Duplicate* creates a layer which is fully-featured and not dependent on the original project. You can then safely remove the linked layer.

8.4 Interagindo com feições

8.4.1 Selecionando características

QGIS provides several tools to select features on the map canvas. Selection tools are available in the *Edit* ► *Select* menu or in the *Selection Toolbar*.

Nota

As ferramentas de seleção trabalham com a camada atualmente ativa.

Seleção manual na tela do mapa

Para selecionar uma ou mais características com o mouse, você pode usar uma das seguintes ferramentas:

- Selecione características por área ou clique único
- Selecionar características por Polígono
- Selecionar características à mão livre
- Selecione características por raio

Nota

Other than Select Features by Polygon, these manual selection tools allow you to select feature(s) on the map canvas with a single click.

Nota

Use the Select Features by Polygon tool to use an existing polygon feature (from any layer) to select overlapping features in the active layer. Right-click in the polygon and choose it from the context menu that shows a list of all the polygons that contain the clicked point. All the overlapping features from the active layer are selected.

Dica

Use the *Edit* ► *Select* ► *Reselect Features* tool to redo your latest selection. Very useful when you have painstakingly made a selection, and then click somewhere else accidentally and clear your selection.

While using the *Select Feature(s)* tool, holding `Shift` or `Ctrl` (`Cmd` on macOS) toggles whether a feature is selected (i.e., either adds to the current selection or removes from it).

For the other tools, different behaviors can be performed by holding down:

- `Shift`: adicionar características à seleção atual
- `Ctrl`/`Cmd`: subtract features from the current selection
- `Ctrl+Shift`/`Cmd+Shift`: intersect with the current selection, i.e., only keep overlapping features from the current selection
- `Alt`: select features that are totally within the selection shape. Combined with `Shift` or `Ctrl`/`Cmd` keys, you can add or subtract features to/from the current selection.

Seleção automática

The other selection tools, most of them available from the *Attribute table*, perform a selection based on a feature's attribute or its selection state (note that attribute table and map canvas show the same information, so if you select one feature in the attribute table, it will be selected on the map canvas too):

- Select Features By Expression… or press `Ctrl+F3` to select features using the expression dialog
- Selecione características por valor… ou pressione `F3`
- Deselect Features from All Layers or press `Ctrl+Alt+A` to deselect all selected features in all layers
- Deselect Features from the Current Active Layer or press `Ctrl+Shift+A`
- Select All Features or press `Ctrl+A` to select all features in the current layer
- Inverter seleção de características para inverter a seleção na camada atual
- Select by Location to select the features based on their spatial relationship with other features (in the same or another layer - see *Selecionar pela localização*)
- Select within distance to select features wherever they are within the specified maximum distance from referenced features - see *Selecionar dentro da distância*)

For example, if you want to find regions that are boroughs from `regions.shp` of the QGIS sample data, you can:

1. Use the Select features using an Expression icon
2. Expandir o grupo *Campos e Valores*
3. Clique duas vezes no campo que você deseja consultar ("TIPO_2")

4. Click *All Unique* in the panel that shows up on the right
5. From the list, double-click 'Borough'. In the *Expression* editor field, write the following query:

```
"TYPE_2"  =  'Borough'
```

6. Click *Select Features*

From the expression builder dialog, you can also use *Function list* ► *Recent (Selection)* to make a selection that you have used before. The dialog remembers the last 20 expressions used. See *Expressões* for more information and examples.

Dica

Salve sua seleção em um novo arquivo

Users can save selected features into a **New Temporary Scratch Layer** or a **New Vector Layer** using *Edit* ► *Copy Features* and *Edit* ► *Paste Features as* in the desired format.

Selecionar Feições Por Valor

This selection tool opens the layer's feature form allowing the user to choose which value to look for for each field, whether the search should be case-sensitive, and the operation that should be used. The tool has also autocompletes, automatically filling the search box with existing values.

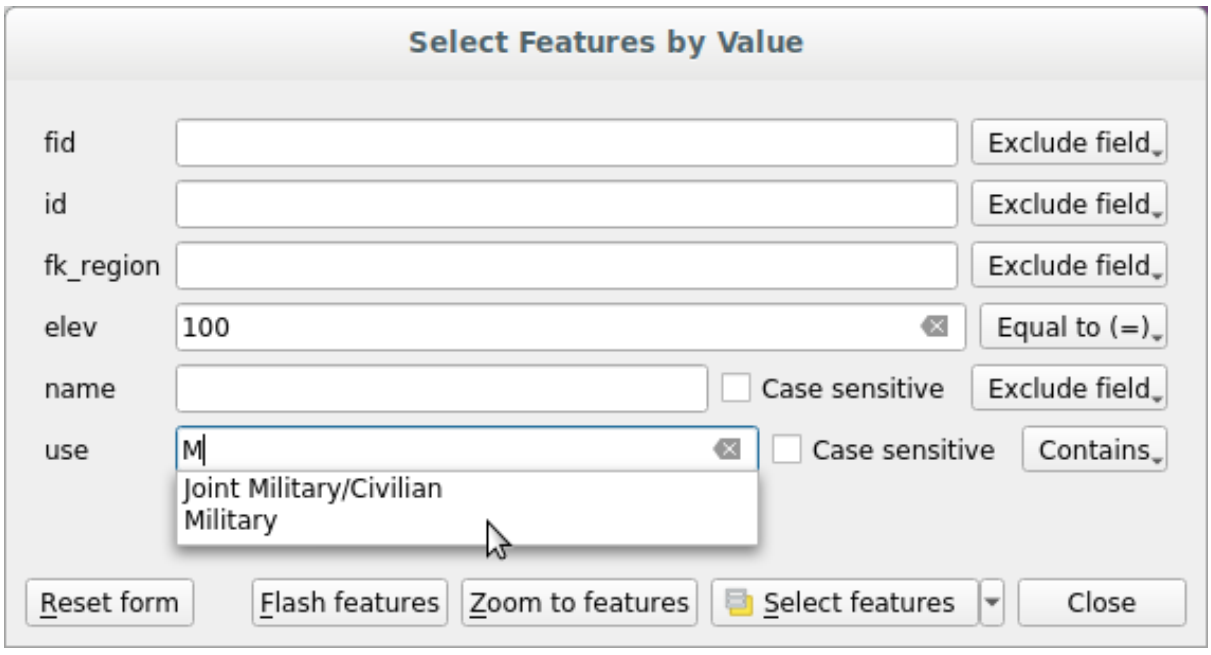


Figura8.10: Filter/Select features using form dialog

Alongside each field, there is a drop-down list with options to control the search behaviour:

Tabela8.3: Query operators per data type

Field search option	String	Numeric	Data
Exclude Field from the search	☑	☑	☑
Equal to (=)	☑	☑	☑
Not equal to (≠)	☑	☑	☑
Greater than (>)		☑	☑
Less than (<)		☑	☑
Greater than or equal to (≥)		☑	☑
Less than or equal to (≤)		☑	☑
Between (inclusive)		☑	☑
Not between (inclusive)		☑	☑
Contains	☑		
Does not contain	☑		
Is missing (null)	☑	☑	☑
Is not missing (not null)	☑	☑	☑
Starts with	☑		
Ends with	☑		

For string comparisons, it is also possible to use the ☑ *Case sensitive* option.

After setting all search options, click *Select features* to select the matching features. The drop-down options are:

- *Selecionar feições*
- *Add to current selection*
- *Remove from current selection*
- *Filtrar seleção atual*

You can also clear all search options using the *Reset form* button.

Uma vez que as condições são definidas, você também pode:

- *Zoom to features* on the map canvas without the need of a preselection
- *Flash features*, highlighting the matching features. This is a handy way to identify a feature without selection or using the Identify tool. Note that the flash does not alter the map canvas extent and would be visible only if the feature is within the bounds of the current map canvas.

8.4.2 Identificando Feições

The Identify Features tool allows you to interact with the map canvas and get information on features or pixels in a pop-up window. It can be used to query most of the layer types supported by QGIS (vector, raster, mesh, point cloud, wms, wfs, …). To identify an element, use either:

- *Exbir* ► *Identificar Feições*
- `Ctrl+Shift+I` (or `Cmd+Shift+I`),
- Identify Features button on the *Attributes* toolbar

Then click on a feature or pixel of the active layer. The identified item gets highlighted in the map canvas while the *Identify Results* dialog opens with detailed information on it. The dialog also shows a set of buttons for advanced configuration.

The Identify Results dialog

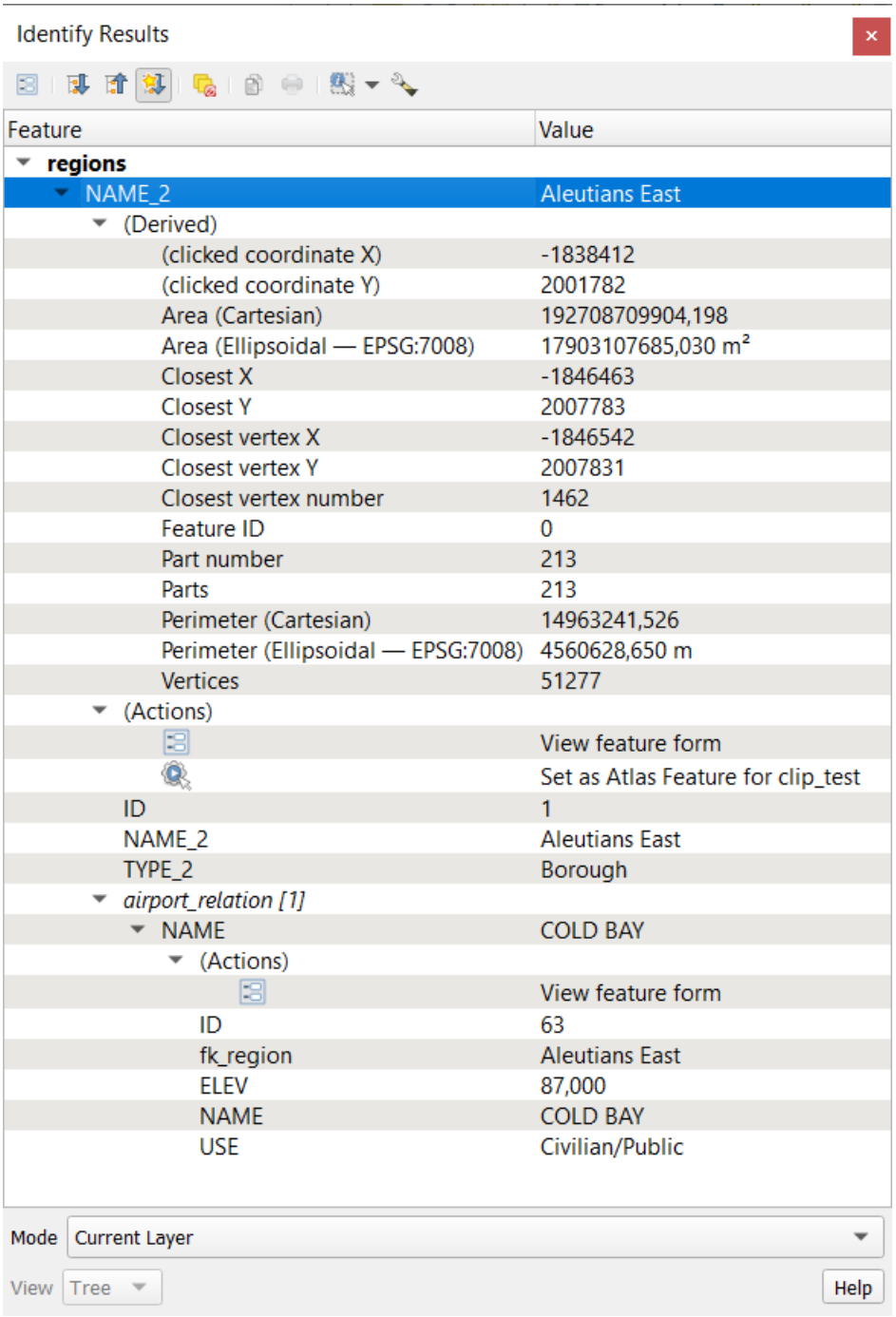


Figura8.11: Identify Results dialog

From bottom to top:

- The *View* controls the general aspect of the dialog and the formatting of the results; it can be set as:

  - **Tree**: this is the default view, and returns the results in a tree-structure where the first item is the name of the layer and its children are its identified element(s).
  - **Table**: available only for raster-based layers, it allows to display the results as a table whose columns are `Layer`, `FID`, `Attribute` and `Value`
  - or **Graph**: available only for raster-based layers

- The *Mode* helps you select the layers from which results could be returned. These layers should be set visible, displaying data in the map canvas, and set *identifiable* from the *Project properties* ► *Data Sources* ► *Layers capabilities*. Available modes are:
  - **Current layer**: only the layer(s) selected in the *Layers* panel return results. If a group is selected, then results are picked from its leaf layers.
  - **Top down, stop at first**: results are from the layer of the top most feature or pixel under the mouse.
  - **Top down**: results are from the layers with feature or pixel under the mouse.
  - **Layer selection**: opens a contextual menu where the user selects the layer to identify features from. If only a single feature is under the mouse, then the results are automatically displayed.
- In the upper part, the information widget: when you identify a data in the map canvas, this is the place where the *Identify Results* dialog will list details about the clicked (or hovered over, depending on the tool in use) items. Their formatting relies on the *selected view*.

  The information displayed by the identify tool will depend on the type of layer you have selected, whether it is:
  - a *vector layer* (including vector tiles or point cloud data),
  - a *raster layer*,
  - a *mesh layer*.
- When in a tree view, the upper part also displays a handful of tools above the results:
  - Open Form of the current feature
  - Expand tree
  - Collapse tree
  - Expand New Results by Default to define whether the next identified feature's information should be collapsed or expanded
  - Clear Results
  - Copy the identified feature to clipboard, suitable for pasting in a spreadsheet.
  - Print selected HTML response: a text-based formatting of the results to print on paper or save as a `.PDF` file
  - the *interactive identifying tools*: a drop-down menu with tools for selecting on the map canvas features or pixels to identify
  - Under Identify Settings, you can activate whether to:
    - *Auto open form for single feature results*: If checked, each time a single feature is identified, a form opens showing its attributes. This is a handy way to quickly edit a feature's attributes.
    - *Hide derived attributes from results* to only show fields actually defined in the layer
    - *Hide NULL values from results*
    - *Show Relations* to show relations in the vector layer *Identify results tree*
  - Help to access the current documentation

Usando a ferramenta Identificar Feições

In its default display (*View: Tree*), the *Identify Results* panel offers several tools to interact with the layers to query. A smart combination of these tools with the *target layers selector* may greatly improve identification operations:

- Identify Feature(s) by single click or click-and-drag
  - single click or click-and-drag: overlaying features in the target layers are returned
  - right-click: overlaying features from target layers are listed in the contextual menu, grouped by layers. You can then choose to:
    - display the result for a specific feature,
    - display the result for all the features of a specific layer,
    - for vector layers, it is also possible to open its attribute table filtered to the returned features
    - or show all of the returned features.
- Identify Features on Mouse over: move over the map canvas and hovered items in the target layers get highlighted and returned in the results panel.
- Identify Features by Polygon: returns items overlapping a drawn or selected polygon.
  - Draw a polygon (left click to add point, right click to close the polygon) and all the overlaying features from target layers are highlighted and returned in the results panel.
  - Right-click and you get the list of all visible polygon features in the project under the click. Pick an entry and QGIS will return all the features from the target layers that overlap the selected polygon.
- Identify Features by Freehand: returns items overlapping a polygon drawn by freehand. Draw a polygon (left-click to start, move the pointer to shape the area and right-click to close the polygon). All the overlaying features from target layers are highlighted and returned in the results panel.
- Identify Features by Radius returns items overlapping a drawn circle. Draw a circle (left-click to indicate the center point, move the pointer to shape the area or enter the radius in the pop-up text box and left-click or press `Enter` to validate the circle). All the overlaying features from target layers are highlighted and returned in the results panel.

> **Dica**
>
> **Filter the layers to query with the Identify Features tool**
>
> Under *Project* ► *Properties…* ► *Data Sources* ► *Layer Capabilities*, uncheck the *Identifiable* column next to a layer to avoid it being queried when using the Identify Features tool. This is a handy way to return features from only layers that are of interest to you.

Results contextual menu

Other functions can be found in the context menu of the identified item. For example, depending on the layer type, you can:

- interact with the individual identified item:
  - *View feature form* or *Edit feature form*, for vector layers
  - *Zoom to feature*: Zooms the map canvas to the extent of the identified feature or pixel
  - *Identify feature*: Restarts and recenters the search results on a *referenced or referencing vector feature* of the identified feature. Only available if *Show Relations* is checked in Identify Settings.

  - *Copy feature*: Copies the vector geometry and attributes, or the pixel geometry
  - *Toggle feature selection*: Adds or removes identified feature to/from the selection
  - *Copy attribute value*: Copies the value in the *Value* column of the tree item you click on
  - *Copy feature attributes*: Copies the attribute names and values of the identified item
  - *Select features by attribute value*: Selects all vector features in the layer that match the selected attribute value
- control the rendering of the layers, the results panel or the map canvas:
  - *Clear result*: Removes results in the window
  - *Clear highlights*: Removes items highlighted on the map
  - *Highlight all*: Highlights in the map canvas all the items identified in the results dialog, regardless of their layer
  - *Highlight layer*: Highlights in the map canvas all the items identified in the layer of the selected entry
  - *Activate layer*: Selects the layer of the identified item in the *Layers* panel
  - *Layer properties*: Opens the layer properties window of the identified item
  - *Expand all* or *Collapse all* the results tree

8.5 Salvar e Compartilhar Propriedades da Camada

8.5.1 Gerenciar Estilos Personalizados

When a vector layer is added to the map canvas, QGIS by default uses a random symbol/color to render its features. However, you can set a default symbol in *Project* ► *Properties…* ► *Default styles* that will be applied to each newly added layer according to its geometry type.

Most of the time, though, you'd rather have a custom and more complex style that can be applied automatically or manually to the layers (with less effort). You can achieve this by using the *Style* menu at the bottom of the Layer Properties dialog. This menu provides you with functions to create, load and manage styles.

A style stores various information set in the layer or project properties dialog to render or interact with the layer. This includes:

Tabela8.4: Components of stored style and their references

Category	Vector	Raster
Layer Configuration	*Layers capabilities*	
	Display Properties	*Display Properties*
Symbology	*Propriedades da simbologia*	*Propriedades da simbologia* ([1])
3D Symbology	*Propriedades da vista 3D*	N/A
Labels	*Propriedades das etiquetas*	N/A
Fields	*Propriedades dos campos* and *Restrições*	N/A
Attributes Form	*Propriedades do formulário de atributos*	N/A
Actions	*Propriedades de Ações*	N/A
Map Tips	*Display Properties*	*Display Properties*
Diagrams	*Propriedades dos diagramas*	N/A
Attribute Table Configuration	*Trabalhando com a tabela de atributos*	
Rendering	*Propriedades de Renderização*	*Propriedades de Renderização* ([1])
Custom Properties	*Propriedades da informação*	*Propriedades da informação*
Geometry Options	*Propriedades de Digitalização*	N/A
Relations	*Setting relations between multiple layers*	N/A
Temporal Properties	*Propriedades Temporais*	*Propriedades Temporais*
Legend Settings	*Propriedades da legenda*	N/A
Elevation Properties	*Elevation Properties*	*Elevation Properties*
Notes	*Notas de camadas*	

By default, the style applied to a loaded layer is named `default`. Once you have got the ideal and appropriate rendering for your layer, you can save it by clicking the *Style* combo box and choosing:

- **Rename Current**: The active style is renamed and updated with the current options
- **Add**: A new style is created using the current options. By default, it will be saved in the QGIS project file. See below to save the style in another file or a database
- **Remover**: Exclua o estilo indesejado, caso tenha mais de um estilo definido para a camada.

At the bottom of the Style drop-down list, you can see the styles set for the layer with the active one checked.

Note that each time you validate the layer properties dialog, the active style is updated with the changes you've made.

You can create as many styles as you wish for a layer but only one can be active at a time. In combination with *Map Themes*, this offers a quick and powerful way to manage complex projects without the need to duplicate any layer in the map legend.

Nota

Given that whenever you apply modifications to the layer properties, changes are stored in the active style, always ensure you are editing the right style to avoid mistakenly altering a style used in a *map theme*.

[1] For raster layers, symbology and rendering items are melted together, meaning that saving/loading one would also save/load items of/from the other.

> **Dica**
>
> **Manage styles from layer context menu**
>
> Right-click on the layer in the *Layers* panel to copy, paste, add or rename layer styles. For both vector and raster layers, you can choose which style categories to copy/paste. Choose *All Style Categories* or one of the available categories, depending on the layer type. Note that for raster layers, the *Symbology* and *Rendering* categories are always copied together.

8.5.2 Storing Styles in a File or a Database

While styles created from the *Style* combo box are by default saved inside the project and can be copied and pasted from layer to layer in the project, it's also possible to save them outside the project so that they can be loaded in another project.

Salvar como arquivo de texto

Clicking the *Style* ► *Save Style*, you can save the style as a:

- QGIS layer style file (`.qml`)
- SLD file (`.sld`), only available for vector layers

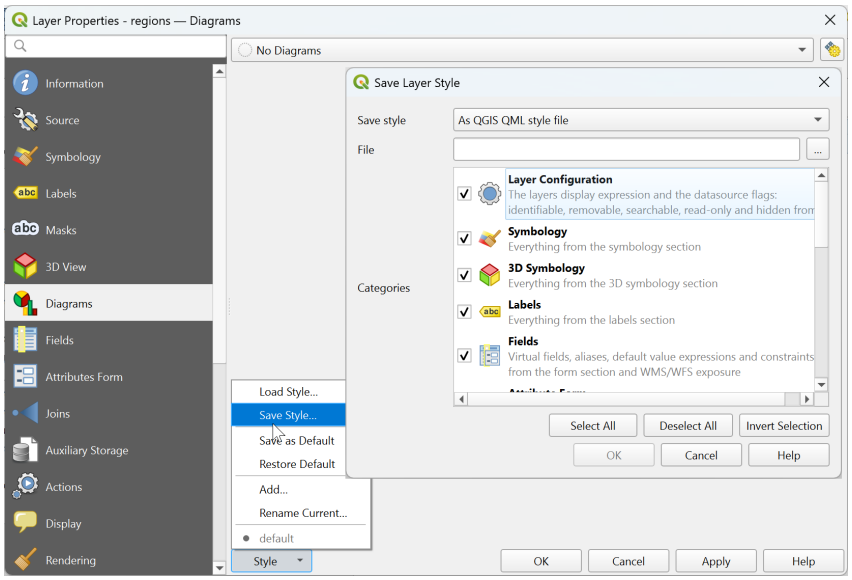


Figura8.12: Vector layer style combo box options

Used on file-based format layers (`.shp`, `.tab`...), *Save as Default* generates a `.qml` file for the layer (with the same name). SLDs can be exported from any type of renderer – single symbol, categorized, graduated or rule-based – but when importing an SLD, either a single symbol or rule-based renderer is created. This means that categorized or graduated styles are converted to rule-based. If you want to preserve those renderers, you have to use the QML format. On the other hand, it can be very handy sometimes to have this easy way of converting styles to rule-based.

Salvar na base de dados

Vector layer styles can also be stored in a database if the layer datasource is a database provider. Supported formats are PostgreSQL, GeoPackage, SpatiaLite, MS SQL Server and Oracle. The layer style is saved inside a table (named `layer_styles`) in the database. Click on *Save Style…* ► *Save in database* then fill in the dialog to define a style name, add a description, a `.ui` file if applicable and to check if the style should be the default style.

You can save several styles for a single table in the database. However, each table can have only one default style. Default styles can be saved in the layer database or in `qgis.db`, a local SQLite database in the active *user profile* directory.

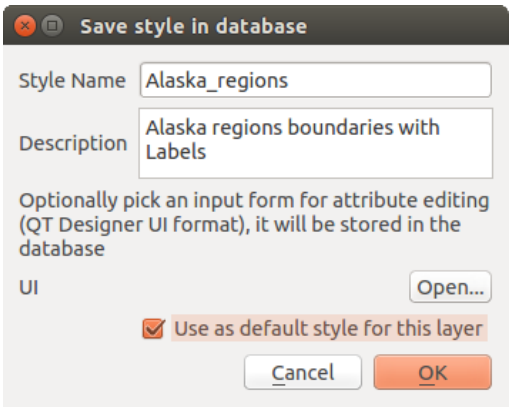


Figura8.13: Salvar estilo na caixa Diálogo do banco de dados

Dica

Compartilhando arquivos de estilo entre bancos de dados

You can only save your style in a database if the layer comes from such a database. You can't mix databases (layer in Oracle and style in MS SQL Server for instance). Use instead a plain text file if you want the style to be shared among databases.

Nota

You may encounter issues restoring the `layer_styles` table from a PostgreSQL database backup. Follow *QGIS layer_style table and database backup* to fix that.

Carregar estilo

When loading a layer in QGIS, if a default style already exists for this layer, QGIS loads the layer with this style. Also *Style* ► *Restore Default* looks for and loads that file, replacing the layer's current style.

Style ► *Load Style* helps you apply any saved style to a layer. While text-file styles (`.sld` or `.qml`) can be applied to any layer whatever its format, loading styles stored in a database is only possible if the layer is from the same database or the style is stored in the QGIS local database.

The *Database Styles Manager* dialog displays a list of styles related to the layer found in the database and all the other styles saved in it, with name and description.

Dica

Compartilhe rapidamente um estilo de camada dentro do projeto

You can also share layer styles within a project without importing a file or database style: right-click on the layer in the *Layers Panel* and, from the *Styles* combo box , copy the style of a layer and paste it to a group or a selection of layers: the style is applied to all the layers that are of the same type (vector vs raster) as the original layer and, in the case of vector layers, have the same geometry type (point, line or polygon).

8.5.3 Arquivo de definição de camada

Layer definitions can be saved as a `Layer Definition File` (`.qlr`) using *Export* ► *Save As Layer Definition File…* in the active layers' context menu. A layer definition file (`.qlr`) includes references to the data source of the layers and their styles. `.qlr` files are shown in the Browser Panel and can be used to add the layers (with the saved style) to the Layers Panel. You can also drag and drop `.qlr` files from the system file manager into the map canvas.

8.6 Documentando seus dados

Além de exibir e simbolizar os dados nas camadas, o QGIS permite preencher:

- **metadata**: information to help people find and understand the dataset, how they can access and use it… these are properties of the datasource and can live out of the QGIS project. Metadata can also be defined for the project file itself.
- **notas**: instruções e comentários sobre a camada no projeto atual

8.6.1 Metadados

In the project or layer properties dialog, the *Metadata* tab provides you with options to create and edit a metadata report on the element. Information to fill concern:

- the data *Identification*: basic attribution of the dataset (parent, identifier, title, abstract, language…);
- the *Categories* the data belongs to. Alongside the **ISO** categories, you can add custom ones;
- the *Keywords* to retrieve the data and associated concepts following a standard based vocabulary;
- the *Access* to the dataset (licenses, rights, fees, and constraints);
- the *Extent* of the dataset, either spatial one (CRS, map extent, altitudes) or temporal;
- the *Contact* of the owner(s) of the dataset;
- the *Links* to ancillary resources and related information;
- the *History* of the dataset.

A summary of the filled information is provided in the *Validation* tab and helps you identify potential issues related to the form. You can then either fix them or ignore them.

Metadata are saved in the project file by default, the *Metadata* drop-down offers options for loading/saving metadata from `.qmd` file and for loading/saving metadata in the "Default" location.

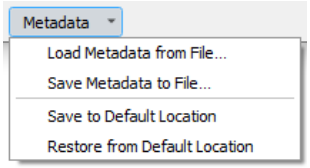


Figura8.14: Metadata load/save options

The “Default” location used by *Save to Default Location* and *Restore from Default Location* changes depending on the underlying data source and on its configuration:

- For PostgreSQL data sources if the configuration option *Allow saving/loading QGIS layer metadata in the database* is checked the metadata are stored inside a dedicated table in the database.
- For GeoPackage data sources *Save to Default Location* always saves the metadata in the internal metadata tables of the GeoPackage.

  When metadata are saved into the internal tables of PostgreSQL or GeoPackage they become available for search and filtering in the browser and in the *layer metadata search panel*.
- For all other file based data sources *Save to Default Location* saves the metadata in a `.qmd` file alongside the file.
- In all other cases *Save to Default Location* saves the metadata in a local `.sqlite` database.

8.6.2 Notas de camadas

Layer notes allow you to document the layer within the current project. They can be place to store important messages for users of the project like to do lists, instructions, warnings, …

From the layer’s contextual menu in *Layers* panel, select *Add layer notes…* and fill the open dialog with necessary texts.

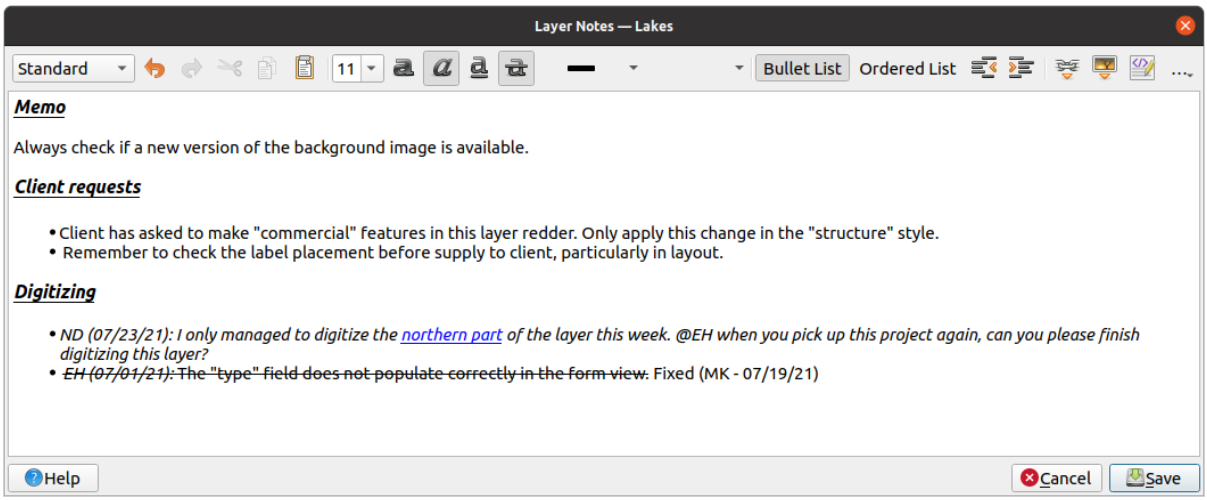


Figura8.15: Adicionando notas a uma camada

The *Add layer notes* dialog provides a html-based multiline text box with a complete set of tools for:

- manipulação de texto: cortar, copiar, colar, desfazer, refazer
- characters formatting, applied to all or parts of the contents: font size and color, bold, italic, underline, strikethrough, background color, URL highlighting
- paragraph structuring: bullet and numbered lists, indentation, predefined headings
- file insertion, even with drag-and-drop
- edição com codificação HTML

From the … drop-down at the far right of the toolbar, you can:

- *Remove all formatting*
- *Remove character formatting*
- *Clear all content*

In the *Layers* panel, a layer with a note is assigned the icon which, upon hover, displays the note. Click the icon to edit the note. You can as well right-click the layer and *Edit layer note…* or *Remove layer note*.

Nota

Notes are part of the *layer style* and can be saved in the `.qml` or `.qlr` file. They can also be transferred from one layer to another while copy-pasting the layer style.

8.7 Armazenando valores em Variáveis

In QGIS, you can use variables to store useful recurrent values (e.g. the project's title, or the user's full name) that can be used in expressions. Variables can be defined at the application's global level, project level, layer level, processing modeler level, layout level, and layout item's level. Just like CSS cascading rules, variables can be overwritten - e.g., a project level variable will overwrite any application global level variables set with the same name. You can use these variables to build text strings or other custom expressions using the `@` character before the variable name. For example in print layout creating a label with this content:

```
This map was made using QGIS [% @qgis_version %]. The project file for this
map is: [% @project_path %]
```

Will render the label like this:

```
This map was made using QGIS 3.4.4-Madeira. The project file for this map is:
/gis/qgis-user-conference-2019.qgs
```

Besides the *preset read-only variables*, you can define your own custom variables for any of the levels mentioned above. You can manage:

- **global variables** from the *Settings* ► *Options* menu
- **project variables** from the *Project Properties* dialog (see *Propriedades do Projeto*)
- **vector layer variables** from the *Layer Properties* dialog (see *Janela de Propriedades de Vetor*);
- **modeler variables** from the *Model Designer* dialog (see *The model designer*);
- **layout variables** do *Layout* painel no layout de impressão (veja *O Painel de Layout*);
- and **layout item variables** from the *Item Properties* panel in the Print layout (see *Opções comuns de itens de esquema*).

To differentiate from editable variables, read-only variable names and values are displayed in italic. On the other hand, higher level variables overwritten by lower level ones are strike through.

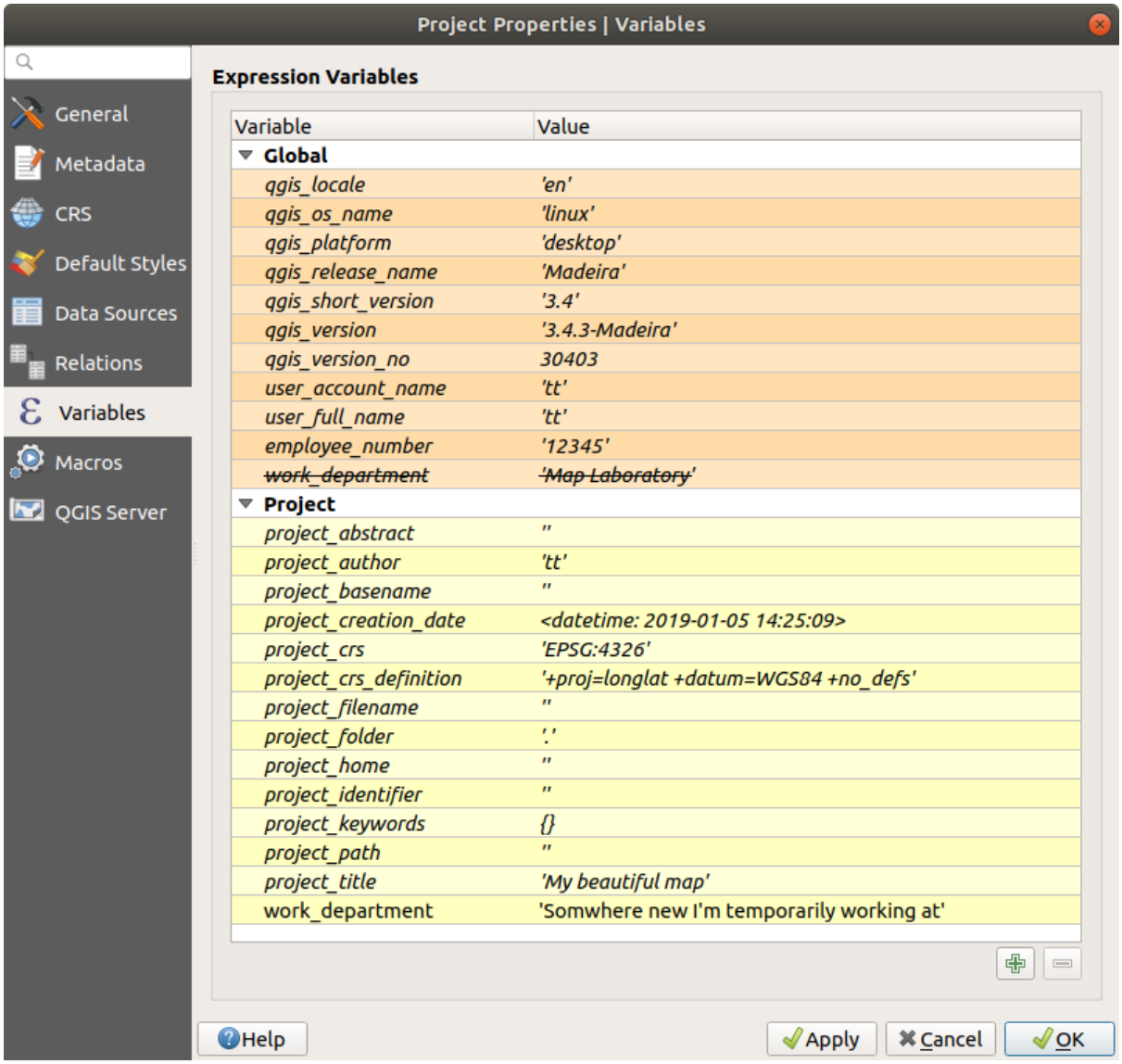


Figura8.16: Editor de variáveis no nível do projeto

Nota

You can read more about variables and find some examples in Nyall Dawson's Exploring variables in QGIS 2.12, part 1, part 2 and part 3 blog posts.

8.8 Elementos comuns

In QGIS, there are some options you'll often have to work with. For convenience, QGIS provides you with special widgets that are presented below.

8.8.1 Seletor de Cor

The color dialog

A caixa de diálogo *Selecionar Cor* aparecerá sempre que você clicar no ícone para escolher uma cor. Os recursos desta caixa de diálogo dependem do estado da caixa de seleção do parâmetro *Usar caixas de diálogo do seletor de cores nativas* em *Configurações* ► *Opções...*► *Geral*. Quando marcada, a caixa de diálogo colorida usada é a nativa do SO no qual o QGIS está sendo executado. Caso contrário, o seletor de cores personalizado QGIS é usado.

The custom color chooser dialog has four different tabs which allow you to select colors by Color ramp, Color wheel, Color swatches or Color picker. With the first two tabs, you can browse to all possible color combinations and apply your choice to the item.

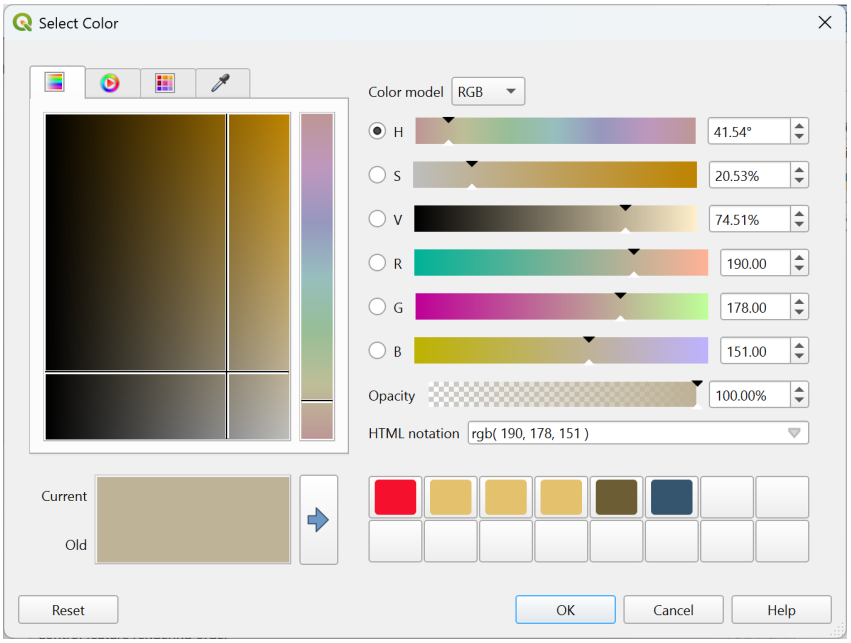


Figura8.17: Color selector ramp tab

In the Color swatches tab, you can choose from a list of color palettes (see *Configurações de cores* for details). All but the *Recent colors* palette can be modified with the Add current color and Remove selected color buttons at the bottom of the frame.

The ... button next to the palette combo box also offers several options to:

- copiar, colar, importar ou exportar cores
- criar, importar ou remover paletas de cores
- add the custom palette to the color selector widget with the *Show in Color Buttons* item (see Fig.8.19)

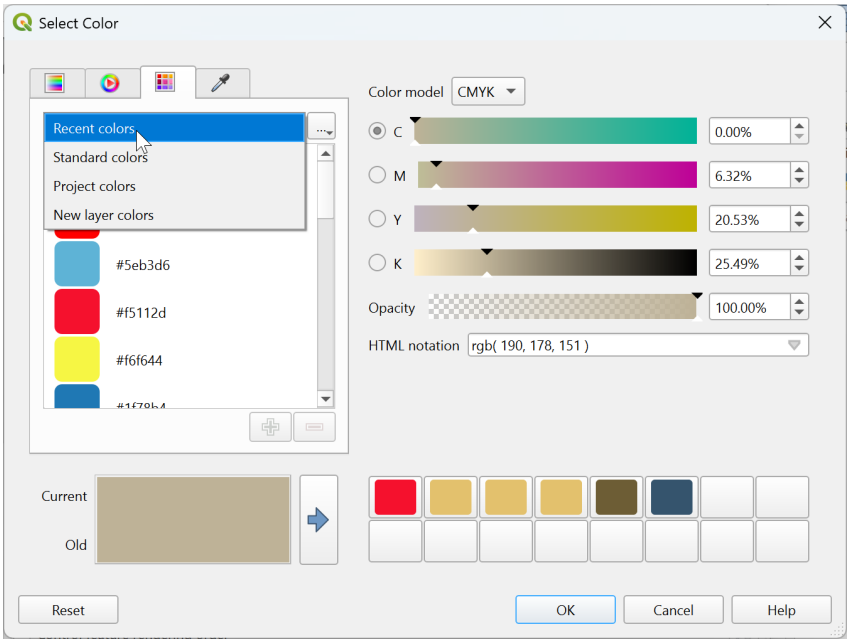


Figura8.18: Color selector swatches tab

Another option is to use the Color picker which allows you to sample a color from under your mouse cursor at any part of the QGIS UI or even from another application: press the space bar while the tab is active, move the mouse over the desired color and click on it or press the space bar again. You can also click the *Sample Color* button to activate the picker.

Whatever method you use, the selected color is always described through options of a *Color model* that defaults to the *project color model*:

- *RGB*: color is set using sliders for `HSV` (Hue, Saturation, Value) or `RGB` (Red, Green, Blue) values.
- *CMYK* (Cyan, Magenta, Yellow, Black)

Modifying a color is as simple as clicking on the color wheel or ramp or on any of the color parameters sliders. You can adjust such parameters with the spinbox beside or by scrolling the mouse wheel over the corresponding slider. You can also type the color in HTML notation. Finally, there is an *Opacity* slider to set transparency level.

The dialog also provides a visual comparison between the *Old* color (applied to object) and the *Current* one (being selected). Using drag-and-drop or pressing the Add color to swatch button, any of these colors can be saved in a slot for easy access.

Dica

Quick color modification

Drag-and-drop a color selector widget onto another one to apply its color.

The color drop-down shortcut

Click the drop-down arrow to the right of the color button to display a widget for quick color selection. This shortcut provides access to:

- a color wheel to pick a color from
- an alpha slider to change color opacity
- the color palettes previously set to *Show in Color Buttons*
- copiar a cor atual e colar em outro elemento
- escolha uma cor em qualquer lugar na tela do computador
- choose a color from the color selector dialog
- drag-and-drop the color from one widget to another for quick modification

Dica

Scroll the mouse wheel over a color selector widget to quickly modify the opacity of the associated color.

Nota

When the color widget is set to a *project color* through the data-defined override properties, the above functions for changing the color are unavailable. You'd first need to *Unlink color* or *Clear* the definition.

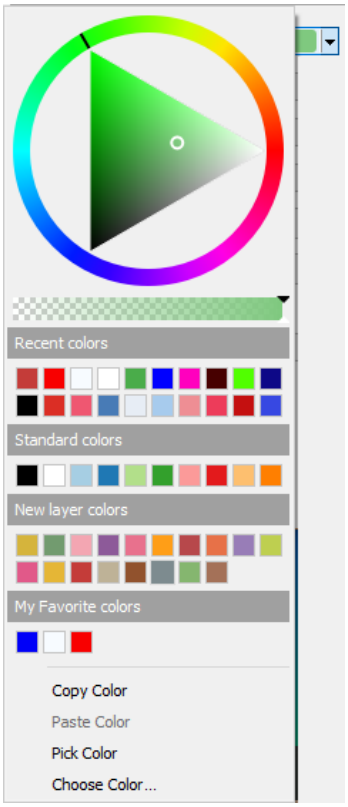


Figura8.19: Quick color selector menu

The color ramp drop-down shortcut

Color ramps are a practical way to apply a set of colors to one or many features. Their creation is described in the *Setting a Color Ramp* section. As for the colors, pressing the color ramp button opens the corresponding color ramp type dialog allowing you to change its properties.

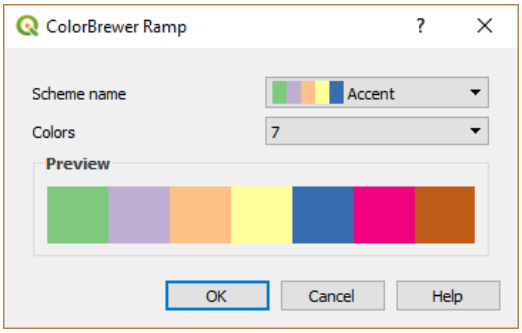


Figura8.20: Customizing a colorbrewer ramp

The drop-down menu to the right of the button gives quick access to a wider set of color ramps and options:

- *Inverter rampa de cor*
- *Clear Current Ramp* to unset any assigned color ramp to the widget (available only in some contexts)
- *Random Colors*: available only in some contexts (e.g., when a color ramp is being used for a layer symbology), checking this entry creates and applies a color ramp with random colors. It also enables a *Shuffle random colors* entry to regenerate a new random color ramp if the current one is not satisfactory.
- a preview of the `gradient` or `catalog: cpt-city` color ramps flagged as **Favorites** in the *Style Manager* dialog
- *All Color Ramps* to access the compatible color ramps database
- guilabel:*Criar nova rampa de cor…* de qualquer tipo suportado que possa ser utilizado no elemtento atual (note que esta rampa de cor não estará disponível em outro lugar, a menos que você a salve na biblioteca)
- *Edit Color Ramp…*, the same as clicking the whole color ramp button
- *Save Color Ramp…*, to save the current color ramp with its customizations in the style library

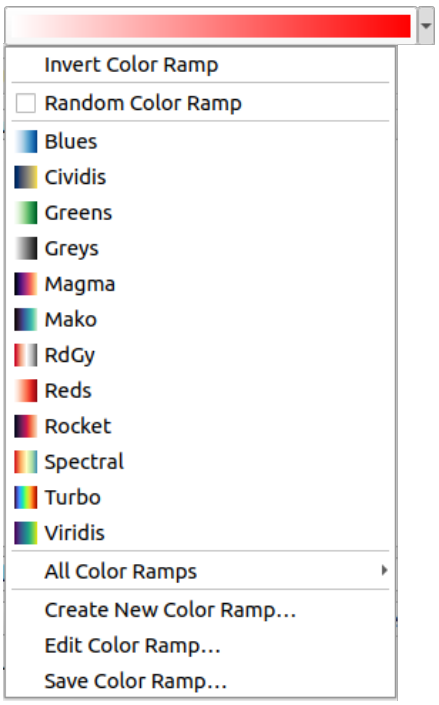


Figura8.21: Quick color ramp selection widget

8.8.2 Symbol Widget

The *Symbol* selector widget is a convenient shortcut when you want to set symbol properties of a feature. Clicking the drop-down arrow shows the following symbol options, together with the features of the *color drop-down widget*:

- *Configure Symbol...*: the same as pressing the symbol selector widget. It opens a dialog to set the *symbol parameters*.
- *Copiar Símbolo* do item atual
- *Colar Símbolo* para o item atual, configuração de velocidade
- *Clear Current Symbol* to unset any assigned symbol to the widget (available only in some contexts)

Dica

Scroll the mouse wheel over a marker or line symbol widget to quickly modify the size of the associated symbol.

8.8.3 Seletor de arquivos remoto ou incorporado

Along with the file selector widget, the ... button will sometimes show a drop-down arrow. This is usually available when using:

- um arquivo SVG em um símbolo ou rótulo
- uma imagem matricial para personalizar símbolos, etiquetas, texturas ou decorações

Pressionar a seta fornecerá um menu para:

- *Select File...*, to load the file from the file system. The file is identified through the file path and QGIS needs to resolve the path in order to display the corresponding image

- *From URL…*, to load the file from a remote URL. As above, the image will only be loaded on successful retrieval of the remote resource
- *Embed File…*, to embed the file inside the current project, style database, or print layout template. The file is then always rendered as part of the item. This is a convenient way to create self-contained projects with custom symbols which can be easily shared amongst different users and installations of QGIS.
- *Extract Embedded File…*, to extract the embedded file from the widget and save it on disk.

8.8.4 Visibility Scale Selector

The visibility scale selector provides options to control the scales at which an element will be made visible in the map canvas. Out of the specified range of scales, the elements are not displayed. It can be applied e.g. to layers, labels or diagrams, from their *Rendering* properties tab.

1. Tick the *Scale dependent visibility* box
2. Fill the *Minimum (exclusive)* box with the most zoomed out desired scale, typing the value or selecting it from the *predefined scales*
3. and/or fill the *Maximum (inclusive)* box with the most zoomed in desired scale

   The Set to current canvas scale button next to the scale boxes sets the current map canvas scale as boundary of the range visibility. Press the arrow next to the button to access scales from layouts' maps and reuse them to fill the box.

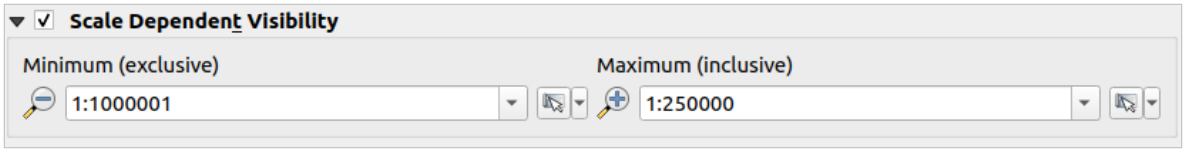


Figura8.22: Visibility scale selector widget

8.8.5 Seletor de Extensão Espacial

The *Extent* selector widget is a convenient shortcut when you want to select a spatial extent to assign to a layer or to limit the actions to run on. Depending on the context, it offers selection between:

- *Current Layer Extent*: e.g. when exporting a layer
- *Calculate from Layer* ►: uses extent of a layer loaded in the current project
- Use current *Map Canvas Extent*
- *Draw on Canvas*: a rectangle whose coordinates are then used
- *Calculate from Bookmark*: uses extent of a saved *bookmark*
- *Calculate from Layout Map*: uses extent of a *layout map*
- Insira as coordenadas como `xmin, xmax, ymin, ymax`

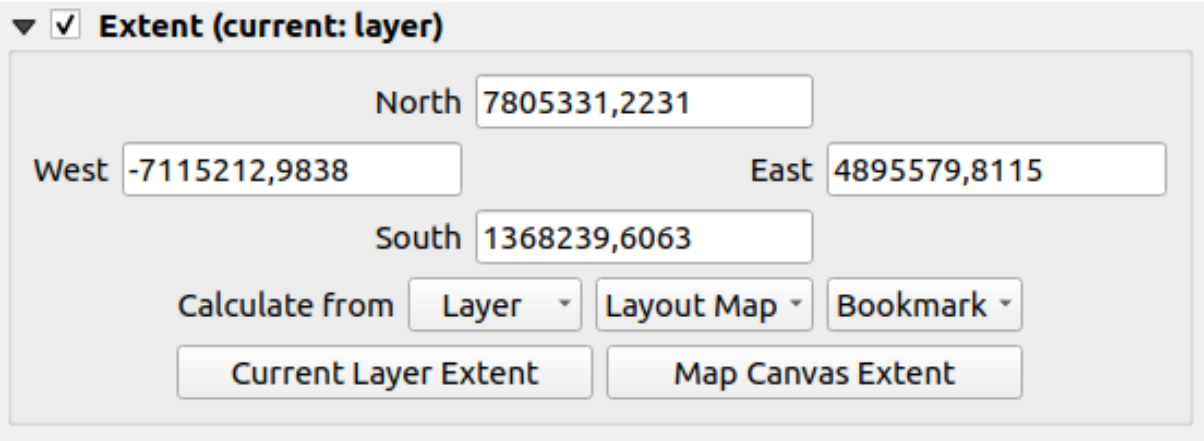


Figura8.23: Extent selector widget

8.8.6 Seletor de Fonte

The *Font* selector widget is a convenient shortcut when you want to set font properties for textual information (feature labels, decoration labels, map legend text, …). Clicking the drop-down arrow shows some or all of the following options:

Figura8.24: Font selector drop-down menu

- *Clear Current Text Format* to unset any assigned text format to the widget (available only in some contexts)
- *Font Size* in the associated unit
- *Recent Fonts* ► menu with the active font checked (at the top)
- *Configure Format...*: same as pressing the font selector widget. It opens the *Text format* dialog, providing advanced formatting options such as color, opacity, orientation, HTML notation, buffer, background, shadow, …

- guilabel:*Copiar Formato* do texto
- *Paste Format* to the text, speeding configuration
- the *color widget* for quick color setting

> **Dica**
>
> Scroll the mouse wheel over a font selector widget to quickly modify the font size of the associated text.

8.8.7 Seletor de Unidade

Size properties of the items (labels, symbols, layout elements, …) in QGIS are not necessarily bound to either the project units or the units of a particular layer. For a large set of properties, the *Unit* selector drop-down menu allows you to tweak their values according to the rendering you want (based on screen resolution, paper size, or the terrain). Available units are:

- *Milímetros*
- *Pontos*
- *Pixels*
- *Polegadas*
- *Percentage*: allows you to set some properties as a percent of another one. For example, this is useful for creation of text formats where the components (buffer size, shadow radius…) nicely scale as the text size is changed, instead of having constant buffer/shadow sizes. So you don't need to adjust those sizes, when the text size changes.
- *Meters at Scale*: This allows you to always set the size in meters, regardless of what the underlying map units are (e.g. they can be in inches, feet, geographic degrees, …). The size in meters is calculated based on the current project ellipsoid setting and a projection of the distances in meters at the center of the current map extent. For maps in a projected coordinate system this is calculated using projected units. For maps in a geographic (latitude/longitude) based system the size is approximated by calculating meter sizes using ellipsoidal calculations for the vertical scale of the map.
- and *Map Units*: The size is scaled according to the map view scale. Because this can lead to too big or too small values, use the button next to the entry to constrain the size to a range of values based on:
  - The *Minimum scale* and the *Maximum scale*: The value is scaled based on the map view scale until you reach any of these scale limits. Out of the range of scale, the value at the nearest scale limit is kept.
  - and/or The *Minimum size* and the *Maximum size* in `mm`: The value is scaled based on the map view scale until it reaches any of these limits; Then the limit size is kept.

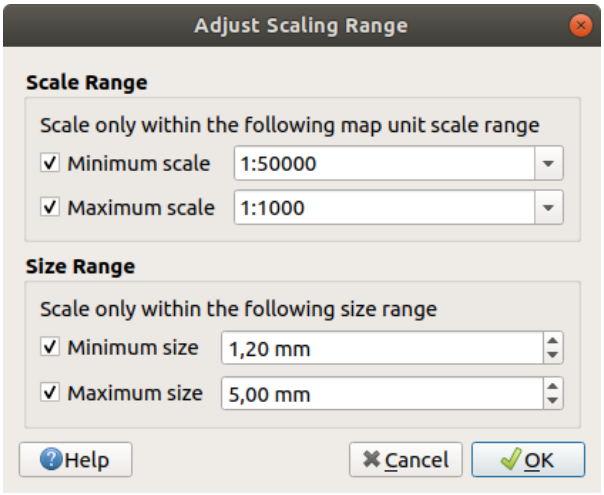


Figura8.25: Adjust scaling range dialog

8.8.8 Formatação de números

Numeric formatters allow formatting of numeric values for display, using a variety of different formatting techniques (for instance scientific notation, currency values, percentage values, custom formatting, etc). One use of this is to set text in a layout scale bar or table, a color ramp legend, an elevation profile plot, …

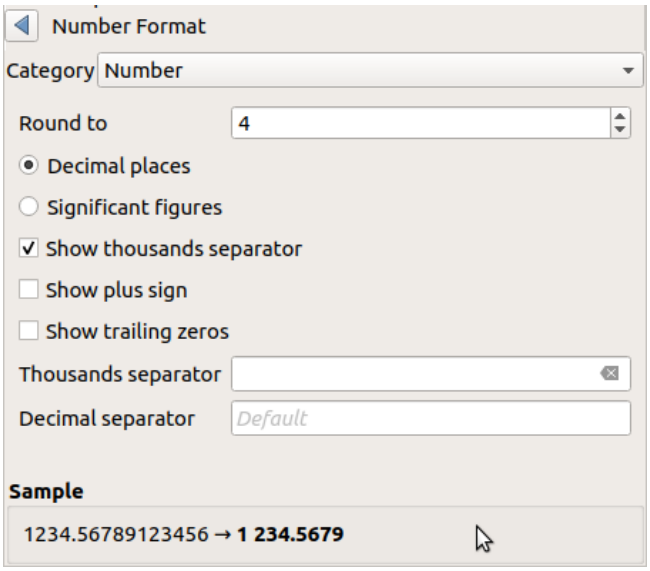


Figura8.26: Formatando valor numérico

Diferentes categorias de formatos são suportadas. Para a maioria, você pode definir parte ou todas as seguintes opções numéricas:

- *Mostrar o separador de milhares*
- *Show plus sign*
- *Show trailing zeros*

Mas eles também podem ter seus ajustes personalizados. Desde que as categorias sejam:

- *General*, the default category: has no setting and displays values as set in the parent widget properties or using the global settings.

- *Custom expression*: allows you to craft a custom *QGIS expression* to format numbers. The expression can use the `@value` variable to retrieve the value to be formatted, and then use any standard QGIS expression function to format this as desired.
- *Número*
  - The value can be *Round to* a self defined number of *Decimal places* or their *Significant figures*
  - personalizar o *Separador de milhas* e *Separador decimal*.
- *Bearing* for a text representation of a direction/bearing using:
  - *Format*: possible ranges of values are `0 to 180°, with E/W suffix`, `-180 to +180°` and `0 to 360°`
  - número de :guilabel:`Casas decimais'.
- *Currency* for a text representation of a currency value.
  - *Prefixo*
  - *Sufixo*
  - número de :guilabel:`Casas decimais'.
- *Fraction* for a vulgar fractional representation of a decimal value (e.g. *1/2* instead of *0.5*)
  - ☐ *Use unicode super/subscript* to show. For example $^{1}/_{2}$ instead of 1/2
  - ☐ *Use dedicated Unicode characters*
  - personalizar o *Separador de milhares*
- guilabel:`Porcentagem` - anexa ``%` aos valores, com configuração de:
  - número de :guilabel:`Casas decimais'.
  - *Scaling* to indicate whether the actual values already represent percentages (then they will be kept as is) or fractions (then they are converted)
- guilabel:*Notação científica* no formulário `2,56e+03`. O número de *Casas decimais* pode ser definido.

A live preview of the settings is displayed under the *Sample* section.

8.8.9 Modos de Mistura

QGIS offers different options for special rendering effects with these tools that you may previously only know from graphics programs. Blending modes can be applied on layers and features, and also on print layout items:

- *Normal*: This is the standard blend mode, which uses the alpha channel of the top pixel to blend with the pixel beneath it. The colors aren't mixed.
- *Lighten*: This selects the maximum of each component from the foreground and background pixels. Be aware that the results tend to be jagged and harsh.
- *Screen*: Light pixels from the source are painted over the destination, while dark pixels are not. This mode is most useful for mixing the texture of one item with another item (such as using a hillshade to texture another layer).
- *Dodge*: Brighten and saturate underlying pixels based on the lightness of the top pixel. Brighter top pixels cause the saturation and brightness of the underlying pixels to increase. This works best if the top pixels aren't too bright. Otherwise the effect is too extreme.
- *Addition*: Adds pixel values of one item to the other. In case of values above the maximum value (in the case of RGB), white is displayed. This mode is suitable for highlighting features.
- *Darken*: Retains the lowest values of each component of the foreground and background pixels. Like lighten, the results tend to be jagged and harsh.

- *Multiply*: Pixel values of the top item are multiplied with the corresponding values for the bottom item. The results are darker.
- *Burn*: Darker colors in the top item cause the underlying items to darken. Burn can be used to tweak and colorize underlying layers.
- *Overlay*: Combines multiply and screen blending modes. Light parts become lighter and dark parts become darker.
- *Soft light*: Very similar to overlay, but instead of using multiply/screen it uses color burn/dodge. This is supposed to emulate shining a soft light onto an image.
- *Hard light*: Hard light is also very similar to the overlay mode. It's supposed to emulate projecting a very intense light onto an image.
- *Difference*: Subtracts the top pixel from the bottom pixel, or the other way around, in order always to get a positive value. Blending with black produces no change, as the difference with all colors is zero.
- *Subtract*: Subtracts pixel values of one item from the other. In the case of negative values, black is displayed.

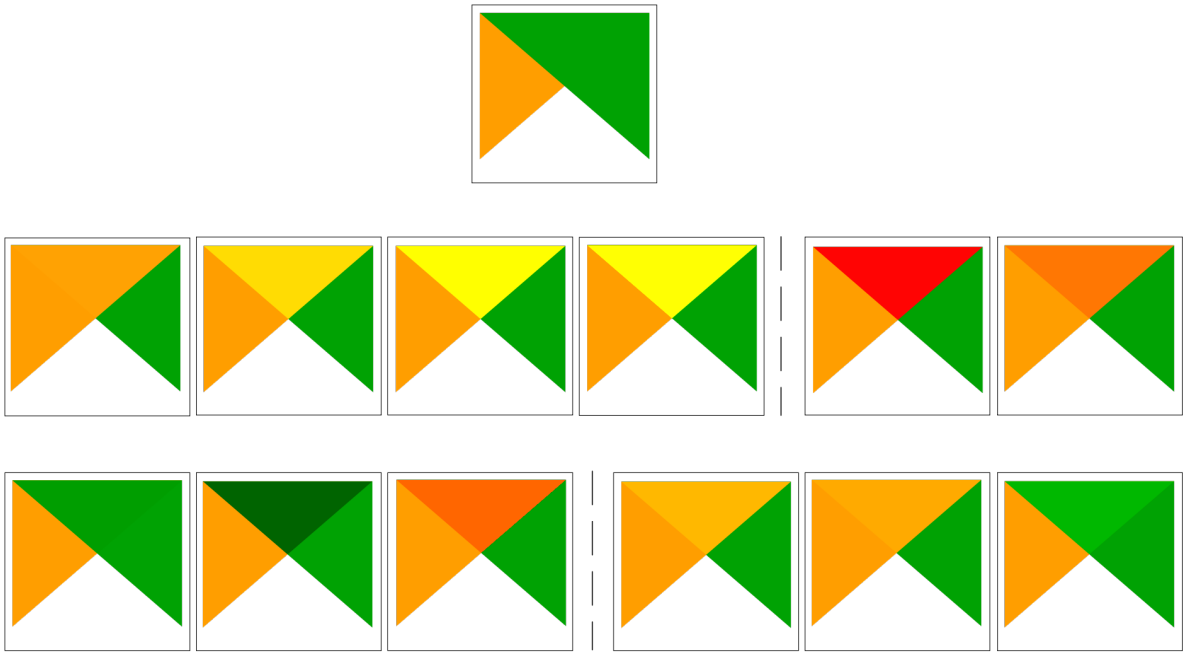


Figura8.27: Examples of blend modes applied to a green feature over an orange
From top to bottom, left to right: Normal – Lighten, Screen, Dodge, Addition – Difference, Subtract – Darken, Multiply, Burn – Overlay, Soft light, Hard light

When a layer is part of a group that *renders layers as a group*, additional blending modes are available for the rendering. They provide methods to clip the render of one layer's content by the content in a second "mask" layer.

- *Masked By Below*: The output is the top pixel, where the opacity is reduced by that of the bottom pixel.
- *Mask Below*: The output is the bottom pixel, where the opacity is reduced by that of the top pixel.
- *Inverse Masked By Below*: The output is the top pixel, where the opacity is reduced by the inverse of the bottom pixel.
- *Inverse Mask Below*: The output is the bottom pixel, where the opacity is reduced by the inverse of the top pixel.
- *Paint Inside Below*: The top pixel is blended on top of the bottom pixel, with the opacity of the top pixel reduced by the opacity of the bottom pixel.

- *Paint Below Inside*: The bottom pixel is blended on top of the top pixel, with the opacity of the bottom pixel reduced by the opacity of the top pixel.

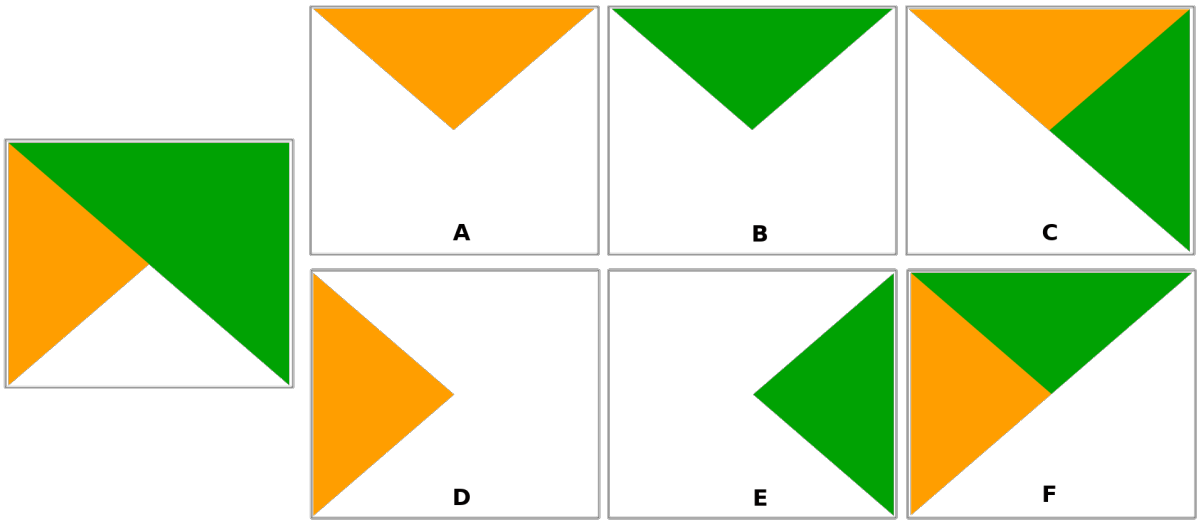


Figura8.28: Examples of blend clipping mode applied to top green layer in a group
A: Mask Below **B**: Masked By Below **C**: Paint Below Inside **D**: Inverse Mask Below **E**: Inverse Masked By Below **F**: Paint Inside Below

8.8.10 Data defined override setup

Next to many options in the vector layer properties dialog or settings in the print layout, you will find a Data defined override icon. Using *expressions* based on layer attributes or item settings, prebuilt or custom functions and *variables*, this tool allows you to set dynamic values for parameters. When enabled, the value returned by this widget is applied to the parameter regardless of its normal value (checkbox, textbox, slider…).

The data defined override widget

Clicking the Data defined override icon shows the following entries:

- *Description…* that indicates if the option is enabled, which input is expected, the valid input type and the current definition. Hovering over the widget also pops up this information.
- *Store data in the project*: a button allowing the property to be stored using the *Propriedades de armazenamento auxiliar* mechanism.
- *Field type*: an entry to select from the layer's fields that match the valid input type.
- *Color*: when the widget is linked to a color property, this menu gives access to the colors defined as part of the current *project's colors* scheme.
- *Variable*: a menu to access the available user-defined *variables*
- *Edit…* button to create or edit the expression to apply, using the *Expression String Builder* dialog. To help you correctly fill in the expression, a reminder of the expected output's format is provided in the dialog.
- botões *Colar* e *Copiar*.
- guilabel:*Limpar* botão para remover a configuração.
- For numeric and color properties, *Assistant…* to rescale how the feature data is applied to the property (more details *below*)

Dica

Use right-click to (de)activate the data override

When the data-defined override option is set up correctly the icon is yellow or . If it is broken, the icon is red or .

You can enable or disable a configured Data-defined override button by simply clicking the widget with the right mouse button.

Dica

Use middle-click to create or edit the expression to apply

You can directly open the *Expression String Builder* dialog to create or edit the expression to apply by simply clicking the Data-defined override widget with the middle mouse button.

Using the data-defined assistant interface

When the Data-defined override button is associated with a size, a width, a rotation, an opacity or a color property, it has an *Assistant…* option that helps you change how the data is applied to the parameter for each feature. Basically, QGIS will scale a range of input values over a range of output values, and render each feature with their corresponding ouput value.

The assistant allows you to:

- Define the *Input* data, i.e.:
  - *Source*: the attribute to represent, using a field or an *expression*
  - the range of values to represent: you can manually enter the values or use the Fetch value range from layer button to fill these fields automatically with the minimum and maximum values returned by the *Source* expression applied to your data. Features whose values are out of the given range will be represented with the property of the closest limit of the output range.
- *Apply transform curve*: by default, output values are applied to input features following a specific scaling method (see below for setting). You can override this logic: enable the transform option, click on the graphic to add break point(s) and drag the point(s) to apply a custom distribution.
- Define the *Output* values: the options vary according to the parameter to define. You can globally set:
  - for color: a linear interpolation of source values is done over a given *color ramp* in order to assign each symbol the correct color. A color to use for symbols with NULL values can also be given.
  - for rotation: a linear interpolation is done over a range of minimum and maximum angles in order to assign each symbol the correct rotation. An angle to use for symbols with NULL values can also be given.
  - for opacity: a polynomial interpolation is done over a range of minimum and maximum opacity values. The *Exponent* field dictates the way source values are mapped to the output range. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite. An opacity value to use for symbols with NULL values can also be given.
  - for width (line symbols): an interpolation is done over a range of minimum and maximum values in order to assign each symbol the correct width. The *Scale method* field controls how the interpolation is done and what the output value represents.

    * **Linear**: linearly interpolates the width of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols width are proportional.
    * **Exponential**: Also interpolates the width, but allows to control the exact power used for scaling through the *Exponent* field. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite.

  The width to use for symbols with NULL values can also be given.

- for size (point symbols): an interpolation is done over a range of minimum and maximum values in order to assign each symbol the correct size "diameter". The *Scale method* field controls how the interpolation is done and what the output value represents.
    * **Radius**: linearly interpolates the diameter of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols diameter are proportional.
    * **Surface**: scales the symbol diameter according to the square root (power of `0.5`) of the *Source* expression value to linearly interpolate the area of the symbol between the *Size from* and the *Size to* when the *Source* expression value varies from *value from* to *value to*. Symbols area are proportional. This method reflects human perception of size variation better than the **radius** one.
    * **Flannery**: Applies a compensation factor (power of `0.57`) to the area interpolation, in order to mitigate the eyes' perceptual underestimation of proportional areas. Maybe relevant only for circular symbols.
    * **Exponential**: Also interpolates the area, but allows to control the exact power used for scaling through the *Exponent* field. Power values greater than `1` will cause the output values distribution to start slowly before accelerating as the source values approach their maximum. Smaller exponents (less than `1`) will do the opposite.

  The size to use for symbols with NULL values can also be given.

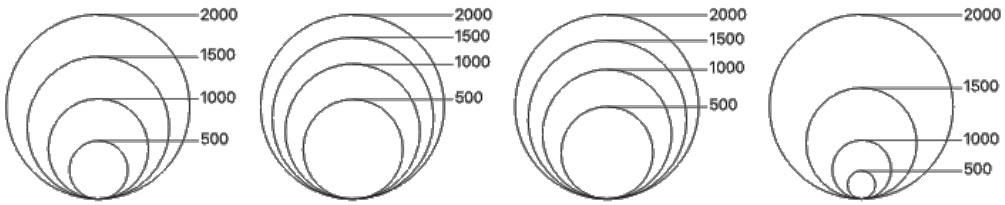


Figura8.29: Scaling values using different methods (from left to right): Radius - Surface - Flannery - Exponential of 2

When compatible with the property, a live-update preview is displayed in the right-hand side of the dialog to help you control the value scaling.

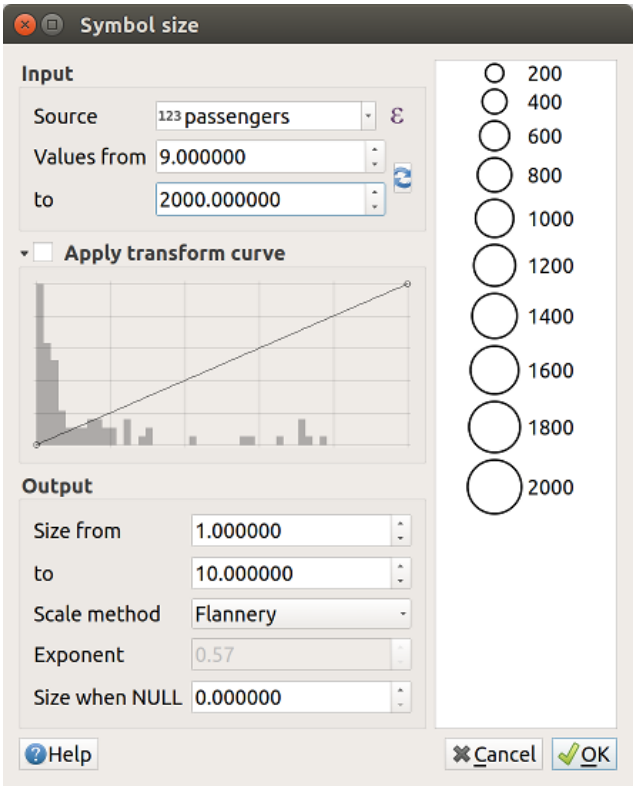


Figura8.30: Scaling feature size based on passengers field's value

The values presented in the varying size assistant above will set the size 'Data-defined override' with the editable formula below:

```
coalesce(scale_polynomial("passengers", 9, 2000, 1, 10, 0.57), 0)
```

8.8.11 Autenticação

Many data providers in QGIS (such as databases, web services or file-based services) require credentials in order to connect. The authentication widget in QGIS helps users supply these credentials securely, and it offers two ways to do so:

- entering basic credentials directly,
- or selecting an existing authentication configuration stored in the encrypted authentication database.

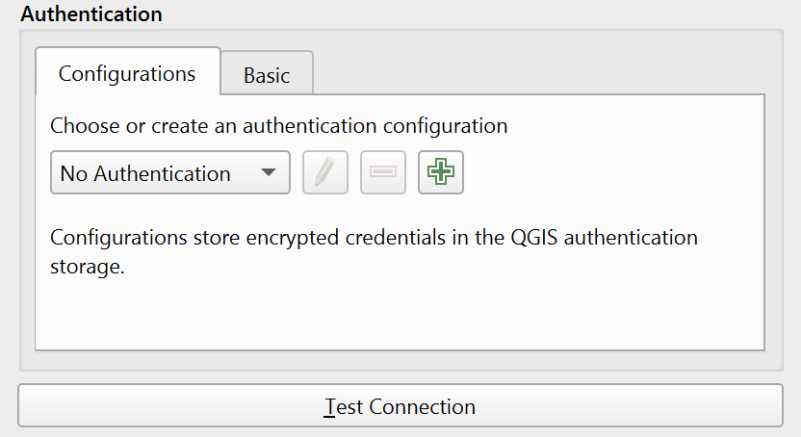


Figura8.31: Authentication widget

Basic

When setting authentication credentials, the *Basic* tab allows you to set:

- *User name*: User name used to connect to the data source.
- *Password*: Password used to connect to the data source.

Convert to configuration will save the user name and password in a new authentication configuration.

You can save any or both of the `User name` and `Password` parameters, in which case they will be used by default each time you need to connect to this database or service. If not saved, you'll be prompted to supply the credentials to connect to the database or service in next QGIS sessions. You can cancel the prompt, in which case the connection will fail. Click the *Ignore for session* button to avoid being prompted again during the current QGIS session. It is also possible to *Ignore for 10 seconds*. This will cancel the prompt but you need to wait 10 seconds before you can attempt to connect again. The connection parameters you entered are stored in a temporary internal cache and returned whenever a username/password for the same database or service is requested, until you end the current QGIS session.

> **Aviso**
>
> **QGIS User Settings and Security**
>
> In the *Authentication* tab, saving **username** and **password** will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you share the project file with someone. Therefore, it is advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab - See *Sistema de Autenticação* for more details) or in a service connection file (see *PostgreSQL Service connection file* for example).

Configurations

QGIS can store credentials under *authentication configurations*. Each configuration safely saves your login details in a portable encrypted database, so you can use them again without typing them every time. These configurations can be safely referenced by their ID tokens in projects and settings.

The authentication configuration selector allows users to choose from existing authentication configurations or create new ones. It provides options to:

- Select an existing authentication configuration from a drop-down list.
- Create a new authentication configuration

- Delete selected configuration
- Edit selected configuration

For more information on the authentication system, see *Sistema de Autenticação*.

CAPÍTULO 9

Subir de nível com expressões

9.1 Expressões

Com base em dados de camada e funções pré-criadas ou definidas pelo usuário, **Expressões** oferecem uma maneira poderosa de manipular valor de atributo, geometria e variáveis para alterar dinamicamente o estilo de geometria, o conteúdo ou a posição do rótulo, o valor do diagrama, a altura de um item de layout, selecionar alguns recursos, criar um campo virtual, …

Nota

Uma lista das funções e variáveis padrões para escrever expressões pode ser encontrada em *Lista de funções*, com informações detalhadas e exemplos.

9.1.1 O Construtor de string de expressão

A caixa de diálogo principal para construir expressões, o *Construtor de string de expressão* está disponível em muitas partes no QGIS e, pode ser acessado quando:

- clicando no botão ;
- *selecionando feições* com a ferramenta Selecionar por Expressão…;
- *editando atributos* com a ferramenta Calculadora de Campo;
- manipulação de parâmetros de simbologia, rótulo ou item de layout com a ferramenta Substituição de dados definida (veja *Data defined override setup*);
- Construindo uma camada de símbolo *geradora de geometria* ;
- Realizando algum *geoprocessamento*.

O diálogo Construtor de Expressão oferece acesso a:

- *Expression* tab which, thanks to a list of *predefined functions*, helps to write and check the expression to use;

- *Aba Editor de Função* a qual ajuda a expandir a lista de funções por meio de criações customizadas de novas funções.

A interface

A aba *Expressão* fornece a interface principal para escrever expressões usando funções, campos de camada e valores. Ele contém os seguintes widgets:

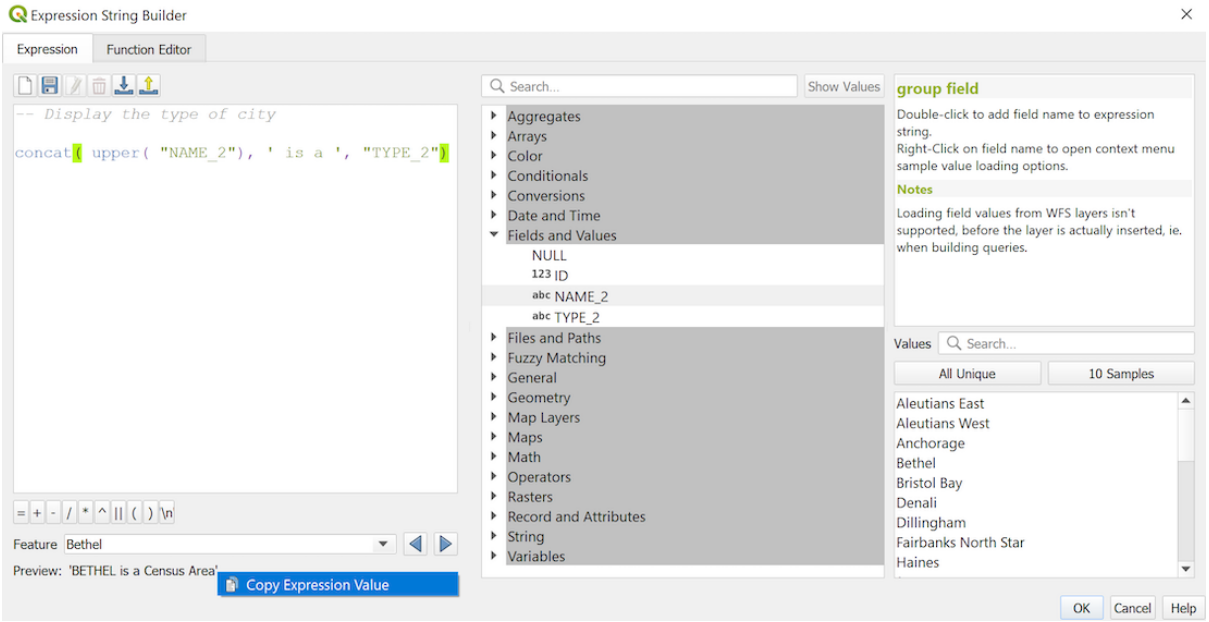


Figura9.1: A guia Expressão

- Uma área de edição de expressão para digitar ou colar expressões. O preenchimento automático está disponível para acelerar a escrita de expressões:
  - Variáveis correspondentes, nomes de funções e nomes de campos para o texto de entrada são mostrados abaixo: use as setas `Up` e `Down` para navegar pelos itens e pressione `Tab` para inserir na expressão ou basta clicar no item desejado.
  - Os parâmetros de função são mostrados ao preenchê-los.

  O QGIS também verifica a correção da expressão e destaca todos os erros usando:
  - *Sublinhado*: para funções desconhecidas, argumentos errados ou inválidos;
  - *Marcador*: para todos os outros erros (por exemplo, parênteses ausentes, caracteres inesperados) em um único local.

> **Dica**
>
> **Documente sua expressão com comentários**
>
> Ao usar expressões complexas, é uma boa prática adicionar texto como um comentário com várias linhas ou comentários embutidos para ajudá-lo a se lembrar.

```
/*
Labels each region with its highest (in altitude) airport(s)
and altitude, eg 'AMBLER : 264m' for the 'Northwest Artic' region
*/
with_variable(
  'airport_alti', -- stores the highest altitude of the region
  aggregate(
```

```
    'airports',
    'max',
    "ELEV", -- the field containing the altitude
    -- and limit the airports to the region they are within
    filter := within( @geometry, geometry( @parent ) )
  ),
    aggregate( -- finds airports at the same altitude in the region
      'airports',
      'concatenate',
      "NAME",
      filter := within( @geometry, geometry( @parent ) )
        and "ELEV" = @airport_alti
    )
    || ' : ' || @airport_alti || 'm'
    -- using || allows regions without airports to be skipped
)
```

- Acima do editor de expressões, um conjunto de ferramentas ajuda você a:
  - Limpar o editor de expressão
  - Criar e gerenciar *expressões do usuário*
- No editor de expressões, você encontra:
  - Um conjunto de operadores básicos para ajudá-lo a construir a expressão
  - Uma indicação do formato esperado quando você está definindo propriedades de uma feição.
  - Uma *Pré visualização de saída* ao vido da expressão (até 60 caracteres), abaliada na primeira feição da camada por padrão. Para visualizar o texto de visualização de saída com mais de 60 caracteres, você pode passar o cursor sobre o texto para exibir um pop-up de dica de ferramenta contendo toda a visualização de saída Para copiar o texto de visualização de saída para sua área de transferência, clique com o botão direito no texto de visualização de saída e selecione *Copiar valor da expressão*.

    Você pode navegar e avaliar outras feições da camada utilizando a *Feature* combobox (os valores serão a partir do *nome exibido* da propriedade da camada).

    Em caso de erro, ele o indica e você pode acessar os detalhes com o hiperlink fornecido.
- Um seletor de funções exibe a lista de funções, variáveis, campos … organizados em grupos. Está disponível uma caixa de pesquisa para filtrar a lista e encontrar rapidamente uma função ou campo específico. Clicar duas vezes em um item o adiciona ao editor de expressão.
- Um painel de ajuda exibe ajuda para cada item selecionado no seletor de funções.

> **Dica**
>
> Pressione `Ctrl+Click` ao passar com o mouse sobre o nome de uma função em uma expressão para exibir automaticamente sua ajuda na caixa de diálogo.

A ferramenta de valores de campo exibida quando um campo é selecionado no seletor de função ajuda a buscar atributos de feições:

- Procure um valor de campo específico
- Exibe a lista de valores *Todos únicos* ou *10 Amostras*. Também disponível ao clicar com o botão direito.

  Quando o campo é mapeado com outra camada ou conjunto de valores, ou seja, se a *ferramenta de campo* for do tipo *Referência de relação*, *Relação de valor* ou *Mapa de valores*, é possível listar todos os valores do campo mapeado (da camada, tabela ou lista referenciada). Além disso, você pode filtrar esta lista para *Mostrar apenas valores em uso* no campo atual.

Clicar duas vezes em um valor de campo na ferramenta o adiciona ao editor de expressão.

> **Dica**
>
> O painel direito, mostrando as funções de ajuda ou valores de campo, pode ser recolhido (invisível) na caixa de diálogo. Pressione o botão *Mostrar valores* ou *Mostrar ajuda* para exibir de novo.

Escrevendo uma expressão

As expressões QGIS são usadas para selecionar feições ou definir valores. Escrever uma expressão no QGIS segue algumas regras:

1. **A caixa de diálogo define o contexto**: se você está acostumado com SQL, provavelmente conhece consultas do tipo *select features from layer where condition* ou *update layer set field = new_value where condition*. Uma expressão QGIS também precisa de todas essas informações, mas a ferramenta que você usa para abrir a caixa de diálogo do construtor de expressões fornece partes delas. Por exemplo, dada uma camada (`building`) com um campo (`height`):

   - clicando na ferramenta Selecionar por expressão significa que você quer "selecionar feições de edifícios". A **condição** é a única informação que você precisa fornecer no elemento de texto de expressão, por exemplo, digite `"altura" > 20` para selecionar construções maiores que 20.
   - com esta seleção feita, pressionando o Calculadora de campo e escolhendo "altura" como *Atualizar campo existente*, você já fornece o comando "atualizar edifícios definir altura = ??? onde altura > 20". Os únicos bits restantes que você precisa fornecer neste caso são o **novo valor**, por exemplo basta digitar `50` na caixa de texto do editor de expressão para definir a altura dos edifícios selecionados anteriormente.

2. **Preste atenção às aspas**: aspas simples retornam um literal, então um texto colocado entre aspas simples (`'145'`) é interpretado como uma string. Aspas duplas lhe darão o valor desse texto, então use-as para campos (`"meucampo"`). Os campos também podem ser usados sem aspas (`meucampo`). Sem aspas para números (`3.16`).

> **Nota**
>
> As funções normalmente tomam como argumento uma string para o nome do campo. Utilize:
>
> ```
> attribute( @atlas_feature, 'height' ) -- returns the value stored in the
> ↪"height" attribute of the current atlas feature
> ```
>
> E não:
>
> ```
> attribute( @atlas_feature, "height" ) -- fetches the value of the attribute␣
> ↪named "height" (e.g. 100), and use that value as a field
>                                 -- from which to return the atlas␣
> ↪feature value. Probably wrong as a field named "100" may not exist.
> ```

> **Dica**
>
> **Use parâmetros nomeados para facilitar a leitura da expressão**
>
> Algumas funções requerem que muitos parâmetros sejam configurados. O mecanismo de expressão suporta o uso de parâmetros nomeados. Isso significa que em vez de escrever a expressão enigmática `clamp( 1, 2, 9)`, você pode usar `clamp( min:=1, valor:=2, max:=9)`. Isso também permite que os argumentos sejam trocados, por exemplo. `clamp( valor:=2, max:=9, min:=1)`. O uso de parâmetros nomeados ajuda a esclarecer a que se referem os argumentos de uma função de expressão, o que é útil quando você está tentando interpretar uma expressão posteriormente!

Alguns casos de uso de expressões

- Na Calculadora de Campo, calcule um campo “pop_density” usando os campos “total_pop” e “area_km2” existentes:

```
"total_pop" / "area_km2"
```

- Rotule ou categorize feições com base em sua área:

```
CASE WHEN $area > 10 000 THEN 'Larger' ELSE 'Smaller' END
```

- Atualiza o campo “density_level” com categorias de acordo com os valores de “pop_density”:

```
CASE WHEN "pop_density" < 50 THEN 'Low population density'
     WHEN "pop_density" >= 50 and "pop_density" < 150 THEN 'Medium population␣
↪density'
     WHEN "pop_density" >= 150 THEN 'High population density'
END
```

- Aplicar um estilo categorizado para todas as feições de acordo com condição do preço médio da casa ser maior ou maior do que 10000€ por metro quadrado:

```
"price_m2" > 10000
```

- Usando a ferramenta “Selecionar por Expressão…”, selecione todas as feições representativas de áreas de “Alta densidade populacional” e as quais possuem valor médio da casa maior do que 10000€ por metro ^{2}:

```
"density_level" = 'High population density' and "price_m2" > 10000
```

  A expressão anterior também pode ser usada para definir quais feições rotular ou mostrar no mapa.

- Select features that overlap a natural zone from the “lands” layer:

```
overlay_intersects( layer:='lands', filter:="zone_type"='Natural' )
```

- Count for each feature the number of buildings they contain:

```
array_length( overlay_contains( layer:='buildings', expression:=@id ) )
```

- Crie um símbolo (tipo) diferente para a camada, usando o gerador de geometria:

```
point_on_surface( @geometry )
```

- Dado uma feição de ponto, gere uma linha fechada (usando `make_line`) em torno de sua geometria:

```
make_line(
  -- using an array of points placed around the original
  array_foreach(
    -- list of angles for placing the projected points (every 90°)
    array:=generate_series( 0, 360, 90 ),
    -- translate the point 20 units in the given direction (angle)
    expression:=project( @geometry, distance:=20, azimuth:=radians( @element )␣
↪)
  )
)
```

- Em uma etiqueta de layout de impressão, exiba o nome das feições de “airports” que estão dentro do item “Mapa 1” do layout:

```
with_variable( 'extent',
               map_get( item_variables( 'Map 1' ), 'map_extent' ),
               aggregate( 'airports', 'concatenate', "NAME",
```

(continua na próxima página)

(continuação da página anterior)

```
                            intersects( @geometry, @extent ), ' ,'
                        )
            )
```

Salvando expressões

Usando o botão Adicionar expressão atual às expressões do usuário acima do quadro do editor de expressões, você pode salvar expressões importantes às quais deseja ter acesso rápido. Eles estão disponíveis no grupo **Expressões do usuário** no painel do meio. Eles são salvos no *user profile* (`<userprofile>/QGIS/QGIS3.ini` file) e estão disponíveis em todas as caixas de diálogo de expressão dentro de todos os projetos do perfil de usuário atual.

Um conjunto de ferramentas disponíveis acima do quadro do editor de expressão ajuda a gerenciar as expressões do usuário:

- Adicionar a expressão atual às expressões do usuário: armazena a expressão no perfil do usuário. Uma etiqueta e um texto de ajuda podem ser adicionados para facilitar a identificação.
- Editar expressão selecionada a partir de expressões do usuário, bem como sua ajuda e rótulo
- Remover a expressão selecionada das expressões do usuário
- Importar expressões de usuário de um arquivo `.json` para a pasta de perfil de usuário ativa
- Exportar expressões do usuário como um arquivo `.json`; todas as expressões do usuário no perfil do usuário `QGIS3.ini` são compartilhadas

9.1.2 Editor de Funções

Com a guia *Editor de Funções*, você pode escrever suas próprias funções na linguagem Python. Isso fornece uma maneira prática e confortável de abordar necessidades específicas que não seriam cobertas pelas funções predefinidas.

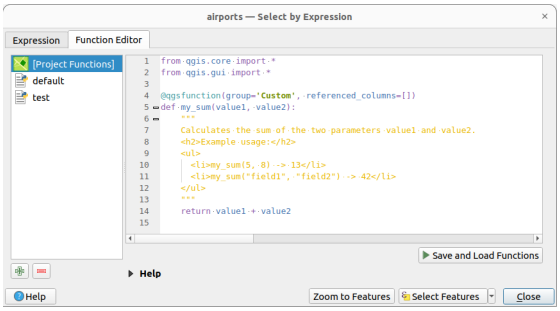


Figura9.2: A guia Editor de Funções

Para criar uma nova função:

1. Pressione o botão Novo Arquivo.
2. Select where to store the new function. You can choose to save it either in a *Function file* or in *Project functions*.
   - If you select *Function file*, you need to enter a name for the Python `.py` file, which is based on a QGIS template file and is stored in the `/python/expressions` folder under the active *user profile* directory, thus available for all projects you may open with that profile.

     Custom Python functions stored as Function files are auto loaded whenever you start a new project under the user profile. Be aware that for sharing such Python functions, you need to share the `.py` file in the `/python/expressions` folder.

- If you select *Project functions*, the new function is stored in the project file and only available for that project.

  Custom Python functions stored as Project functions can be easily shared by sharing the project file where they were saved. When opening a project with project functions, QGIS can load or ignore them depending on whether *embedded Python code is enabled*.

A new item is added in the left panel of the *Function Editor* tab.

3. The right panel displays the content of the file: a python script template.
4. Uncomment the code block and start editing it according to your needs. You can rely on code autocompletion, code formatting and typing helpers, as defined in *Python settings*. Online API documentation for Qt and PyQGIS objects is also accessible pressing the `F1` key.

   You will find in the *Help* widget general information for setting the @qgsfunction decorator.

5. Pressione o botão *Salvar e Carregar Funções*. A função que você escreveu é adicionada à árvore de funções na guia *Expressão*, por padrão, no grupo``Personalizada``.
6. Aproveite sua nova função.
7. Se a função precisar de melhorias, ative a guia *Editor de Funções*, faça as alterações e pressione novamente o botão *Salvar e Carregar Funções* para disponibilizá-las no arquivo, e portanto, em qualquer guia de expressão.

Para excluir uma função personalizada:

1. Habilite a aba *Editor de funções*
2. Selecione a função na lista
3. Press the Remove selected function. The function is removed from the list and, depending on the storage, the corresponding `.py` file is deleted from the user profile folder, or the Python functions are removed from the project file.

> **Nota**
>
> When a project file with Project functions is closed, the corresponding Project functions are unloaded and are no longer available in the QGIS session.

Exemplo

Aqui está um pequeno exemplo de como criar sua própria função `my_sum` que irá operar com dois valores.

```
from qgis.core import *
from qgis.gui import *

@qgsfunction(group='Custom', referenced_columns=[])
def my_sum(value1, value2):
    """
    Calculates the sum of the two parameters value1 and value2.
    <h2>Example usage:</h2>
    <ul>
      <li>my_sum(5, 8) -> 13</li>
      <li>my_sum("field1", "field2") -> 42</li>
    </ul>
    """
    return value1 + value2
```

O decorador `@qgsfunction` aceita os seguintes argumentos:

- O argumento `group` indica em qual grupo a função deverá ser listada no Diálogo de expressões.

- `usesgeometry`: Set this to `True` if the expression requires access to the features geometry. By default `False`.
- `handlesnull`: Set this to `True` if the expression has custom handling for NULL values. If `False` (default), the result will always be NULL as soon as any parameter is NULL.
- `referenced_columns`: An array of attribute names that are required to run the function. Defaults to `[QgsFeatureRequest.ALL_ATTRIBUTES]`. Specifying a subset of fields or an empty list will result in a faster execution.
- `params_as_list`: Set this to `True` to pass the function parameters as a list. If `False` (default), the function will receive the parameters as individual arguments.

The function itself takes any number and type of parameters you want to pass to your function. These should be set before the following optional arguments:

- `feature`: a feição atual
- `parent`: o objeto `QgsExpression`
- `context`: se houver um argumento chamado `context` na última posição, esta variável irá conter um objeto `QgsExpressionContext` , que fornecerá acesso a várias informações adicionais como variáveis de expressões. E.g. `context.variable( 'layer_id' )`

If those arguments are present in the function signature, they will be automatically passed to the function, without the need to specify them in the expression.

A função de exemplo anterior pode ser usada em expressões:

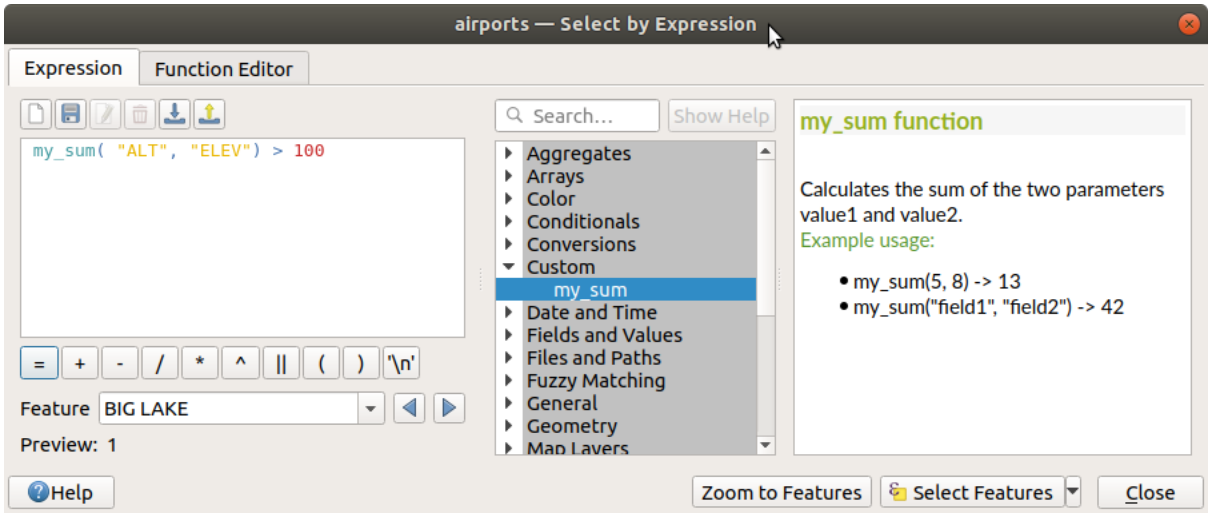


Figura9.3: Função Personalizada adicionada à guia Expressão

Find more examples and additional arguments for custom functions at `qgsfunction`. Further information about creating Python code can be found in the PyQGIS-Developer-Cookbook.

9.2 Lista de funções

As funções, operadores e variáveis disponíveis no QGIS estão listadas abaixo, agrupadas por categorias.

9.2.1 Funções de Agregação

Este grupo contém funções que agregam valores em camadas e campos.

aggregate

Retorna um valor agregado calculado usando feições de outra camada.

Sintaxe	aggregate(layer, aggregate, expression, [filter], [concatenator=''], [order_by]) [] indica argumentos opcionais
Argumentos	• **layer** - uma string, representando um nome de camada ou um ID de camada • **aggregate** - uma string correspondente à agregação a ser calculada. As opções válidas são: – count – count_distinct – count_missing – minimum or min – maximum or max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: primeiro quartil – q3:terceiro quartil – iqr: intervalo interquartil – min_length: comprimento mínimo da string – max_length: comprimento máximo da string – concatenate: unir strings com um concatenador – concatenate_unique: junte strings únicas com um concatenador – collect: cria uma geometria multiparte agregada – array_agg: cria um array de valores agregados • **expression** - sub expressão ou nome do campo para agregar • **filter** - expressão de filtro opcional para limitar as feições usados para calcular a agregação. Campos e geometria são das feições na camada unida. A feição de origem pode ser acessado com a variável @parent. • **concatenator** - string opcional a ser usada para unir valores nas agregações 'concatenate' e 'concatenate_unique' • **order_by** - expressão de filtro opcional para ordenar as feições usadas para calcular o agregado. Campos e geometria são as das feições da camada unida. Por padrão, as feições serão retornadas em uma ordem não especificada.
Exemplos	• `aggregate(layer:='estações_ferroviárias',aggregate:='sum', expression:="passageiros")` → soma de todos os valores do campo de passageiros na camada estações_ferroviárias • `aggregate('estações_ferroviárias','sum', "passageiros"/7)` → calcula uma média diária de "passageiros" dividindo o campo "passageiros" por 7 antes de somar os valores • `aggregate(layer:='estações_ferroviárias',aggregate:='sum', expression:="passageiros",filter:="classe">3)` → soma todos os valores do campo "passageiros" das feições onde o atributo "classe" é maior que 3 apenas • `aggregate(layer:='estações_ferroviárias', aggregate:='concatenate,expression:="nome",concatenator:'. ')` → lista separada por vírgulas do campo de nome para todas as feições na camada estações_ferroviárias • `aggregate(layer:='countries', aggregate:='max', expression:="code", filter:=intersects( @geometry, geometry(@parent) ) )` → The country code of an intersecting country on the layer 'countries' • `aggregate(layer:='rail_stations',aggregate:='sum', expression:="passengers",filter:=contains( @atlas_geometry, @geometry ) )` → sum of all values from the passengers field in the rail_stations within the current atlas feature • `aggregate(layer:='rail_stations',aggregate:='collect', expression:=centroid(@geometry), filter:="region_name" = attribute(@parent,'name') )` → aggregates centroid geometries of the rail_stations of the same region as current feature

array_agg

Retorna um array de valores agregados de um campo ou expressão.

Sintaxe	array_agg(expression, [group_by], [filter], [order_by]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação • **order_by** - expressão opcional a ser usada para ordenar feições usadas para calcular a agregação. Por padrão, as feições serão devolvidas em uma ordem não especificada.
Exemplos	• `array_agg("nome",group_by:="estado")` → lista de valores de name, agrupados por campo state

collect

Retorna a geometria multipartes de geometrias agregadas de uma expressão

Sintaxe	collect(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expressão** - expressão geométrica para agregar • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `collect( @geometry )` → multipart geometry of aggregated geometries • `collect( centroid(@geometry), group_by:="region", filter:= "use" = 'civilian' )` → aggregated centroids of the civilian features based on their region value

concatenate

Retorna todas as strings agregadas de um campo ou expressão unidas por um delimitador.

Sintaxe	concatenate(expression, [group_by], [filter], [concatenator], [order_by]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação • **concatenator** - string opcional a ser usada para unir valores. Vazio por padrão. • **order_by** - expressão opcional a ser usada para ordenar feições usadas para calcular a agregação. Por padrão, as feições serão devolvidas em uma ordem não especificada.
Exemplos	• `concatenate("tow_name",group_by:="state", concatenator:=', ')` → lista separada por vírgulas de town_name, agrupadas pelo campo de state

concatenate_unique

Retorna todas as strings exclusivas de um campo ou expressão unidas por um delimitador.

Sintaxe	concatenate_unique(expression, [group_by], [filter], [concatenator], [order_by]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação • **concatenator** - string opcional a ser usada para unir valores. Vazio por padrão. • **order_by** - expressão opcional a ser usada para ordenar feições usadas para calcular a agregação. Por padrão, as feições serão devolvidas em uma ordem não especificada.
Exemplos	• `concatenate_unique("town_name",group_by:="state", concatenator:=',')` → lista separada por vírgulas de town_name exclusivos, agrupados pelo campo state

count

Retorna a contagem de feições correspondentes.

Sintaxe	count(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `count("stations",group_by:="status")` → contagem de stations, agrupadas pelo campo state

count_distinct

Retorna a contagem de valores distintos.

Sintaxe	count_distinct(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `count_distinct("stations",group_by:="state")` → contagem dos valores de stations distintos, agrupados pelo campo state

count_missing

Retorna a contagem de valores ausentes (NULL).

Sintaxe	count_missing(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `count_missing("stations",group_by:="state")` → contagem dos valores faltantes (NULL) de stations. agrupados pelo campo state

iqr

Retorna o intervalo interquartil calculado de um campo ou expressão.

Sintaxe	iqr(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `iqr("population",group_by:="state")` → intervalo inter quartil dos valores em population, agrupados pelo campo state

majority

Retorna a maioria agregada dos valores (o valor mais comum) de um campo ou expressão.

Sintaxe	majority(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `majority("class", group_by:="state")` → valor de class mais comum, agrupado pelo campo state

max_length

Retorna o comprimento máximo de strings de um campo ou expressão.

Sintaxe	max_length(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `max_length("town_name",group_by:="state")` → tamanho máximo de town_name, agrupado pelo campo state

maximum

Retorna o valor máximo agregado de um campo ou expressão.

Sintaxe	maximum(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `maximum("population",group_by:="state")` → valor máximo em population, agrupado pelo campo state

mean

Retorna o valor médio agregado de um campo ou expressão.

Sintaxe	mean(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `mean("population",group_by:="state")` → mediana dos valores em population, agrupados pelo campo state

median

Retorna o valor médio agregado de um campo ou expressão.

Sintaxe	median(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `median("population",group_by:="state")` → mediana dos valores em population, agrupado pelo campo state

min_length

Retorna o comprimento mínimo de strings de um campo ou expressão.

Sintaxe	min_length(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `min_length("town_name",group_by:="state")` → menor tamanho de town_name, agrupado pelo campo state

minimum

Retorna o valor mínimo agregado de um campo ou expressão.

Sintaxe	minimum(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `minimum("population",group_by:="state")` → valor mínimo de population, agrupado pelo campo state

minority

Retorna a minoria agregada de valores (valor de menor ocorrência) de um campo ou expressão.

Sintaxe	minority(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `minority("class",group_by:="state")` → valor de menor ocorrência em class, agrupado pelo campo state

q1

Retorna o primeiro quartil calculado de um campo ou expressão.

Sintaxe	q1(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `q1("population",group_by:="state")` → primeiro quartil do valor population, agrupado pelo campo state

q3

Retorna o terceiro quartil calculado de um campo ou expressão.

Sintaxe	q3(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `q3("population",group_by:="state")` → terceiro quartil do valor population agrupado pelo campo estado

range

Retorna o intervalo agregado de valores (máximo - mínimo) de um campo ou expressão.

Sintaxe	range(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `range("population",group_by:="state")` → range de valores da população agrupados pelo campo estado

relation_aggregate

Retorna um valor agregado calculado usando todas as feições filhas que coincidirem com a relação da camada.

Sintaxe	relation_aggregate(relation, aggregate, expression, [concatenator=''], [order_by]) [] indica argumentos opcionais
Argumentos	• **relation** - uma strinng, representando o ID da relação • **aggregate** - uma string correspondente à agregação a ser calculada. As opções válidas são: – count – count_distinct – count_missing – minimum or min – maximum or max – sum – mean – median – stdev – stdevsample – range – minority – majority – q1: primeiro quartil – q3:terceiro quartil – iqr: intervalo interquartil – min_length: comprimento mínimo da string – max_length: comprimento máximo da string – concatenate: unir strings com um concatenador – concatenate_unique: junte strings únicas com um concatenador – collect: cria uma geometria multiparte agregada – array_agg: cria um array de valores agregados • **expression** - sub expressão ou nome do campo para agregar • **concatenator** - string opcional a ser usada para unir valores para agregação 'concatenar' • **order_by** - expressão opcional para ordenação das feições usadas no cálculo da agregação. Campos e geometria são provenientes das feições da camada unida. Por padrão, as feições serão retornadas sem ordem específica.
Exemplos	• `relation_aggregate(relation:='my_relation', aggregate:='mean',expression:="passengers")` → valor médio de todas feições filhas encontradas utilizando a relação 'my_relation' • `relation_aggregate('my_relation','sum', "passengers"/7)` → soma do campo passengers divididos por 7 para todas as feições filhas usando a relação 'my_relation' • `relation_aggregate('my_relation','concatenate', "towns", concatenator:=',')` → lista separada por vírgulas do campo towns para todas as camadas filhas encontradas utilizando a relação 'my_relation' • `relation_aggregate('my_relation','array_agg', "id")` → array do campo id de todas as feições filhas encontradas usando a relação 'my_relation'

Leitura adicional: *Setting relations between multiple layers*

stdev

Retorna o valor agregado do desvio padrão de um campo ou expressão.

Sintaxe	stdev(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `stdev("population",group_by:="state")` → desvio padrão do valor em population, agrupado pelo campo state

sum

Retorna o valor somado agregado de um campo ou expressão.

Sintaxe	sum(expression, [group_by], [filter]) [] indica argumentos opcionais
Argumentos	• **expression** - sub expressão do campo a ser agregado • **group_by** - expressão opcional a ser usada para agrupar cálculos agregados • **filter** - expressão opcional a ser usada para filtrar feições usadas para calcular a agregação
Exemplos	• `sum("population",group_by:="state")` → soma dos valores population, agrupados pelo campo state

9.2.2 Funções de Array

Este grupo contém funções para criar e manipular arrays (também conhecidas como estruturas de dados da lista). A ordem dos valores no array é importante, diferentemente da *'map' data structure*, onde a ordem dos pares chave-valor é irrelevante e os valores são identificados por suas chaves.

array

Retorna um array contendo todos os valores passados como parâmetros.

Sintaxe	matriz(valor1, valor2, ...)
Argumentos	• **valor** - um valor
Exemplos	• `matriz(2,10)` → [2, 10] • `matriz(2,10)[0]` → 2

array_all

Retorna VERDADEIRO se um array contiver todos os valores de um dado array.

Sintaxe	array_all(array_a, array_b)
Argumentos	• **matriz_a** - uma matriz • **array_b** - o array de valores a procurar
Exemplos	• `array_all(array(1,2,3),array(2,3))` → TRUE • `array_all(array(1,2,3),array(1,2,4))` → FALSE

array_append

Retorna uma matriz com o valor fornecido adicionado no final.

Sintaxe	array_append(array, value)
Argumentos	• **array** - um array • **value** - o valor a ser adicionado
Exemplos	• `array_append(array(1,2,3),4)` → [1, 2, 3, 4]

array_cat

Retorna um array com todos os arrays informados concatenados.

Sintaxe	array_cat(array1, array2, …)
Argumentos	• **array** - um array
Exemplos	• `array_cat(array(1,2),array(2,3))` → [1, 2, 2, 3]

array_contains

Retorna VERDADEIRO se um array contém o valor informado.

Sintaxe	array_contains(array, value)
Argumentos	• **array** - um array • **value** - valor a pesquisar
Exemplos	• `array_contains(array(1,2,3),2)` → TRUE

array_count

Conta o número de ocorrências de um determinado valor em um array.

Sintaxe	array_count(array, value)
Argumentos	• **array** - um array • **value** - o valor a ser contado
Exemplos	• `array_count(array('a', 'b', 'c', 'b'), 'b')` → 2

array_distinct

Retorna um array contendo valores distintos do array fornecido.

Sintaxe	array_distinct(array)
Argumentos	• **array** - um array
Exemplos	• `matriz_distinta(matriz(1,2,3,2,1))` → [1, 2, 3]

array_filter

Retorna um array com apenas os itens para os quais a expressão é avaliada como verdadeira.

Sintaxe	filtro_matriz(matriz, expressão, [limite=0]) [] indica argumentos opcionais
Argumentos	• **array** - um array • **expression** - uma expressão a ser aplicada em cada item. A variável *@element* será substituída pelo valor atual • **limit** - número máximo de elementos a serem retornados. Use 0 para retornar todos os valores.
Exemplos	• `array_filter(array(1,2,3),@element < 3)` → [1, 2] • `array_filter(array(1,2,3),@element < 3, 1)` → [1]

array_find

Retorna o menor index (0 para o primeiro) do valor dentro de um array. Retorna -1 se o valor não for encontrado.

Sintaxe	array_find(array, value)
Argumentos	• **array** - um array • **value** - valor a pesquisar
Exemplos	• `array_find(array('a', 'b', 'c'), 'b')` → 1 • `array_find(array('a', 'b', 'c', 'b'), 'b')` → 1

array_first

Retorna o primeiro valor de um array.

Sintaxe	array_first(array)
Argumentos	• **array** - um array
Exemplos	• `array_first(array('a','b','c'))` → 'a'

array_foreach

Retorna um array com a expressão aplicada em cada item.

Sintaxe	array_foreach(array, expression)
Argumentos	• **array** - um array • **expression** - an expression to evaluate on each item. The variable *@element* will be replaced by the current value and the variable *@counter* by the current index (starting with 0).
Exemplos	• `array_foreach(array('a','b','c'),upper(@element))` → ['A', 'B', 'C'] • `array_foreach(array(1,2,3),@element + 10)` → [11, 12, 13] • `array_foreach(array(1,2,3),@element + @counter)` → [1, 3, 5]

array_get

Retorna o enésimo (N) valor (0 para o primeiro) ou ultimo -enésimo valor (-1 para o último) de um array.

Sintaxe	array_get(array, pos)
Argumentos	• **array** - um array • **pos** - índice a ser acessado (baseado em 0)
Exemplos	• `array_get(array('a','b','c'),1)` → 'b' • `array_get(array('a','b','c'),-1)` → 'c'

Dica

Você pode usar também o *operador de índice ([])* para pegar um valor de um array.

array_insert

Retorna um array com o valor fornecido adicionado na posição especificada.

Sintaxe	array_insert(array, pos, value)
Argumentos	• **array** - um array • **pos** - posição onde adicionar (baseado em 0) • **value** - o valor a ser adicionado
Exemplos	• `array_insert(array(1,2,3),1,100)` → [1, 100, 2, 3]

array_intersect

Retorna VERDADEIRO se pelo menos um elemento do array1 existe no array2.

Sintaxe	array_intersect(array1, array2)
Argumentos	• **array1** - um array • **array2** - outro array
Exemplos	• `array_intersect(array(1,2,3,4),array(4,0,2,5))` → TRUE

array_last

Retorna o último valor de um array.

Sintaxe	array_last(array)
Argumentos	• **array** - um array
Exemplos	• `array_last(array('a','b','c'))` → 'c'

array_length

Retorna o número de elementos de um array.

Sintaxe	array_length(array)
Argumentos	• **array** - um array
Exemplos	• `array_length(array(1,2,3))` → 3

array_majority

Retorna os valores mais comuns em um array.

Sintaxe	array_majority(array, [option='all']) [] indica argumentos opcionais
Argumentos	• **array** - um array • **option='all'** - uma string especificando o tratamento dos valores retornados. As opções válidas são: – todos: Por padrão, todos os valores mais comuns são devolvidos em um array. – qualquer: Retorna um dos valores mais comuns. – median: Retorna a mediana dos valores mais comuns. Os valores não aritméticos são ignorados. – real_majority: Retorna o valor que ocorre em mais da metade do tamanho do array.
Exemplos	• `array_majority(array(0,1,42,42,43), 'all')` → [42] • `array_majority(array(0,1,42,42,43,1), 'all')` → [42, 1] • `array_majority(array(0,1,42,42,43,1), 'any')` → 1 or 42 • `array_majority(array(0,1,1,2,2), 'median')` → 1.5 • `array_majority(array(0,1,42,42,43), 'real_majority')` → NULL • `array_majority(array(0,1,42,42,43,42), 'real_majority')` → NULL • `array_majority(array(0,1,42,42,43,42,42), 'real_majority')` → 42

array_max

Retorna o valor máximo de um array.

Sintaxe	array_max(array)
Argumentos	• **array** - um array
Exemplos	• `array_max(array(0,42,4,2))` → 42

array_mean

Retorna a média dos valores aritméticos de um array. Valores não numéricos no array são ignorados.

Sintaxe	array_mean(array)
Argumentos	• **array** - um array
Exemplos	• `array_mean(array(0,1,7,66.6,135.4))` → 42 • `array_mean(array(0,84,'a','b','c'))` → 42

array_median

Retorna a mediana dos valores aritméticos de um array. Valores não aritméticos no array são ignorados.

Sintaxe	array_median(array)
Argumentos	• **array** - um array
Exemplos	• `array_median(array(0,1,42,42,43))` → 42 • `array_median(array(0,1,2,42,'a','b'))` → 1.5

array_min

Retorna o valor mínimo de um array.

Sintaxe	array_min(array)
Argumentos	• **array** - um array
Exemplos	• `array_min(array(43,42,54))` → 42

array_minority

Retorna os valores menos comuns de um array.

Sintaxe	array_minority(array, [option='all']) [] indica argumentos opcionais
Argumentos	• **array** - um array • **option='all'** - uma string especificando o tratamento dos valores retornados. As opções válidas são: – all: Padrão, todos os valores menos comuns são retornados em um array. – any: Retorna um dos valores menos comuns. – median: Retorna a mediana dos valores menos comuns. Valores não aritméticos são ignorados. – real_minority: Rentorna os valores com ocorrência menor que a metade do tamanho do array.
Exemplos	• `array_minority(array(0,42,42), 'all')` → [0] • `array_minority(array(0,1,42,42), 'all')` → [0, 1] • `array_minority(array(0,1,42,42,43,1), 'any')` → 0 or 43 • `array_minority(array(1,2,3,3), 'median')` → 1.5 • `array_minority(array(0,1,42,42,43), 'real_minority')` → [42, 43, 0, 1] • `array_minority(array(0,1,42,42,43,42), 'real_minority')` → [42, 43, 0, 1] • `array_minority(array(0,1,42,42,43,42,42), 'real_minority')` → [43, 0, 1]

array_prepend

Retorna um array com o valor informado inserido no início.

Sintaxe	array_prepend(array, value)
Argumentos	• **array** - um array • **value** - o valor a ser adicionado
Exemplos	• `array_prepend(array(1,2,3),0)` → [0, 1, 2, 3]

array_prioritize

Retorna um array ordenado utilizando a ordenação especificada em outro array. Os valores que estiverem presentes no primeiro array mas faltam no segundo array serão inseridos no final do resultado.

Sintaxe	array_prioritize(array, array_prioritize)
Argumentos	• **array** - um array • **array_prioritize** - an array with values ordered by priority
Exemplos	• `array_prioritize(array(1, 8, 2, 5), array(5, 4, 2, 1, 3, 8))` → [5, 2, 1, 8] • `array_prioritize(array(5, 4, 2, 1, 3, 8), array(1, 8, 6, 5))` → [1, 8, 5, 4, 2, 3]

array_remove_all

Retorna um array com todas as entradas do valor informado removidas.

Sintaxe	array_remove_all(array, value)
Argumentos	• **array** - um array • **value** - os valores a serem removidos
Exemplos	• `array_remove_all(array('a','b','c','b'),'b')` → ['a', 'c']

array_remove_at

Retorna um array com o item no índice informado removido. Suporta indice positivo (0 para o primeiro elemento) e negativo (o ultimo -enésimo valor, -1 para o ultimo elemento).

Sintaxe	array_remove_at(array, pos)
Argumentos	• **array** - um array • **pos** - posição a ser removida (baseado em 0)
Exemplos	• `array_remove_at(array(1, 2, 3), 1)` → [1, 3] • `array_remove_at(array(1, 2, 3), -1)` → [1, 2]

array_replace

Retorna um array com o valor fornecido, array, ou um map dos valores substituídos.

Value & variant do array

Retorna um array com o valor fornecido ou array dos valores substituídos por outro valor ou um array de valores

Sintaxe	array_replace(array, before, after)
Argumentos	• **array** - array de entrada • **before** - o valor ou matriz de valores a serem substituídos • **after** - o valor ou matriz de valores a serem usados como substituição
Exemplos	• `array_replace(array('QGIS','SHOULD','ROCK'),'SHOULD', 'DOES')` → ['QGIS', 'DOES', 'ROCK'] • `array_replace(array(3,2,1),array(1,2,3),array(7,8,9))` → [9, 8, 7] • `array_replace(array('Q','G','I','S'),array('Q','S'),'-')` → ['-', 'G', 'I', '-']

Variação por mapeamento

Retorna um array com as chaves do map fornecido substituído pelo valores pareados.

Sintaxe	array_replace(array, map)
Argumentos	• **array** - array de entrada • **map** - map que contém chaves e valores
Exemplos	• `array_replace(array('APP', 'SHOULD', 'ROCK'),map('APP', 'QGIS','SHOULD','DOES'))` → ['QGIS', 'DOES', 'ROCK']

Further reading: *replace*, *regexp_replace*

array_reverse

Retorna a matriz fornecida com valores da matriz na ordem inversa.

Sintaxe	array_reverse(array)
Argumentos	• **array** - um array
Exemplos	• `array_reverse(array(2,4,0,10))` → [10, 0, 4, 2]

array_slice

Retorna uma parcela do array, O fatiamento é definido pelos argumentos start_pos e end_pos.

Sintaxe	array_slice(array, start_pos, end_pos)
Argumentos	• **array** - um array • **start_pos** - o index da posição inicial da fatia (baseada em 0). O índice strart_pos é incluido na fatia. Se você usar um start_pos negativo, o índice será contado a partir do final da lista (baseado em -1). • **end_pos** - O índice da posição final da fatia (base 0). O índice endo_pos é incluido na fatia. Se você usar um endo_pos negativo, o índice é contato a partir do final da lista (base -1)
Exemplos	• `array_slice(array(1,2,3,4,5),0,3)` → [1, 2, 3, 4] • `array_slice(array(1,2,3,4,5),0,-1)` → [1, 2, 3, 4, 5] • `array_slice(array(1,2,3,4,5),-5,-1)` → [1, 2, 3, 4, 5] • `array_slice(array(1,2,3,4,5),0,0)` → [1] • `array_slice(array(1,2,3,4,5),-2,-1)` → [4, 5] • `array_slice(array(1,2,3,4,5),-1,-1)` → [5] • `array_slice(array('Dufour','Valmiera','Chugiak', 'Brighton'),1,2)` → ['Valmiera', 'Chugiak'] • `array_slice(array('Dufour','Valmiera','Chugiak', 'Brighton'),-2,-1)` → ['Chugiak', 'Brighton']

array_sort

Retorna o array fornecido com os elementos ordenados.

Sintaxe	array_sort(array, [ascending=true]) [] indica argumentos opcionais
Argumentos	• **array** - um array • **ascending** - defina esse parâmetro para false para ordenar o array na ordem decrescente.
Exemplos	• `array_sort(array(3,2,1))` → [1, 2, 3]

array_sum

Retorna a soma dos valores aritméticos em uma array. Valores não numéricos na array são ignorados.

Sintaxe	array_sum(array)
Argumentos	• **array** - um array
Exemplos	• `array_sum(array(0,1,39.4,1.6,'a'))` → 42.0

array_to_string

Contatena os elementos do array em uma string separada pelo delimitador e utilizando uma string opcional para valores vazios.

Sintaxe	array_to_string(array, [delimiter=','], [empty_value='']) [] indica argumentos opcionais
Argumentos	• **array** - array de entrada • **delimiter** - string delimitadora usada para separar os elementos concatenados • **empty_value** - string opcional a ser usada em substituição para correspondências vazias (tamanho 0)
Exemplos	• `array_to_string(array('1','2','3'))` → '1,2,3' • `array_to_string(array(1,2,3),'-')` → '1-2-3' • `array_to_string(array('1','','3'),',','0')` → '1,0,3'

Further reading: *string_to_array*

generate_series

Cria uma array contendo uma sequência de números.

Sintaxe	generate_series(start, stop, [step=1]) [] indica argumentos opcionais
Argumentos	• **start** - primeiro valor da sequência • **stop** - valor que termina a sequência uma vez alcançada • **step** - valor usado como incremento entre valores
Exemplos	• `generate_series(1,5)` → [1, 2, 3, 4, 5] • `generate_series(5,1,-1)` → [5, 4, 3, 2, 1]

geometries_to_array

Divide a geometria em geometrias mais simples em um array.

Sintaxe	geometries_to_array(geometry)
Argumentos	• **geometry** - geometria de entrada
Exemplos	• `geometries_to_array(geom_from_wkt('MultiPoint (1 2, 5 21)'))` → An array containing 'Point (1 2)' and 'Point (5 21)' • `geometries_to_array(geom_from_wkt('GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'))` → um array de geometrias com um polígono e uma linha. • `geom_to_wkt(geometries_to_array(geom_from_wkt('GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))'))[0])` → 'Polygon ((5 8, 4 1, 3 2, 5 8))' • `geometries_to_array(geom_from_wkt('MULTIPOLYGON(((5 5,0 0, 0 10,5 5)),((5 5,10 10,10 0,5 5))'))` → um array com duas geometrias de polígonos

regexp_matches

Retorna um array de todas as strings capturadas pelos grupos de captura, na ordem em que os próprios grupos aparecem na expressão regular fornecida em relação a uma string.

Sintaxe	regexp_matches(string, regex, [empty_value='']) [] indica argumentos opcionais
Argumentos	• **string** - string para capturar grupos em relação a expressão regular • **regex** - expressão regular para os grupos de captura • **empty_value** - string opcional a ser usada em substituição para correspondências vazias (tamanho 0)
Exemplos	• `regexp_matches('QGIS=>rocks','(.*)=>(.*)')` → ['QGIS', 'rocks'] • `regexp_matches('key=>','(.*)=>(.*)','empty value')` → ['key', 'empty value']

Further reading: *substr*, *regexp_substr*

string_to_array

Divide a string em um array usando o delimitador fornecido e string opcional fornecida para valores vazios,

Sintaxe	string_to_array(string, [delimiter=','], [empty_value='']) [] indica argumentos opcionais
Argumentos	• **string** - string de entrada • **delimiter** - string delimitadora utilizada para dividir a string de entrada • **empty_value** - string opcional a ser usada em substituição para correspondências vazias (tamanho 0)
Exemplos	• `string_to_array('1,2,3',',')` → ['1', '2', '3'] • `string_to_array('1,,3',',','0')` → ['1', '0', '3']

Further reading: *array_to_string*

9.2.3 Funções de Cor

Este grupo contém funções para manipulação de cores.

color_cmyk

Retorna uma representação em string de uma cor com base em seus componentes ciano, magenta, amarelo e preto

Sintaxe	color_cmyk(cyan, magenta, yellow, black)
Argumentos	• **cyan** - componente ciano da cor, como um valor percentual inteiro de 0 a 100 • **magenta** - componente magenta da cor, como um valor percentual inteiro de 0 a 100 • **yellow** - componente amarelo da cor, como um valor percentual inteiro de 0 a 100 • **balck** - componente preto da cor, como um valor percentual inteiro de 0 a 100
Exemplos	• `color_cmyk(100,50,0,10)` → '0,115,230'

color_cmyka

Retorna uma representação em string de uma cor com base em seus componentes ciano, magenta, amarelo, preto e alfa (transparência)

Sintaxe	color_cmyka(cyan, magenta, yellow, black, alpha)
Argumentos	• **cyan** - componente ciano da cor, como um valor percentual inteiro de 0 a 100 • **magenta** - componente magenta da cor, como um valor percentual inteiro de 0 a 100 • **yellow** - componente amarelo da cor, como um valor percentual inteiro de 0 a 100 • **balck** - componente preto da cor, como um valor percentual inteiro de 0 a 100 • **alpha** - componente alpha como um valor inteiro a partir do 0 (completamente transparente) até 255 (opaco)
Exemplos	• `color_cmyka(100,50,0,10,200)` → '0,115,230,200'

color_cmykf

Returns a color object based on its cyan, magenta, yellow, black and alpha components.

Sintaxe	color_cmykf(cyan, magenta, yellow, black, [alpha=1.0]) [] indica argumentos opcionais
Argumentos	• **cyan** - cyan component as a float value from 0.0 to 1.0 • **magenta** - magenta component as a float value from 0.0 to 1.0 • **yellow** - yellow component as a float value from 0.0 to 1.0 • **black** - black component as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Exemplos	• `color_cmykf(1,0.9,0.81,0.62)` → CMYKA: 1.00,0.90,0.81,0.62,1.00

color_grayscale_average

Applies a grayscale filter to a color and returns it. Returned type is the same as color argument, i.e. a color string representation or a color object.

Sintaxe	color_grayscale_average(color)
Argumentos	• **color** - a color string representation or a color object
Exemplos	• `color_grayscale_average('255,100,50')` → '135,135,135,255' • `color_grayscale_average(color_cmykf(0.6,0.5,0.1,0.8))` → CMYKA: 0.40,0.40,0.40,0.80,1.00

color_hsl

Retorna uma representação em string de uma cor com base em seus atributos de matiz, saturação e luminosidade.

Sintaxe	color_hsl(hue, saturation, lightness)
Argumentos	• **hue** - matiz da cor, um valor inteiro de 0 a 360 • **saturation** - porcentagem de saturação da cor como um valor inteiro de 0 a 100 • **lightness** - porcentagem de luminosidade da cor como um valor inteiro de 0 a 100
Exemplos	• `color_hsl(100,50,70)` → '166,217,140'

color_hsla

Retorna uma representação em string de uma cor com base em seus atributos de matiz, saturação, brilho e alfa (transparência)

Sintaxe	color_hsla(hue, saturation, lightness, alpha)
Argumentos	• **hue** - matiz da cor, um valor inteiro de 0 a 360 • **saturation** - porcentagem de saturação da cor como um valor inteiro de 0 a 100 • **lightness** - porcentagem de luminosidade da cor como um valor inteiro de 0 a 100 • **alpha** - componente alpha como um valor inteiro a partir do 0 (completamente transparente) até 255 (opaco)
Exemplos	• `color_hsla(100,50,70,200)` → '166,217,140,200'

color_hslf

Returns a color object based on its hue, saturation, and lightness attributes.

Sintaxe	color_hslf(hue, saturation, lightness, [alpha=1.0]) [] indica argumentos opcionais
Argumentos	• **hue** - hue of the color, as a float value from 0.0 to 1.0 • **saturation** - saturation of the color as a float value from 0.0 to 1.0 • **lightness** - lightness of the color as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Exemplos	• `color_hslf(0.3,0.52,0.7)` → HSLA: 0.30,0.52,0.70,1.00

color_hsv

Retorna uma representação em string de uma cor baseada na sua matiz, saturação e valores dos atributos.

Sintaxe	color_hsv(hue, saturation, value)
Argumentos	• **hue** - matiz da cor, um valor inteiro de 0 a 360 • **saturation** - porcentagem de saturação da cor como um valor inteiro de 0 a 100 • **value** - porcentagem do valor da cor como um número inteiro de 0 a 100
Exemplos	• `color_hsv(40,100,100)` → '255,170,0'

color_hsva

Retorna uma representação em string de uma cor com base em seus atributos de matiz, saturação, valor e alfa (transparência).

Sintaxe	color_hsva(hue, saturation, value, alpha)
Argumentos	• **hue** - matiz da cor, um valor inteiro de 0 a 360 • **saturation** - porcentagem de saturação da cor como um valor inteiro de 0 a 100 • **value** - porcentagem do valor da cor como um número inteiro de 0 a 100 • **alfa** - componente alfa como um valor inteiro de 0 (completamente transparente) a 255 (opaco)
Exemplos	• `color_hsva(40,100,100,200)` → '255,170,0,200'

color_hsvf

Returns a color object based on its hue, saturation, and value attributes.

Sintaxe	color_hsvf(hue, saturation, value, [alpha=1.0]) [] indica argumentos opcionais
Argumentos	• **hue** - hue of the color, as a float value from 0.0 to 1.0 • **saturation** - saturation of the color as a float value from 0.0 to 1.0 • **value** - value of the color as as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Exemplos	• `color_hsvf(0.4,1,0.6)` → HSVA: 0.40,1.00,0.60,1.00

color_mix

Returns a color mixing the red, green, blue, and alpha values of two provided colors based on a given ratio. Returned type is the same as color arguments, i.e. a color string representation or a color object.

Sintaxe	color_mix(color1, color2, ratio)
Argumentos	• **color1** - a color string or a color object • **color2** - a color string or a color object • **ratio** - a ratio
Exemplos	• `color_mix_rgb('0,0,0','255,255,255',0.5)` → '127,127,127,255' • `color_mix(color_cmykf(0.9,0.9,0.9,0.9),color_cmykf(0.1,0.1,0.1,0.1),0.5)` → CMYKA: 0.50,0.50,0.50,0.50,1.00

color_mix_rgb

Returns a string representing a color mixing the red, green, blue, and alpha values of two provided colors based on a given ratio.

Sintaxe	color_mix_rgb(color1, color2, ratio)
Argumentos	• **color1** - a color string • **color2** - a color string • **ratio** - a ratio
Exemplos	• `color_mix_rgb('0,0,0','255,255,255',0.5)` → '127,127,127,255'

color_part

Returns a specific component from a color string or color object, e.g., the red component or alpha component.

Sintaxe	color_part(color, component)
Argumentos	• **color** - a color string or a color object • **component** - uma string correspondente com o componente da cor a retornar. As opções válidas são: – red: componente vermelho RGB (0-255) – green: componente verde RGB (0-255) – blue: componente azul RGB (0-255) – alfa: valor alfa (transparência) (0-255) – hue: matiz HSV (0-360) – saturation: saturação HSV (0-100) – value: valor HSV (0-100) – hsl_hue: HSL hue (0-360) – hsl_saturation: saturação HSL (0-100) – lightness: luminosidade HSL (0-100) – cyan: ciano componente CMYK (0-100) – magenta: componente magenta CMYK (0-100) – yellow: componente amarelo CMYK (0-100) – black: componente preto CMYK (0-100)
Exemplos	• `color_part('200,10,30','green')` → 10 • `to_int(color_part(color_cmykf(0.1,0.2,0.3,0.9),'black'))` → 90

color_rgb

Retorna uma representação em string de uma cor baseada em seus componentes vermelho, verde e azul.

Sintaxe	color_rgb(red, green, blue)
Argumentos	• **red** - componente vermelho como um valor inteiro de 0 a 255 • **verde** - componente verde como um valor inteiro de 0 a 255 • **blue** - componente azul como um valor inteiro de 0 a 255
Exemplos	• `color_rgb(255,127,0)` → '255,127,0'

color_rgba

Retorna uma representação em string de uma cor baseada em seus componentes vermelho, verde, azul e alpha (transparência).

Sintaxe	color_rgba(red, green, blue, alpha)
Argumentos	• **red** - componente vermelho como um valor inteiro de 0 a 255 • **verde** - componente verde como um valor inteiro de 0 a 255 • **blue** - componente azul como um valor inteiro de 0 a 255 • **alpha** - componente alpha como um valor inteiro a partir do 0 (completamente transparente) até 255 (opaco)
Exemplos	• `color_rgba(255,127,0,200)` → '255,127,0,200'

color_rgbf

Returns a color object based on its red, green, blue and alpha components.

Sintaxe	color_rgbf(red, green, blue, [alpha=1.0]) [] indica argumentos opcionais
Argumentos	• **red** - red component as a float value from 0.0 to 1.0 • **green** - green component as a float value from 0.0 to 1.0 • **blue** - blue component as a float value from 0.0 to 1.0 • **alpha** - alpha component as a float value from 0.0 to 1.0
Exemplos	• `color_rgbf(1.0,0.5,0)` → RGBA: 1.00,0.50,0.00,1.00

create_ramp

Retorna um rampa gradiente a partir de um mapa de cores e passos.

Sintaxe	create_ramp(map, [discrete=false]) [] indica argumentos opcionais
Argumentos	• **map** - um mapa de string de cores e passos • **discrete** - defina esse parâmetro para true para criar uma rampa de cor discreta
Exemplos	• `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → '255,0,0,255'

darker

Returns a darker (or lighter) color. Returned type is the same as color arguments, i.e. a color string representation or a color object.

Sintaxe	darker(color, factor)
Argumentos	• **color** - a color string or a color object • **factor** - um inteiro correspondente ao fator de escurecimento: – se o fator for maior que 100, esta função retornará uma cor mais escura (por exemplo, definir o fator para 200 retornará uma cor com metade do brilho); – se o fator for menor que 100, a cor de retorno é mais clara, mas é recomendado usar a função lighter() para este fim; – se o fator for 0 ou negativo, o valor de retorno não será especificado.
Exemplos	• `darker('200,10,30', 200)` → '100,5,15,255'

Leitura adicional: *lighter*

lighter

Returns a lighter (or darker) color. Returned type is the same as color arguments, i.e. a color string representation or a color object.

Sintaxe	lighter(color, factor)
Argumentos	• **color** - a color string or a color object • **factor** - um iinteiro correspondente ao fator de clareamento: – se o fator for maior que 100, esta função retornará uma cor mais clara (por exemplo, definir o fator para 150 retornará uma cor 50% mais clara); – se o fator for menor que 100, a cor de retorno é mais escura, mas é recomendado usar a função darker() para este fim; – se o fator for 0 ou negativo, o valor de retorno não será especificado.
Exemplos	• `lighter('200,10,30', 200)` → '255,158,168,255'

Leitura adicional: *darker*

project_color

Retorna uma cor do esquema de cores do projeto.

Sintaxe	project_color(name)
Argumentos	• **name** - um nome de cor
Exemplos	• `project_color('Logo color')` → '20,140,50'

Leitura adicional: *configurando cores do projeto*

project_color_object

Returns a color from the project's color scheme. Contrary to project_color which returns a color string representation, project_color_object returns a color object.

Sintaxe	project_color_object(name)
Argumentos	• **name** - um nome de cor
Exemplos	• `project_color_object('Logo color')` → RGBA: 0.08,0.55,0.20,1.00

ramp_color

Retorna uma representação em string de uma rampa de cores.

Variante de rampa salva

Retorna uma representação em string de uma cor a partir de uma rampa salva

Sintaxe	ramp_color(ramp_name, value)
Argumentos	• **ramp_name** - the name of the color ramp as a string, for example 'Spectral' • **value** - posição na rampa para selecionar a cor a partir de um número real entre 0 e 1
Exemplos	• `ramp_color('Spectral',0.3)` → '253,190,115,255'

Nota

As rampas de cores podem variar entre as instalações do QGIS. Esta função pode não retornar o resultado esperado se você mudar seu projeto QGIS entre instalações.

Variante de rampa criada por expressão

Retorna uma representação em string de uma cor a partir de uma rampa criada por expressão

Sintaxe	ramp_color(ramp, value)
Argumentos	• **ramp** - a rampa de cor • **value** - posição na rampa para selecionar a cor a partir de um número real entre 0 e 1
Exemplos	• `ramp_color(create_ramp(map(0,'0,0,0',1,'255,0,0')),1)` → '255,0,0,255'

Leitura adicional: *Setting a Color Ramp*, *The color ramp drop-down shortcut*

ramp_color_object

Returns a color object from a color ramp. Contrary to ramp_color which returns a color string representation, ramp_color_object returns a color object.

Variante de rampa salva

Returns a color object from a saved ramp

Sintaxe	ramp_color_object(ramp_name, value)
Argumentos	• **ramp_name** - the name of the color ramp as a string, for example 'Spectral' • **value** - posição na rampa para selecionar a cor a partir de um número real entre 0 e 1
Exemplos	• `ramp_color_object('Spectral',0.3)` → RGBA: 0.99,0.75,0.45,1.00

Nota

As rampas de cores podem variar entre as instalações do QGIS. Esta função pode não retornar o resultado esperado se você mudar seu projeto QGIS entre instalações.

Variante de rampa criada por expressão

Returns a color object from an expression-created ramp

Sintaxe	ramp_color_object(ramp, value)
Argumentos	• **ramp** - a rampa de cor • **value** - posição na rampa para selecionar a cor a partir de um número real entre 0 e 1
Exemplos	• `ramp_color_object(create_ramp(map(0,color_rgbf(0,0,0),1,color_rgbf(1,0,0))),1)` → RGBA: 1.00,0.00,0.00,1.00

set_color_part

Sets a specific color component for a color string or a color object, e.g., the red component or alpha component.

Sintaxe	set_color_part(color, component, value)
Argumentos	• **color** - a color string or a color object • **component** - uma string correspondente ao componente da cor a ser definida, As opções válidas são: – red: componente vermelho RGB (0-255) – green: componente verde RGB (0-255) – blue: componente azul RGB (0-255) – alfa: valor alfa (transparência) (0-255) – hue: matiz HSV (0-360) – saturation: saturação HSV (0-100) – value: valor HSV (0-100) – hsl_hue: HSL hue (0-360) – hsl_saturation: saturação HSL (0-100) – lightness: luminosidade HSL (0-100) – cyan: ciano componente CMYK (0-100) – magenta: componente magenta CMYK (0-100) – yellow: componente amarelo CMYK (0-100) – black: componente preto CMYK (0-100) • **value** - novo valor do componente da cor, respeitando os intervalos listados acima
Exemplos	• `set_color_part('200,10,30','green',50)` → '200,50,30,255' • `set_color_part(color_cmykf(0.21,0,0.92,0.70),'black',100)` → CMYKA: 0.21,0.00,0.92,1.00,1.00

9.2.4 Funções Condicionais

Este grupo contém funções para lidar com verificações condicionais em expressões.

CASE

CASE é utilizado para verificar uma série de condições e retornar o resultado para a primeira condição que atender. As condições são verificadas sequencialmente, e se uma condição for verdadeira, a verificação para, e o valor correspondente é retornado, Se nenhuma condição for verdadeira, o valor na cláusula ELSE é retornada. No entanto, se nenhuma cláusula ELSE for declarada e nenhuma das condições forem atendidas, NULL é retornado.

CASE

WHEN *condição* THEN *resultado*

[…n]

[ELSE *resultado*]

END

[] indica componentes opcionais

Argumentos	• **WHEN condição** - Uma expressão de condição para avaliar • **THEN resultado** - Se a *condição* for avaliada como Verdadeira, então *resultado* será avaliado e retornado. • **ELSE resultado** - Se nenhuma das condições acima for avaliada como Verdadeiro, o *resultado* será avaliado e retornado.
Exemplos	• `CASE WHEN "name" IS NULL THEN 'None' END` → Retorna a sintring 'None' se o campo "name" for NULL • `CASE WHEN $area > 10000 THEN 'Big property' WHEN $area > 5000 THEN 'Medium property' ELSE 'Small property' END` → Retorna a string 'Big property' se a área for maior que 10000, 'Medium property' se a área estiver entre 5000 e 10000, e 'Small property' para as outras

coalesce

Retorna o primeiro valor não NULO da lista de expressões.

Esta função pode receber qualquer número de argumentos.

Sintaxe	coalesce(expression1, expression2, …)
Argumentos	• **expression** - qualquer expressão ou valor válido, independentemente do tipo.
Exemplos	• `coalesce(NULL, 2)` → 2 • `coalesce(NULL, 2, 3)` → 2 • `coalesce(7, NULL, 3*2)` → 7 • `coalesce("fieldA", "fallbackField", 'ERROR')` → valor do campo fieldA se ele for não-NULL se não o valor de "fallbackField" ou a string 'ERROR' se ambos forem NULL

if

Testa uma condição e retorna um resultado diferente dependendo da verificação condicional.

Sintaxe	if(condition, result_when_true, result_when_false)
Argumentos	• **condition** - a condição que deve ser verificada • **result_when_true** - resultado que será retornado quando a condição for verdadeira ou outro valor que não seja falso. • **result_when_false** - resultado a ser retornado quando a condição for falsa ou outro valor que seja falso como o 0 ou ". NULL também é convertido para falso.
Exemplos	• `if( 1+1=2, 'Sim', 'Não' )` → 'Sim' • `if( 1+1=3, 'Sim', 'Não' )` → 'Não' • `if( 5 > 3, 1, 0)` → 1 • `if( '', 'é verdadeiro (não está vazio)', 'é falso (vazio )`)` → 'é falso (vazio)' • `if(' ', 'é verdadeiro (não está vazio)', 'é falso (vazio)`)` → 'é verdadeiro (não está vazio)' • `if( 0, 'Um', 'Zero' )` → 'Zero' • `if( 10, 'Um', 'Zero' )` → 'Um'

nullif

Retorna um valor NULL se value1 for igual ao value2; caso contrário irá retornar value1. Isso pode ser usado para substituir valores condicionalmente por NULL.

Sintaxe	nullif(value1, value2)
Argumentos	• **value1** - O valor que deve ser usado ou substituído por NULO. • **value2** - O valor de controle que acionará a substituição NULO.
Exemplos	• `nullif('(none)', '(none)')` → NULL • `nullif('text', '(none)')` → 'text' • `nullif("name", '')` → NULL, se name for uma string vazia (ou já NULL), ou name em qualquer outro caso.

regexp_match

Retorna a primeira posição que corresponda com a expressão regular em uma string unicode, ou 0 se a sub string não for encontrada.

Sintaxe	regexp_match(input_string, regex)
Argumentos	• **input_string** - a string a ser usada contra a expressão regular • **regex** - A expressão regular a ser testada contra a string, os caracteres de barra invertida precisam ser escapados (Ex., "\\s" para combinar com espaço em branco ou "\\b" para corresponder a um limite de palavra).
Exemplos	• `regexp_match('QGIS ROCKS','\\sROCKS')` → 5 • `regexp_match('Budač','udač\\b')` → 2

try

Testa uma expressão e retorna seu valor se não tiver erros. Se a expressão retornar algum erro, um valor alternativo será retornado quando fornecido, caso contrário a função retornará NULL.

Sintaxe	try(expression, [alternative]) [] indica argumentos opcionais
Argumentos	• **expression** - expressão que deve ser executada • **alternative** - o resultado que será retornado se a expressão retornar um erro.
Exemplos	• `try( to_int( '1' ), 0 )` → 1 • `try( to_int( 'a' ), 0 )` → 0 • `try( to_date( 'invalid_date' ) )` → NULL

9.2.5 Funções de Conversão

Este grupo contém funções para conversão de um tipo de dado em outro (ex.: string de/para inteiro, binário de/para string, string para data, …).

from_base64

Decodifica uma string codificada em Base64 em um valor binário.

Sintaxe	from_base64(string)
Argumentos	• **string** - a cadeia a ser decodificada
Exemplos	• `from_base64('UUdJUw==')` → 'QGIS'

hash

Cria uma hash de uma string fornecida com um determinado método. Um byte (8 bits) é representado por dois "dígitos" hexadecimais, então 'md4' (16 bytes) gera uma string hexadecimal de 16 * 2 = 32 caracteres de extensão e 'keccak_512' (64 bytes) gera uma string hexadecimal de 64 * 2 = 128 caracteres de extensão.

Sintaxe	hash(string, method)
Argumentos	• **string** - string para transformar em hash • **method** - O método hash entre 'md4', 'md5', 'sha1', 'sha224', 'sha384', 'sha512', 'sha3_224', 'sha3_256', 'sha3_384', 'sha3_512', 'keccak_224', 'keccak_256', 'keccak_384', 'keccak_512'
Exemplos	• `hash('QGIS', 'md4')` → 'c0fc71c241cdebb6e888cbac0e2b68eb' • `hash('QGIS', 'md5')` → '57470aaa9e22adaefac7f5f342f1c6da' • `hash('QGIS', 'sha1')` → 'f87cfb2b74cdd5867db913237024e7001e62b114' • `hash('QGIS', 'sha224')` → '4093a619ada631c770f44bc643ead18fb393b93d6a6af1861fcfece0' • `hash('QGIS', 'sha256')` → 'eb045cba7a797aaa06ac58830846e40c8e8c780bc0676d3393605fae50c • `hash('QGIS', 'sha384')` → '91c1de038cc3d09fdd512e99f9dd9922efadc39ed21d3922e69a4305cc2 • `hash('QGIS', 'sha512')` → 'c2c092f2ab743bf8edbeb6d028a745f30fc720408465ed369421f0a4e20 • `hash('QGIS', 'sha3_224')` → '467f49a5039e7280d5d42fd433e80d203439e338eaabd701f0d6c17d • `hash('QGIS', 'sha3_256')` → '540f7354b6b8a6e735f2845250f15f4f3ba4f666c55574d9e9354575 • `hash('QGIS', 'sha3_384')` → '96052da1e77679e9a65f60d7ead961b28797782314478638 6eb4364 • `hash('QGIS', 'sha3_512')` → '900d079dc69761da113980253aa8ac0414a8bd6d09879a916228f874 • `hash('QGIS', 'keccak_224')` → '5b0ce6acef8b0a121d4ac4f3eaa8503c799ad4e26a3392d1fb2014 • `hash('QGIS', 'keccak_256')` → '991c520aa6815392de24087f61b2ae0fd56abbfeee4a8ca019c101 • `hash('QGIS', 'keccak_384')` → 'c57a3aed9d856fa04e5eeee9b62b6e027cca81ba574116d3cc1f0d4

md5

Cria uma hash md5 de uma string.

Sintaxe	md5(string)
Argumentos	• **string** - string para transformar em hash
Exemplos	• `md5('QGIS')` → '57470aaa9e22adaefac7f5f342f1c6da'

sha256

Cria uma hash sha256 de uma string.

Sintaxe	sha256(string)
Argumentos	• **string** - string para transformar em hash
Exemplos	• `sha256('QGIS')` → 'eb045cba7a797aaa06ac58830846e40c8e8c780bc0676d3393605fae50c05309'

to_base64

Codifica um valor binário em uma string, usando a codificação Base64.

Sintaxe	to_base64(value)
Argumentos	• **value** - o valor binário a ser codificado
Exemplos	• `to_base64('QGIS')` → 'UUdJUw=='

to_bool

Converts a given value to a boolean. The function will return false if the value is NULL, an empty string, an empty list, or 0.

Sintaxe	to_bool(value)
Argumentos	• **value** - value to convert to boolean
Exemplos	• `to_bool('')` → false • `to_bool('123')` → true • `to_bool('false')` → true • `to_bool(0)` → false • `to_bool(1)` → true • `to_bool(null)` → false

to_date

Converte uma string em um objeto de data. Uma string de formatação pode ser fornecida para analisar a string; veja QDate::fromString ou a documentação da função format_date para documentação adicional da fomatação. Por padrão o locale atual do usuário do QGIS será utilizado.

Sintaxe	to_date(string, [format], [language]) [] indica argumentos opcionais
Argumentos	• **string** - string representando um valor de data • **format** - formato usado para converter a cadeia em uma data • **language** - idioma (minúsculo, duas ou três letras, código de idioma ISO 639) usado para converter a cadeia em uma data. Por padrão, a localidade atual do usuário QGIS é usada.
Exemplos	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, se o locale atual utilizar o nome 'June' para o sexto mês, caso contrário irá ocorrer um erro • `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29

Further reading: *format_date*

to_datetime

Converte uma string em um objeto datetime. Uma string opcional pode ser fornecido para avaliar a string; veja QDate::fromString, QTime::fromString ou a documentação da função format_date para documentação adicional do formato. Por padrão a localização atual do usuário do QGIS é utilizada.

Sintaxe	to_datetime(string, [format], [language]) [] indica argumentos opcionais
Argumentos	• **string** - string representando o valor datetime • **format** - formato utilizado para converter a string em datetime • **language** - lingua (caixa baixa, duas ou três letras, código de lingua ISO 639) usada para converter a string em datetime. Por padrão a localização atual do usuário do QGIS é utilizada.
Exemplos	• `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00 • `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34, se a localização atual utilizar o nome 'June' para o sexto mês, caso contrário acontece um erro • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm', 'fr')` → 2019-06-29T12:34

Further reading: *format_date*

to_decimal

Converte uma coordenada de grau, minuto e segundo em seu equivalente decimal.

Sintaxe	to_decimal(value)
Argumentos	• **value** - Uma string grau, minuto, segundo.
Exemplos	• `to_decimal('6°21\'16.445')` → 6.3545680555

to_dm

Converte uma coordenada em grau, minuto.

Sintaxe	to_dm(coordinate, axis, precision, [formatting=]) [] indica argumentos opcionais
Argumentos	• **coordinate** - Um valor de latitude ou longitude. • **axys** - O eixo da coordenada. Ou 'x' ou 'y'. • **precision** - Número de casas decimais. • **formatting** - Designa o tipo da formatação. Valores aceitáveis são NULL (padrão), 'aligned' ou 'suffix'.
Exemplos	• `to_dm(6.1545681, 'x', 3)` → 6°9.274′ • `to_dm(6.1545681, 'y', 4, 'aligned')` → 6°09.2741′N • `to_dm(6.1545681, 'y', 4, 'suffix')` → 6°9.2741′N

to_dms

Converte uma coordenada para grau, minuto, segundo.

Sintaxe	to_dms(coordinate, axis, precision, [formatting=]) [] indica argumentos opcionais
Argumentos	• **coordinate** - Um valor de latitude ou longitude. • **axys** - O eixo da coordenada. Ou 'x' ou 'y'. • **precision** - Número de casas decimais. • **formatting** - Designa o tipo da formatação. Valores aceitáveis são NULL (padrão), 'aligned' ou 'suffix'.
Exemplos	• `to_dms(6.1545681, 'x', 3)` → 6°9′16.445″ • `to_dms(6.1545681, 'y', 4, 'aligned')` → 6°09′16.4452″N • `to_dms(6.1545681, 'y', 4, 'suffix')` → 6°9′16.4452″N

to_int

Converts a string to integer number. If a value cannot be converted to integer the expression is invalid (e.g '123asd' is invalid).

Sintaxe	to_int(string)
Argumentos	• **string** - string a ser convertida para número inteiro
Exemplos	• `to_int('123')` → 123

to_interval

Converte uma string para o tipo intervalo. Pode ser utilizado dias, horas, mês, etc de uma data.

Sintaxe	to_interval(string)
Argumentos	• **string** - um string representando um intervalo. Formatos permitidos são {n} days {n} hours {n} months.
Exemplos	• `to_interval('1 day 2 hours')` → interval: 1.08333 dias • `to_interval( '0.5 hours' )` → interval: 30 minutos • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

to_real

Converts a string to a real number. If a value cannot be converted to real the expression is invalid (e.g '123.56asd' is invalid). Numbers are rounded after saving changes if the precision is smaller than the result of the conversion.

Sintaxe	to_real(string)
Argumentos	• **string** - string a ser convertida para número real
Exemplos	• `to_real('123.45')` → 123.45

to_string

Converts a number to string. The conversion is not locale-aware, see 'format_number' for a locale-aware alternative.

Sintaxe	to_string(number)
Argumentos	• **number** - Interio ou valor real. Número a ser convertido para string.
Exemplos	• `to_string(1.23)` → '1.23'

Further reading: *format_number*

to_time

Converte uma string em um objeto time. Uma string de formato opcional pode ser fornecida para analisar a string; veja QTime::fromString para documentação adicional sobre o formato.

Sintaxe	to_time(string, [format], [language]) [] indica argumentos opcionais
Argumentos	• **string** - string representando um valor time • **format** - formato utilizado para converter a string em time • **language** - lingua (caixa baixa, duas ou três letras, codigo de linguagem ISO 639) usado para converter a string em time
Exemplos	• `to_time('12:30:01')` → 12:30:01 • `to_time('12:34','HH:mm')` → 12:34:00 • `to_time('12:34','HH:mm','fr')` → 12:34:00

Further reading: *format_date*

9.2.6 CRS Functions

This group contains functions to operate on coordinate reference system objects.

crs_from_text

Creates a coordinate reference system from a string definition. The string can represent an authority ID, a WKT definition, or a PROJ string definition of the CRS.

Sintaxe	crs_from_text(definition)
Argumentos	• **definition** - CRS definition
Exemplos	• `crs_from_text('EPSG:3857')` → crs value 'EPSG:3857 - WGS 84 / Pseudo-Mercator' • `crs_from_text('PROJ:+proj=merc +a=6378137 +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +k=1 +units=m +nadgrids=@null +wktext +no_defs +type=crs')` → crs value 'EPSG:3857 - WGS 84 / Pseudo-Mercator'

crs_to_authid

Returns the authority ID string for a coordinate reference system.

Sintaxe	crs_to_authid(crs)
Argumentos	• **crs** - crs value
Exemplos	• `crs_to_authid(crs_from_text('PROJ:+proj=merc +a=6378137 +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +k=1 +units=m +nadgrids=@null +wktext +no_defs +type=crs'))` → 'EPSG:3857'

9.2.7 Funções Personalizadas

Este grupo contém funções criadas pelo usuário. Veja *Editor de Funções* para mais detalhes.

9.2.8 Funções de Data e Hora

Este grupo contém funções para manipulação de data e hora. este grupo compartilha várias funções com os grupos *Funções de Conversão* (to_date, to_time, to_datetime, to_interval) e *Funções de String* (format_date).

Nota

Armazenando data, datetime e intervalos em campos

A habilidade para armazenar valores *data*, *hora* and *datetime* depende diretamente do provedor da fonte (ex., Shapefile aceita o formato *date*, mas não *datetime* ou formato *time*). A seguir estão algumas sugestões para superar essa limitação:

- *date*, *datetime* and *time* podem ser convertidos e armazenados em campos do tipo texto usando a função *format_date()* .
- *Intervals* podem ser armazenados em campos do tipo inteiro ou decimal após a utilização de uma das funções de extração de data (ex., *day()* para pegar o intervalo em dias)

age

Retorna a diferença entre duas datas ou datetimes.

A diferença é retornada como um `Interval` e precisa ser usada com uma das seguintes funções para extrair informações úteis:

- `year`
- `month`
- `week`
- `day`
- `hour`
- `minute`
- `second`

Sintaxe	age(datetime1, datetime2)
Argumentos	• **datetime1** - uma string, date ou datetime representando a data posterior • **datetime2** - uma string, data ou datetime representando a data anterior
Exemplos	• `day(age('2012-05-12','2012-05-02'))` → 10 • `hour(age('2012-05-12','2012-05-02'))` → 240

datetime_from_epoch

Retorna um datetime cujo a data e hora são o número de milisegundos, mseg, que se passaram desde 1970-01-01T00:00:00.000, Coordinated Universal Time (Qt.UTC), e convertido para Qt.LocalTime.

Sintaxe	datetime_from_epoch(int)
Argumentos	• **int** - número (milissegundos)
Exemplos	• `datetime_from_epoch(1483225200000)` → 2017-01-01T00:00:00

day

Extrai o dia de uma data ou o número de dias de um intervalo.

Variante Data

Extrai o dia de um date ou datetime.

Sintaxe	day(date)
Argumentos	• **date** - um valor do tipo data ou datetime
Exemplos	• `day('2012-05-12')` → 12

Variante Intervalo

Calcula a duração em dias de um intervalo.

Sintaxe	day(interval)
Argumentos	• **interval** - valor do intervalo para retornar o número de dias
Exemplos	• `day(to_interval('3 days'))` → 3 • `day(to_interval('3 weeks 2 days'))` → 23 • `day(age('2012-01-01','2010-01-01'))` → 730

day_of_week

Retorna o dia da semana para um date ou datetime específico. O valor retornado varia de 0 até 6, onde 0 corresponde a Domingo e 6 Sábado.

Sintaxe	day_of_week(date)
Argumentos	• **date** - valor date ou datetime
Exemplos	• `day_of_week(to_date('2015-09-21'))` → 1

epoch

Retorna o intervalo em milisegundos entre a epoc unix e um valor de data fornecida.

Sintaxe	epoch(date)
Argumentos	• **date** - um valor do tipo data ou datetime
Exemplos	• `epoch(to_date('2017-01-01'))` → 1483203600000

format_date

Fomata o tipo date ou string em uma string formadata customizada. Usa o formato de string do Qt date/time . Veja QDateTime::toString.

Sintaxe	format_date(datetime, format, [language]) [] indica argumentos opcionais
Argumentos	• **datetime** - valor data, hora ou datetime • **format** - String do modelo usado para formatar a string.

Expressão	Output
d	o dia como número sem zero à esquerda (1 a 31)
dd	o dia como número com um zero à esquerda (01 a 31)
ddd	localização abreveada do nome do dia (ex. 'Seg' a 'Dom')
dddd	o nome longo do dia localizado (ex., 'Segunda-feira' a 'Domingo')
M	o mês como número sem zero à esquerda (1-12)
MM	o mês como número com um zero à esquerda (01-12)
MMM	o nome abreviado do mês localizado (ex., 'Jan' a 'Dez')
MMMM	o nome longo do mês localizado (ex., 'janeiro' a 'dezembro')
yy	o ano como número de dois dígitos (00-99)
yyyy	o ano como número de quatro dígitos

Estas expressões podem ser usadas para a parte da hora da string de formatação:

Expressão	Output
h	a hora sem zero à esquerda (0 a 23 ou 1 a 12 se for exibido AM/PM)
hh	a hora com um zero à esquerda (00 a 23 ou 01 a 12 se for exibido AM/PM)
H	a hora sem zero à esquerda (0 a 23, mesmo com exibição AM/PM)
HH	a hora com um zero à esquerda (00 a 23, mesmo com exibição AM/PM)
m	o minuto sem zero à esquerda (0 a 59)
mm	o minuto com um zero à esquerda (00 a 59)
s	o segundo sem zero à esquerda (0 a 59)
ss	o segundo com um zero à esquerda (00 a 59)
z	os milissegundos sem zeros à direita (0 a 999)
zzz	os milissegundos com zeros à direita (000 a 999)
AP or A	Iinterprete como uma hora AM/PM. *AP* precisa ser ambos 'AM' ou 'PM'.
ap or a	Interprete como uma hora AM/PM. *ap* precisa ser ambos 'am' ou 'pm'.

- **language** - lingua (caixa baixa, duas ou três letras, código de linguagem ISO 639) usada para formatar a data em uma string customizada. Por padrão a localização do usuário atual do QGIS é utilizada.

Exemplos

- `format_date('2012-05-15','dd.MM.yyyy')` → '15.05.2012'
- `format_date('2012-05-15','d MMMM yyyy','fr')` → '15 mai 2012'
- `format_date('2012-05-15','dddd')` → 'Terça-feira', se a localização é uma variante do Português.
- `format_date('2012-05-15 13:54:20','dd.MM.yy')` → '15.05.12'
- `format_date('13:54:20','hh:mm AP')` → '01:54 PM'

hour

Extrai a parte das horas de uma daterime ou hora, ou o número de horas de um intervalo.

Variação de tempo

Extrai a parte das horas de uma daterime ou hora.

Sintaxe	hour(datetime)
Argumentos	• **datetime** - um valor de hora ou datetime
Exemplos	• `hour( to_datetime('2012-07-22 13:24:57') )` → 13

Variante Intervalo

Calcula a duração em horas de um intervalo.

Sintaxe	hour(interval)
Argumentos	• **interval** - valor do intervalo para retornar o número de horas
Exemplos	• `hour(to_interval('3 hours'))` → 3 • `hour(age('2012-07-22T13:00:00','2012-07-22T10:00:00'))` → 3 • `hour(age('2012-01-01','2010-01-01'))` → 17520

make_date

Cria um valor de data a partir de números de ano, mês e dia.

Sintaxe	make_date(year, month, day)
Argumentos	• **year** - Número do ano. Os anos de 1 a 99 são interpretados como estão. O ano 0 é inválido. • **month** - Número do mês, onde 1=Janeiro • **day** - Número do dia, começando com 1 para o primeiro dia do mês
Exemplos	• `make_date(2020,5,4)` → date value 2020-05-04

make_datetime

Cria um valor de data e hora a partir de números de ano, mês, dia, hora, minuto e segundo.

Sintaxe	make_datetime(year, month, day, hour, minute, second)
Argumentos	• **year** - Número do ano. Os anos de 1 a 99 são interpretados como estão. O ano 0 é inválido. • **month** - Número do mês, onde 1=Janeiro • **day** - Número do dia, começando com 1 para o primeiro dia do mês • **hour** - Número da hora • **minute** - Minutos • **second** - Segundos (valores fracionários incluem milissegundos)
Exemplos	• `make_datetime(2020,5,4,13,45,30.5)` → datetime value 2020-05-04 13:45:30.500

make_interval

Cria um intervalo de valores para os valores de ano, mês, semanas, dias, horas, minutos e segundos.

Sintaxe	make_interval([years=0], [months=0], [weeks=0], [days=0], [hours=0], [minutes=0], [seconds=0]) [] indica argumentos opcionais
Argumentos	• **years** - Número de anos (supõe uma duração de ano de 365,25 dias). • **months** - Número de meses (considera uma duração de mês de 30 dias) • **weeks** - Número de semanas • **days** - Número de dias • **hours** - Número de horas • **minutes** - Número de minutos • **seconds** - Número de segundos
Exemplos	• `make_interval(hours:=3)` → interval: 3 hours • `make_interval(days:=2, hours:=3)` → interval: 2.125 days • `make_interval(minutes:=0.5, seconds:=5)` → interval: 35 seconds

make_time

Cria um valor de tempo a partir de números de hora, minuto e segundo.

Sintaxe	make_time(hour, minute, second)
Argumentos	• **hour** - Número da hora • **minute** - Minutos • **second** - Segundos (valores fracionários incluem milissegundos)
Exemplos	• `make_time(13,45,30.5)` → time value 13:45:30.500

minute

Extrai a parte de minutos de um datetime ou hora, ou o número de minutos de um intevalor.

Variação de tempo

Extrai a parte dos minutos de uma hora ou datetime.

Sintaxe	minute(datetime)
Argumentos	• **datetime** - um valor de hora ou datetime
Exemplos	• `minute( to_datetime('2012-07-22 13:24:57') )` → 24

Variante Intervalo

Calcula a extensão em minutos de um intervalo.

Sintaxe	minute(interval)
Argumentos	• **interval** - valor do intervalo para retornar o número de minutos
Exemplos	• `minute(to_interval('3 minutes'))` → 3 • `minute(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 20 • `minute(age('2012-01-01','2010-01-01'))` → 1051200

month

Extrai a parte do mês de uma data ou o número de meses de um intervalo.

Variante Data

Extrai a parte de mês de uma data ou datetime.

Sintaxe	month(date)
Argumentos	• **date** - um valor do tipo data ou datetime
Exemplos	• `month('2012-05-12')` → 05

Variante Intervalo

Calcula a duração em meses de um intervalo.

Sintaxe	month(interval)
Argumentos	• **interval** - valor do intervalo para retornar o número de meses
Exemplos	• `month(to_interval('3 months'))` → 3 • `month(age('2012-01-01','2010-01-01'))` → 4.03333

now

Retorna a data e hora atuais. A função é estática e retornará resultados consistentes durante a avaliação. A hora retornada é a hora em que a expressão é preparada.

Sintaxe	now()
Exemplos	• `now()` → 2012-07-22T13:24:57

second

Extrai a parte dos segundos de um datetime ou hora, ou o número de segundos de um intervalo.

Variação de tempo

Extrai a parte do segundos de uma hora ou datetime.

Sintaxe	second(datetime)
Argumentos	• **datetime** - um valor de hora ou datetime
Exemplos	• `second( to_datetime('2012-07-22 13:24:57') )` → 57

Variante Intervalo

Calcula a duração em segundos de um intervalo.

Sintaxe	second(interval)
Argumentos	• **interval** - valor do intervalo para retornar o número de segundos
Exemplos	• `second(to_interval('3 minutes'))` → 180 • `second(age('2012-07-22T00:20:00','2012-07-22T00:00:00'))` → 1200 • `second(age('2012-01-01','2010-01-01'))` → 63072000

to_date

Converte uma string em um objeto de data. Uma string de formatação pode ser fornecida para analisar a string; veja QDate::fromString ou a documentação da função format_date para documentação adicional da fomatação. Por padrão o locale atual do usuário do QGIS será utilizado.

Sintaxe	to_date(string, [format], [language]) [] indica argumentos opcionais
Argumentos	• **string** - string representando um valor de data • **format** - formato usado para converter a cadeia em uma data • **language** - idioma (minúsculo, duas ou três letras, código de idioma ISO 639) usado para converter a cadeia em uma data. Por padrão, a localidade atual do usuário QGIS é usada.
Exemplos	• `to_date('2012-05-04')` → 2012-05-04 • `to_date('June 29, 2019','MMMM d, yyyy')` → 2019-06-29, se o locale atual utilizar o nome 'June' para o sexto mês, caso contrário irá ocorrer um erro • `to_date('29 juin, 2019','d MMMM, yyyy','fr')` → 2019-06-29

Further reading: *format_date*

to_datetime

Converte uma string em um objeto datetime. Uma string opcional pode ser fornecido para avaliar a string; veja QDate::fromString, QTime::fromString ou a documentação da função format_date para documentação adicional do formato. Por padrão a localização atual do usuário do QGIS é utilizada.

Sintaxe	to_datetime(string, [format], [language]) [] indica argumentos opcionais
Argumentos	• **string** - string representando o valor datetime • **format** - formato utilizado para converter a string em datetime • **language** - lingua (caixa baixa, duas ou três letras, código de lingua ISO 639) usada para converter a string em datetime. Por padrão a localização atual do usuário do QGIS é utilizada.
Exemplos	• `to_datetime('2012-05-04 12:50:00')` → 2012-05-04T12:50:00 • `to_datetime('June 29, 2019 @ 12:34','MMMM d, yyyy @ HH:mm')` → 2019-06-29T12:34, se a localização atual utilizar o nome 'June' para o sexto mês, caso contrário acontece um erro • `to_datetime('29 juin, 2019 @ 12:34','d MMMM, yyyy @ HH:mm', 'fr')` → 2019-06-29T12:34

Further reading: *format_date*

to_interval

Converte uma string para o tipo intervalo. Pode ser utilizado dias, horas, mês, etc de uma data.

Sintaxe	to_interval(string)
Argumentos	• **string** - um string representando um intervalo. Formatos permitidos são {n} days {n} hours {n} months.
Exemplos	• `to_interval('1 day 2 hours')` → interval: 1.08333 dias • `to_interval( '0.5 hours' )` → interval: 30 minutos • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

to_time

Converte uma string em um objeto time. Uma string de formato opcional pode ser fornecida para analisar a string; veja QTime::fromString para documentação adicional sobre o formato.

Sintaxe	to_time(string, [format], [language]) [] indica argumentos opcionais
Argumentos	• **string** - string representando um valor time • **format** - formato utilizado para converter a string em time • **language** - lingua (caixa baixa, duas ou três letras, codigo de linguagem ISO 639) usado para converter a string em time
Exemplos	• `to_time('12:30:01')` → 12:30:01 • `to_time('12:34','HH:mm')` → 12:34:00 • `to_time('12:34','HH:mm','fr')` → 12:34:00

Further reading: *format_date*

week

Extrai o número da semana de uma data ou o número de semanas de um intervalo.

Variante Data

Extrai o número da semana de uma data ou datetime.

Sintaxe	week(date)
Argumentos	• **date** - um valor do tipo data ou datetime
Exemplos	• `week('2012-05-12')` → 19

Variante Intervalo

Calcula a duração em semanas de um intervalo.

Sintaxe	week(interval)
Argumentos	• **interval** - valor do intervalo para retornar o número de meses
Exemplos	• `week(to_interval('3 weeks'))` → 3 • `week(age('2012-01-01','2010-01-01'))` → 104.285

year

Extrai a parte do ano de uma data ou o número de anos de um intervalo.

Variante Data

Extrai a parte do ano de uma data ou de um datetime.

Sintaxe	year(date)
Argumentos	• **date** - um valor do tipo data ou datetime
Exemplos	• `year('2012-05-12')` → 2012

Variante Intervalo

Calcula a duração em anos de um intervalo.

Sintaxe	year(interval)
Argumentos	• **interval** - valor do intervalo para retornar o número de anos
Exemplos	• `year(to_interval('3 years'))` → 3 • `year(age('2012-01-01','2010-01-01'))` → 1.9986

Alguns exemplos:

Além dessas funções, subtrair datas, datas ou horas usando o operador − (menos) retornará um intervalo.

Adicionar ou subtrair um intervalo a datas, data/hora ou horas, usando os operadores + (mais) e − (menos), retornará uma data/hora.

- Obtenha o número de dias até o lançamento do QGIS 3.0:

```
to_date('2017-09-29') - to_date(now())
-- Returns <interval: 203 days>
```

- O mesmo com o tempo:

```
to_datetime('2017-09-29 12:00:00') - now()
-- Returns <interval: 202.49 days>
```

- Obtenha a data/hora de 100 dias a partir de agora:

```
now() + to_interval('100 days')
-- Returns <datetime: 2017-06-18 01:00:00>
```

9.2.9 Campos e Valores

Contém uma lista de campos da camada ativa e valores especiais. Lista de campos inclui os que estão armazenados no dataset, *virtuais* e *auxiliares* bem como os *unidos*.

Clique duas vezes no nome de um campo para adicioná-lo à sua expressão. Você também pode digitar o nome do campo (de preferência entre aspas duplas) ou seu *alias*.

Para recuperar os valores dos campos a serem usados em uma expressão, selecione o campo apropriado e, no widget mostrado, escolha entre *10 Amostras* e *Todos os Únicos*. Os valores solicitados são exibidos e você pode usar a caixa *Pesquisar* na parte superior da lista para filtrar o resultado. Valores de amostra também podem ser acessados clicando com o botão direito do mouse em um campo.

Para adicionar um valor à expressão que você está escrevendo, clique duas vezes nela na lista. Se o valor for de um tipo de string, deve ser simples entre aspas, caso contrário, nenhuma citação será necessária.

NULL

Equivale a um valor NULO.

Sintaxe	NULL
Exemplos	• `NULO` → um valor NULO

Nota

Para testar NULO, use uma expressão *IS NULL* ou *IS NOT NULL*.

9.2.10 Funções de Arquivos e Caminhos

Este grupo contém funções que manipulam nomes de arquivos e caminhos.

base_file_name

Retorna o nome base do arquivo sem o diretório ou sufixo do arquivo.

Sintaxe	base_file_name(path)
Argumentos	• **path** - um caminho de arquivo ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada.
Exemplos	• `base_file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries'

exif

Recupera valores de tag exif de um arquivo de imagem.

Sintaxe	exif(path, [tag]) [] indica argumentos opcionais
Argumentos	• **path** - Um caminho de arquivo de imagem ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada. • **tag** - A tag a ser retornada. se vazio, um mapa com todos os valores será retornado.
Exemplos	• `exif('/my/photo.jpg','Exif.Image.Orientation')` → 0

file_exists

Retorna TRUE se o caminho do arquivo existir.

Sintaxe	file_exists(path)
Argumentos	• **path** - um caminho de arquivo ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada.
Exemplos	• `file_exists('/home/qgis/data/country_boundaries.shp')` → TRUE

file_name

Retorna o nome de um arquivo (incluindo a extensão do arquivo), excluindo o diretório.

Sintaxe	file_name(path)
Argumentos	• **path** - um caminho de arquivo ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada.
Exemplos	• `file_name('/home/qgis/data/country_boundaries.shp')` → 'country_boundaries.shp'

file_path

Retorna o componente de diretório de um caminho de arquivo. Isso não inclui o nome do arquivo.

Sintaxe	file_path(path)
Argumentos	• **path** - um caminho de arquivo ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada.
Exemplos	• `file_path('/home/qgis/data/country_boundaries.shp')` → '/home/qgis/data'

file_size

Retorna o tamanho (em bytes) de um arquivo.

Sintaxe	file_size(path)
Argumentos	• **path** - um caminho de arquivo ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada.
Exemplos	• `file_size('/home/qgis/data/country_boundaries.geojson')` → 5674

file_suffix

Retorna o sufixo de um arquivo (extensão) de um caminho de arquivo.

Sintaxe	file_suffix(path)
Argumentos	• **path** - um caminho de arquivo ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada.
Exemplos	• `file_suffix('/home/qgis/data/country_boundaries.shp')` → 'shp'

is_directory

Retorna VERDADEIRO se um caminho corresponder a um diretório.

Sintaxe	is_directory(path)
Argumentos	• **path** - um caminho de arquivo ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada.
Exemplos	• `is_directory('/home/qgis/data/country_boundaries.shp')` → FALSE • `is_directory('/home/qgis/data/')` → TRUE

is_file

Retorna VERDADEIRO se um caminho corresponder a um arquivo.

Sintaxe	is_file(path)
Argumentos	• **path** - um caminho de arquivo ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada.
Exemplos	• `is_file('/home/qgis/data/country_boundaries.shp')` → TRUE • `is_file('/home/qgis/data/')` → FALSE

9.2.11 Funções de Formulário

Este grupo contém funções que operam exclusivamente sobre os atributos de um contexto. Por exemplo, na configuração dos *widgets de campos*.

current_parent_value

Utilizável apenas em um contexto de formulário incorporado, esta função retorna o valor atual e não salvo de um campo no formulário pai que está sendo editado no momento. Isso será diferente dos valores de atributo reais da feição pai para feições que estão sendo editadas no momento ou que ainda não foram adicionadas a uma camada pai. Quando usada em um widget de filtro por expressão de relação de valor, essa função deve ser agrupada em um 'coalesce()' que pode recuperar o recurso pai real da camada quando o formulário não é usado em um contexto incorporado.

Sintaxe	current_parent_value(field_name)
Argumentos	• **field_name** - um nome de campo no fomulátio pai atual
Exemplos	• `current_parent_value( 'FIELD_NAME' )` → O valot atual no campo 'FIELD_NAME' no formulário pai.

current_value

Retorna o valor atual não salvo de um campo no formulário ou linha da tabela que está sendo editada no momento. Isso será diferente dos valores de atributo reais da feição para feições que estão sendo editadas no momento ou que ainda não foram adicionadas a uma camada.

Sintaxe	current_value(field_name)
Argumentos	• **field_name** - um nome de campo no formulário pai ou linha da tabela.
Exemplos	• `current_value( 'FIELD_NAME' )` → O valor atual do campo 'FIELD_NAME'.

9.2.12 Funções de Combinação Fuzzy

Este grupo contém funções para comaparações fuzzy entre valores.

hamming_distance

Retorna a distância de Hamming entre duas strings. Isso equivale ao número de caracteres em posições correspondentes nas strings de entrada onde os caracteres são diferentes. As cadeias de caracteres de entrada devem ter o mesmo comprimento e a comparação diferencia maiúsculas de minúsculas.

Sintaxe	hamming_distance(string1, string2)
Argumentos	• **string1** - a string • **string2** - a string
Exemplos	• `hamming_distance('abc','xec')` → 2 • `hamming_distance('abc','ABc')` → 2 • `hamming_distance(upper('abc'),upper('ABC'))` → 0 • `hamming_distance('abc','abcd')` → NULL

levenshtein

Retorna a distância de edição Levenshtein entre duas cadeias. Isso equivale ao número mínimo de edições de caracteres (inserções, exclusões ou substituições) necessárias para alterar uma cadeia para outra.

A distância Levenshtein é uma medida da semelhança entre duas astrings. Distâncias menores significam que as string são mais semelhantes, e distâncias maiores indicam mais strings diferentes. A distância diferencia maiúsculas de minúsculas.

Sintaxe	levenshtein(string1, string2)
Argumentos	• **string1** - a string • **string2** - a string
Exemplos	• `levenshtein('kittens','mitten')` → 2 • `levenshtein('Kitten','kitten')` → 1 • `levenshtein(upper('Kitten'),upper('kitten'))` → 0

longest_common_substring

Retorna a sub string comum mais longa entre duas strings. Essa sub string é a cadeia mais longa que é uma sub string das duas strings de entrada. Por exemplo, a sub string comum mais longa de “ABABC” e “BABCA” é “BABC”. A substring faz distinção entre maiúsculas e minúsculas.

Sintaxe	longest_common_substring(string1, string2)
Argumentos	• **string1** - a string • **string2** - a string
Exemplos	• `longest_common_substring('ABABC','BABCA')` → ‘BABC’ • `longest_common_substring('abcDeF','abcdef')` → ‘abc’ • `longest_common_substring(upper('abcDeF'),upper('abcdex'))` → ‘ABCDE’

soundex

Retorna a representação Soundex de uma string. Soundex é um algoritmo de correspondência fonética, portanto, strings com sons semelhantes devem ser representadas pelo mesmo código Soundex.

Sintaxe	soundex(string)
Argumentos	• **string** - a string
Exemplos	• `soundex('robert')` → 'R163' • `soundex('rupert')` → 'R163' • `soundex('rubin')` → 'R150'

9.2.13 Funções Gerais

Este grupo contém funções gerais variadas.

env

Obtém uma variável de ambiente e retorna seu conteúdo como uma string. Se a variável não for encontrada, NULL será retornado. Isso é útil para injetar configurações específicas do sistema, como letras de unidade ou prefixos de caminho. A definição de variáveis de ambiente depende do sistema operacional, verifique com o administrador do sistema ou a documentação do sistema operacional como isso pode ser definido.

Sintaxe	env(name)
Argumentos	• **name** - O nome da variável de ambiente que deve ser recuperada.
Exemplos	• `env( 'LANG' )` → 'en_US.UTF-8' • `env( 'MY_OWN_PREFIX_VAR' )` → 'Z:' • `env( 'I_DO_NOT_EXIST' )` → NULL

eval

Avalia uma expressão que é passada em uma string. Útil para expandir parâmetros dinâmicos passados como campos ou variáveis de contexto.

Sintaxe	eval(expression)
Argumentos	• **expression** - uma string de expressão
Exemplos	• `eval('\'nice\'')` → 'nice' • `eval(@expression_var)` → [qualquer que seja o resultado da avaliação de @expression_var…]

eval_template

Avalia um modelo que é passado em uma string. Útil para expandir parâmetros dinâmicos passados como variáveis de contexto ou campos.

Sintaxe	eval_template(template)
Argumentos	• **template** - uma string modelo
Exemplos	• `eval_template('QGIS [% upper(\'rocks\') %]')` → QGIS ROCKS

is_layer_visible

Retorna VERDADEIRO se a camada especificada estiver visível.

Sintaxe	is_layer_visible(layer)
Argumentos	• **layer** - uma string, representando um nome de camada ou um ID de camada
Exemplos	• `is_layer_visible('baseraster')` → TRUE

mime_type

Retorna o mime type dos dados binários.

Sintaxe	mime_type(bytes)
Argumentos	• **bytes** - os dados binários
Exemplos	• `mime_type('<html><body></body></html>')` → text/html • `mime_type(from_base64('R0lGODlhAQABAAAAACH5BAEKAAEALAAAAAABAAEAAAIAOw==` → image/gif

var

Retorna o valor armazenado em uma variável especificada.

Sintaxe	var(name)
Argumentos	• **name** - um nome de variável
Exemplos	• `var('qgis_version')` → '2.12'

Leitura adicional: Lista de *variáveis* padrão

with_variable

Esta função define uma variável para qualquer código de expressão que será fornecido como 3º argumento. Isso só é útil para expressões complicadas, onde o mesmo valor calculado precisa ser usado em lugares diferentes.

Sintaxe	with_variable(name, value, expression)
Argumentos	• **name** - o nome da variável a ser definida • **value** - o valor a ser definido • **expression** - a expressão para a qual a variável estará disponível
Exemplos	• `with_variable('my_sum', 1 + 2 + 3, @my_sum * 2 + @my_sum * 5)` → 42

9.2.14 Funções de Geometria

Este grupo contém funções que operam em objetos de geometria (por exemplo, buffer, transform, $area).

affine_transform

Retorna a geometria após uma transformação afim. Os cálculos estão no Sistema de Referência Espacial desta geometria. As operações são realizadas em escala, rotação, ordem de translação. Se houver um deslocamento Z ou M, mas a coordenada não estiver presente na geometria, ela será adicionada.

Sintaxe	affine_transform(geometry, delta_x, delta_y, rotation_z, scale_x, scale_y, [delta_z=0], [delta_m=0], [scale_z=1], [scale_m=1]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **delta_x** - tradução do eixo x • **delta_y** - tradução do eixo y • **rotation_z** - rotação em torno do eixo z em graus no sentido anti-horário • **scale_x** - fator de escala do eixo x • **scale_y** - fator de escala do eixo y • **delta_z** - tradução do eixo z • **delta_m** - tradução do eixo m • **scale_z** - fator de escala do eixo z • **scale_m** - fator de escala do eixo m
Exemplos	• `geom_to_wkt(affine_transform(geom_from_wkt('LINESTRING(1 1, 2 2)'), 2, 2, 0, 1, 1))` → 'LineString (3 3, 4 4)' • `geom_to_wkt(affine_transform(geom_from_wkt('POLYGON((0 0, 0 3, 2 2, 0 0))'), 0, 0, -90, 1, 2))` → 'Polygon ((0 0, 6 0, 4 -2, 0 0))' • `geom_to_wkt(affine_transform(geom_from_wkt('POINT(3 1)'), 0, 0, 0, 1, 1, 5, 0))` → 'PointZ (3 1 5)'

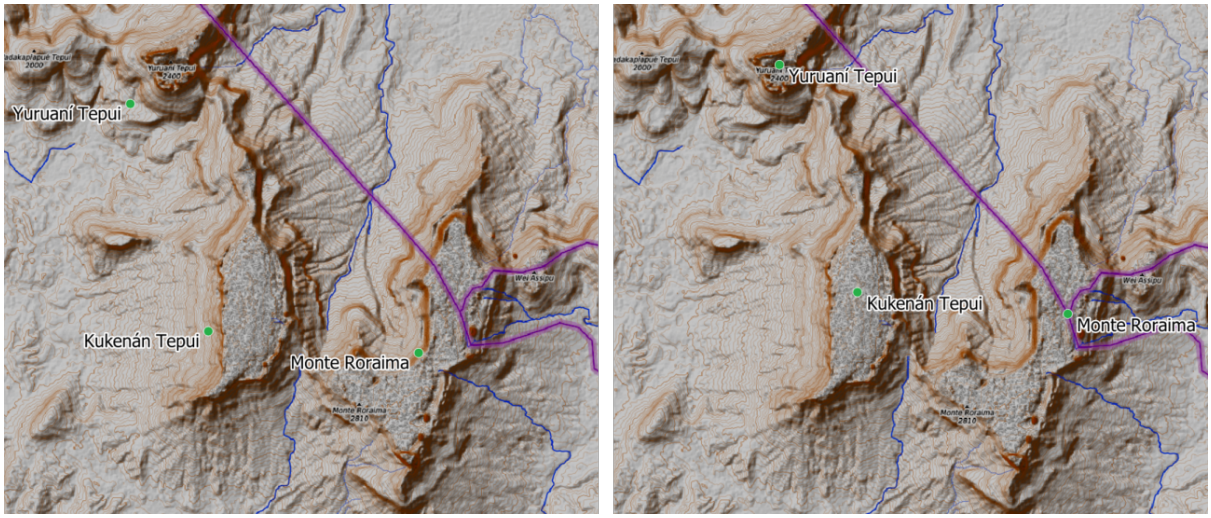


Figura9.4: Ponto da camada de vetor (pontos verdes) antes (esquerda) e depois (direita) de uma transformação afim.

angle_at_vertex

Retorna o ângulo da bissetriz (ângulo médio) para a geometria de um vértice especificado em uma geometria de linha. Os ângulos estão em graus no sentido horário a partir do norte.

Sintaxe	angle_at_vertex(geometry, vertex)
Argumentos	• **geometry** - Uma geometria de linha • **vertex** - índice do vértice, a partir de 0; se o valor for negativo, o índice do vértice selecionado será sua contagem total menos o valor absoluto
Exemplos	• `angle_at_vertex(geometry:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 45.0

apply_dash_pattern

Aplica um padrão de traço a uma geometria, retornando uma geometria MultiLineString que é a geometria de entrada traçada ao longo de cada linha/anel com o padrão especificado.

Sintaxe	apply_dash_pattern(geometry, pattern, [start_rule=no_rule], [end_rule=no_rule], [adjustment=both], [pattern_offset=0]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria (aceita (multi)linhas ou (multi)polígonos). • **pattern** - padrão de traço, como uma matriz de números representando comprimentos de traço e intervalo. Deve conter um número par de elementos. • **start_rule** - regra opcional para restringir o início do padrão. Os valores válidos são 'no_rule', 'full_dash', 'half_dash', 'full_gap', 'half_gap'. • **end_rule** - regra opcional para restringir o fim do padrão. Os valores válidos são 'no_rule', 'full_dash', 'half_dash', 'full_gap', 'half_gap'. • **adjustment** - regra opcional para especificar quais partes dos padrões são ajustadas para caber nas regras de padrão desejadas. Os valores válidos são 'both', 'dash', 'gap'. • **pattern_offset** - Distância opcional especificando uma distância específica ao longo do padrão para começar.
Exemplos	• `geom_to_wkt(apply_dash_pattern(geom_from_wkt('LINESTRING(1 1, 10 1)'), array(3, 1)))` → MultiLineString ((1 1, 4 1),(5 1, 8 1),(9 1, 10 1, 10 1)) • `geom_to_wkt(apply_dash_pattern(geom_from_wkt('LINESTRING(1 1, 10 1)'), array(3, 1), start_rule:='half_dash'))` → MultiLineString ((1 1, 2.5 1),(3.5 1, 6.5 1),(7.5 1, 10 1, 10 1))

$area

Retorna a área do recurso atual. A área calculada por esta função respeita tanto a configuração do elipsóide do projeto atual quanto as configurações da unidade de área. Por exemplo, se um elipsóide foi definido para o projeto, a área calculada será elipsoidal, e se nenhum elipsóide for definido, a área calculada será planimétrica.

Sintaxe	$area
Exemplos	• `$area` → 42

area

Retorna a área de um objeto de geometria polígono. Os cálculos são sempre planimétricos no Sistema de Referência Espacial (SRS) dessa geometria, e as unidades da área retornada corresponderão às unidades do SRS. Isso difere dos cálculos realizados pela função $area, que realizará cálculos elipsoidais com base nas configurações de elipsóide e unidade de área do projeto.

Sintaxe	area(geometry)
Argumentos	• **geometry** - objeto de geometria de polígono
Exemplos	• `area(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 8.0

azimuth

Retorna o azimute baseado no norte como o ângulo em radianos medido no sentido horário da vertical em point_a a point_b.

Sintaxe	azimuth(point_a, point_b)
Argumentos	• **point_a** - geometria de ponto • **point_b** - geometria de ponto
Exemplos	• `degrees( azimuth( make_point(25, 45), make_point(75, 100) ) )` → 42.273689 • `degrees( azimuth( make_point(75, 100), make_point(25,45) ) )` → 222.273689

bearing

Returns the north-based bearing as the angle in radians measured clockwise on the ellipsoid from the vertical on point_a to point_b.

Sintaxe	bearing(point_a, point_b, [source_crs], [ellipsoid]) [] indica argumentos opcionais
Argumentos	• **point_a** - geometria de ponto • **point_b** - geometria de ponto • **source_crs** - an optional string or CRS object representing the source CRS of the points. By default the current layer's CRS is used. • **ellipsoid** - an optional string representing the acronym or the authority ID (e.g., 'EPSG:7030') of the ellipsoid on which the bearing should be measured. By default the current project's ellipsoid setting is used.
Exemplos	• `degrees( bearing( make_point(16198544, -4534850), make_point(18736872, -1877769), 'EPSG:3857', 'EPSG:7030') )` → 49.980071 • `degrees( bearing( make_point(18736872, -1877769), make_point(16198544, -4534850), 'EPSG:3857', 'WGS84') )` → 219.282386

boundary

Retorna o fechamento do limite combinatório da geometria (ou seja, o limite topológico da geometria). Por exemplo, uma geometria de polígono terá um limite que consiste nas cadeias de linhas para cada anel no polígono. Alguns tipos de geometria não têm um limite definido, por exemplo, pontos ou coleções de geometria, e retornarão NULO.

Sintaxe	boundary(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `geom_to_wkt(boundary(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → 'LineString(1 1,0 0,-1 1,1 1)' • `geom_to_wkt(boundary(geom_from_wkt('LineString(1 1,0 0,-1 1)')))` → 'MultiPoint ((1 1),(-1 1))'

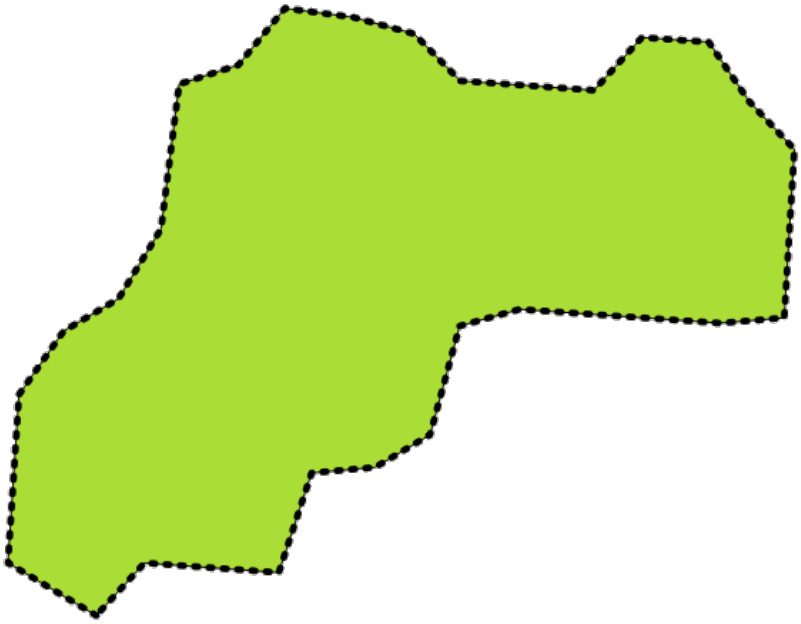


Figura9.5: Boundary (black dashed line) of the source polygon layer

Leitura adicional: algoritmo *Limite*

bounds

Retorna uma geometria que representa a caixa delimitadora de uma geometria de entrada. Os cálculos estão no Sistema de Referência Espacial desta geometria.

Sintaxe	bounds(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `bounds(@geometry)` → bounding box of the current feature's geometry • `geom_to_wkt(bounds(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → 'Polygon ((-1 0, 1 0, 1 1, -1 1, -1 0))'

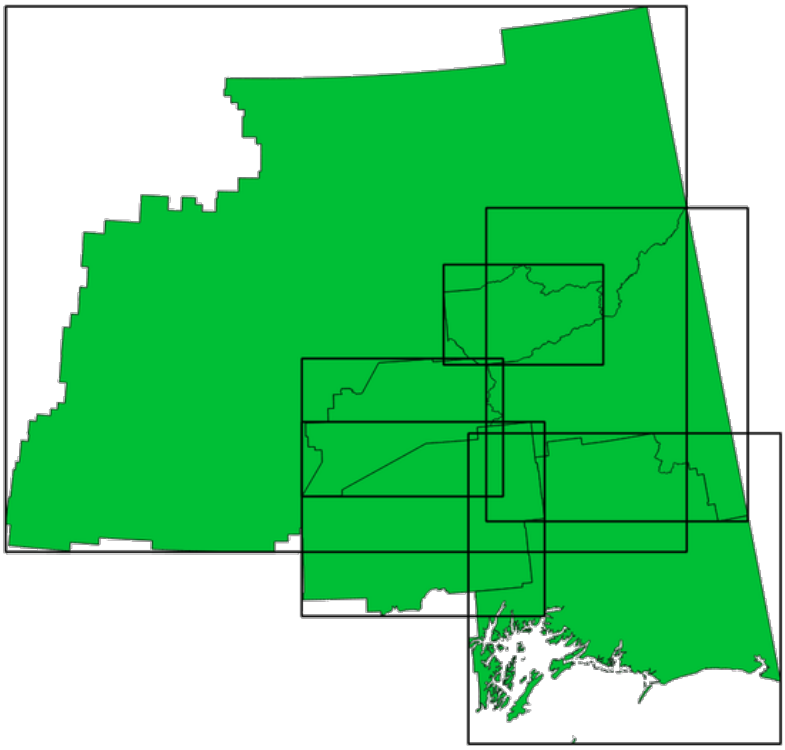


Figura9.6: As linhas pretas representam as caixas delimitadoras de cada feição do polígono

Leitura adicional: algoritmo *Caixas delimitadoras.*

bounds_height

Retorna a altura do rentângulo envolvente de uma geometria. O cálculos são no Sistema de Referencia Espacial da geometria.

Sintaxe	bounds_height(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `bounds_height(@geometry)` → height of bounding box of the current feature's geometry • `bounds_height(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 1

bounds_width

Retorna a largura do retângulo envolvente de uma geometria. Os cálculos estão no Sistema de Referência Espacial desta geometria.

Sintaxe	bounds_width(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `bounds_width(@geometry)` → width of bounding box of the current feature's geometry • `bounds_width(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))'))` → 2

buffer

Retorna uma geometria que representa todos os pontos cuja distância dessa geometria é menor ou igual à distância. Os cálculos estão no Sistema de Referência Espacial desta geometria.

Sintaxe	buffer(geometry, distance, [segments=8], [cap='round'], [join='round'], [miter_limit=2]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **distance** - distância do buffer em unidades de camada • **segments** - número de segmentos a serem usados para representar um quarto de círculo quando um estilo de junção arredondada é usado. Um número maior resulta em um buffer mais suave com mais nós. • **cap** - estilo do end cap style do buffer. Valores válidos são 'round', 'flat' ou 'square' • **join** - estilo da junção do buffer. Valores válidos são 'round', 'bevel' ou 'miter'. • **miter_limit** - distância limite do miter, para ser utilizado quando o estilo de junção for configurado para 'miter'
Exemplos	• `buffer(@geometry, 10.5)` → polygon of the current feature's geometry buffered by 10.5 units

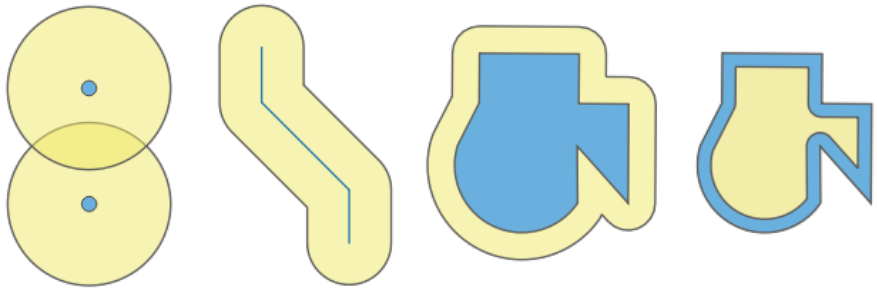


Figura9.7: Buffer (em amarelo) de pontos, linha, polígono com buffer positivo e polígono com buffer negativo

Leitura adicional: algoritmo *Buffer*

buffer_by_m

Cria um buffer ao longo de uma geometria de linha onde o diâmetro do buffer varia de acordo com os valores m nos vértices da linha.

Sintaxe	buffer_by_m(geometry, [segments=8]) [] indica argumentos opcionais
Argumentos	• **geometry** - geometria de entrada. Deve ser uma geometria de (multi)linha com valores m. • **segments** - número de segmentos para aproximar as curvas de um quarto de círculo no buffer.
Exemplos	• `buffer_by_m(geometry:=geom_from_wkt('LINESTRINGM(1 2 0.5, 4 2 0.2)'),segments:=8)` → Um buffer de largura variável começando com um diâmetro de 0,5 e terminando com um diâmetro de 0,2 ao longo da geometria da linha

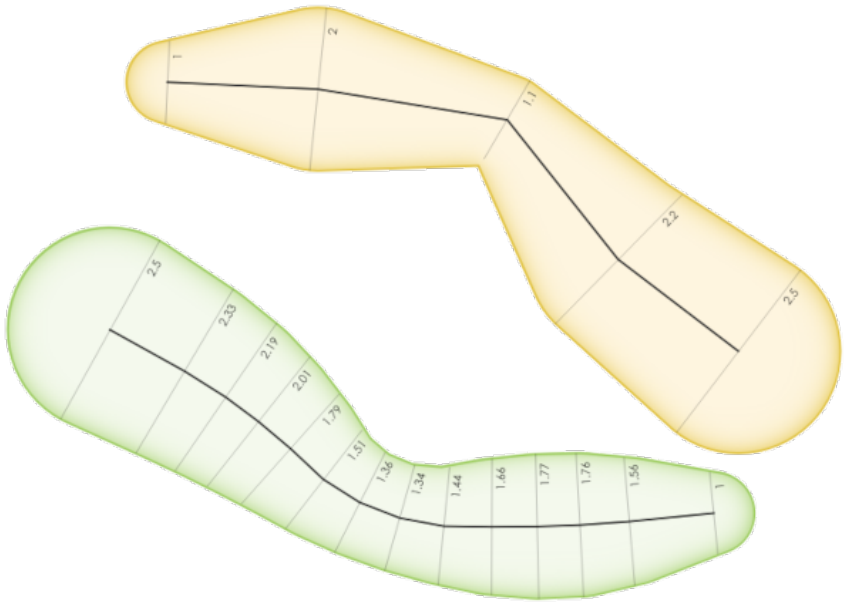


Figura9.8: Buffer nas feições de linha usando o valor m nos vértices

Leitura adicional: algoritmo *Buffer de largura variável (por valor M)*

centroid

Retorna o centro geométrico de uma geometria.

Sintaxe	centroid(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `centroid(@geometry)` → a point geometry

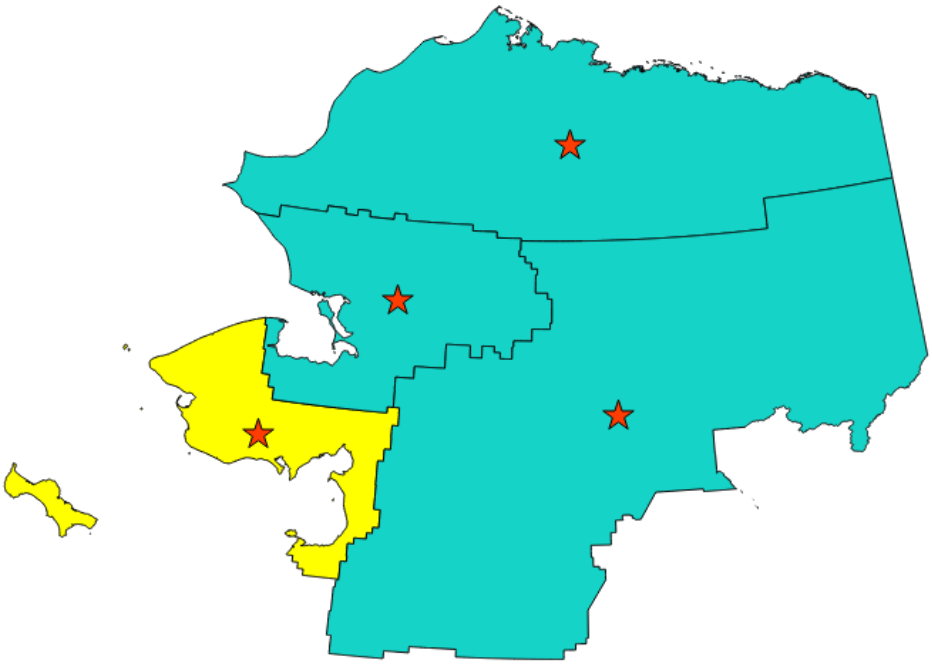


Figura9.9: As estrelas vermelhas representam os centróides das feições da camada de entrada.

Leitura adicional: algoritmo *Centroides*

close_line

Retorna uma linha fechada da linha de entrada anexando o primeiro ponto ao final da linha, se ainda não estiver fechada. Se a geometria não for uma sequência de linhas ou sequência de várias linhas, o resultado será NULL.

Sintaxe	close_line(geometry)
Argumentos	• **geometry** - uma geometria de linha
Exemplos	• `geom_to_wkt(close_line(geom_from_wkt('LINESTRING(0 0, 1 0, 1 1)')))` → 'LineString (0 0, 1 0, 1 1, 0 0)' • `geom_to_wkt(close_line(geom_from_wkt('LINESTRING(0 0, 1 0, 1 1, 0 0)')))` → 'LineString (0 0, 1 0, 1 1, 0 0)'

closest_point

Retorna o ponto em geometria1 que está mais próximo de geometria2.

Sintaxe	closest_point(geometry1, geometry2)
Argumentos	• **geometry1** - geometria para encontrar o ponto mais próximo em • **geometry2** - geometria para encontrar o ponto mais próximo de
Exemplos	• `geom_to_wkt(closest_point(geom_from_wkt('LINESTRING (20 80, 98 190, 110 180, 50 75 )'),geom_from_wkt('POINT(100 100)')))` → 'Point(73.0769 115.384)'

collect_geometries

Coleta uma série de geometrias em um objeto de geometria multi parte.

Lista de variantes de argumentos

As partes geométricas são especificadas como argumentos separados para a função.

Sintaxe	collect_geometries(geometry1, geometry2, …)
Argumentos	• **geometry** - uma geometria
Exemplos	• `geom_to_wkt(collect_geometries(make_point(1,2), make_point(3,4), make_point(5,6)))` → 'MultiPoint ((1 2),(3 4),(5 6))'

Variante array

As partes de geometria são especificadas como um array de partes de geometria.

Sintaxe	collect_geometries(array)
Argumentos	• **array** - array de objetos de geometria
Exemplos	• `geom_to_wkt(collect_geometries(array(make_point(1,2), make_point(3,4), make_point(5,6))))` → 'MultiPoint ((1 2),(3 4),(5 6))'

Leitura adicional: algoritmo *Coletar geometrias*

combine

Retorna a combinação de duas geometrias.

Sintaxe	combine(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `geom_to_wkt( combine( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 2 1)' ) ) )` → 'MULTILINESTRING((4 4, 2 1), (3 3, 4 4), (4 4, 5 5))' • `geom_to_wkt( combine( geom_from_wkt( 'LINESTRING(3 3, 4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 6 6, 2 1)' ) ) )` → 'LINESTRING(3 3, 4 4, 6 6, 2 1)'

concave_hull

Retorna um polígono possivelmente côncavo que contém todos os pontos na geometria

Sintaxe	concave_hull(geometry, target_percent, [allow_holes=False]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **target_percent** - a porcentagem de área do cobertura convexa que a solução tenta se aproximar. Um target_percent de 1 dá o mesmo resultado que a cobertura convexa. Um target_percent entre 0 e 0,99 produz um resultado que deve ter uma área menor que a cobertura convexa. • **allow_holes** - optional argument specifying whether to allow holes within the output geometry. Defaults to FALSE, set to TRUE to allow including holes in the output geometry.
Exemplos	• `geom_to_wkt(concave_hull(geom_from_wkt('MULTILINESTRING((106 164,30 112,74 70,82 112,130 94,130 62,122 40,156 32,162 76,172 88),(132 178,134 148,128 136,96 128,132 108,150 130,170 142,174 110,156 96,158 90,158 88),(22 64,66 28,94 38,94 68,114 76,112 30,132 10,168 18,178 34,186 52,184 74,190 100,190 122,182 148,178 170,176 184,156 164,146 178,132 186,92 182,56 158,36 150,62 150,76 128,88 118))'), 0.99))` → 'Polygon ((30 112, 36 150, 92 182, 132 186, 176 184, 190 122, 190 100, 186 52, 178 34, 168 18, 132 10, 112 30, 66 28, 22 64, 30 112))'

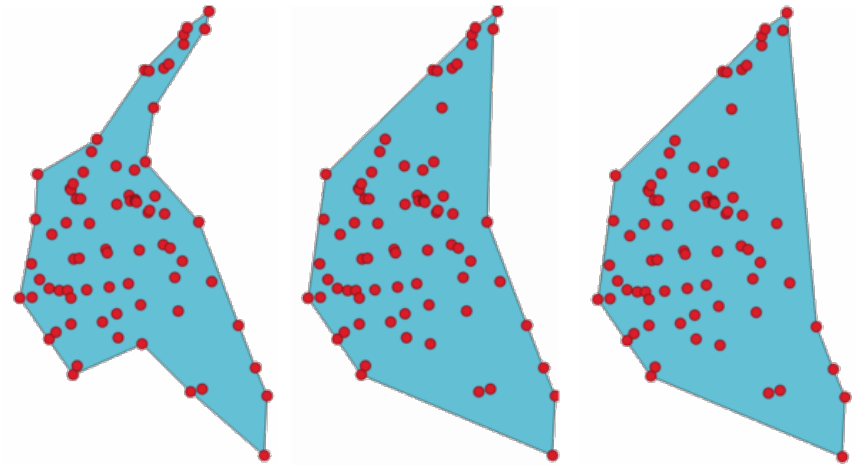


Figura9.10: Concave hulls with increasing target_percent parameter

Further reading: *convex_hull*, *Concave hull* algorithm

contains

Testa se uma geometria contém outra. Retorna TRUE se e somente se nenhum ponto da geometria2 estiver no exterior da geometria1 e pelo menos um ponto do interior da geometria2 estiver no interior da geometria1.

Sintaxe	contains(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `contains( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ), geom_from_wkt( 'POINT(0.5 0.5 )' ) )` → TRUE • `contains( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

Leitura adicional: *overlay_contains*

convex_hull

Retorna o convex hull de uma geometria. Ele representa a geometria convexa mínima que encerra todas as geometrias dentro do conjunto.

Sintaxe	convex_hull(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `geom_to_wkt( convex_hull( geom_from_wkt( 'LINESTRING(3 3, 4 4, 4 10)' ) ) )` → 'POLYGON((3 3, 4 10, 4 4, 3 3))'

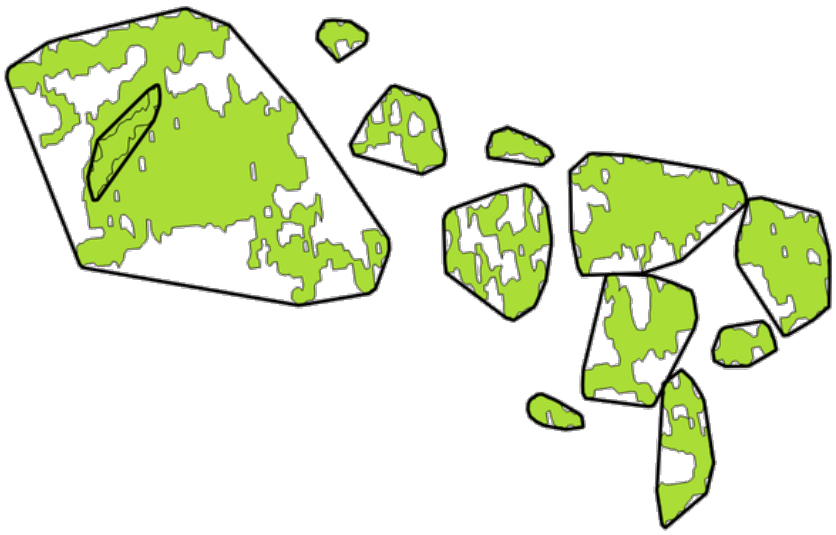


Figura9.11: Black lines identify the convex hull for each feature

Leitura adicional: *concave_hull*, algoritmo *Casco convexo.*

crosses

Testa quando uma geometria cruza a outra. Retorna VERDADEIRO se as geometrias fornecidas possuirem algum, mas não todos, pontos internos em comum.

Sintaxe	crosses(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `crosses( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `crosses( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

Leitura adicional: *overlay_crosses*

densify_by_count

Recebe uma camada com geometria do tipo polígono ou linha e gera uma nova onde as geometrias possuem uma quantidade maior de vertices que a original.

Sintaxe	densify_by_count(geometry, vertices)
Argumentos	• **geometry** - uma geometria (aceita (multi)linhas ou (multi)polígonos). • **vertices** - número de vertices a serem adicionados (por segmento)
Exemplos	• `geom_to_wkt(densify_by_count(geom_from_wkt('LINESTRING(1 1, 10 1)'), 3))` → LineString (1 1, 3.25 1, 5.5 1, 7.75 1, 10 1)

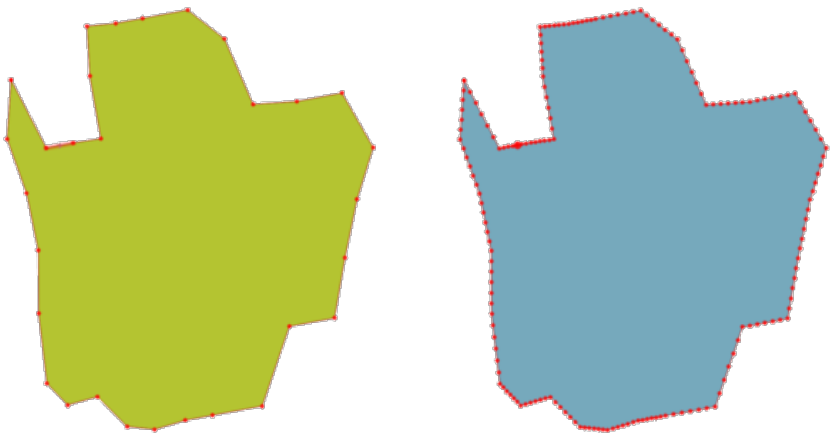


Figura9.12: Os pontos vermelhos mostram os vértices antes e depois da densificação

Leitura adicional: algoritmo *Densificar por contagem.*

densify_by_distance

Pega um camada de polígono ou geometria e gera uma nova na qual as geometrias são densificadas adicionando novos vértices em arestas que têm uma distância máxima da distância de intervalo especificada.

Sintaxe	densify_by_distance(geometry, distance)
Argumentos	• **geometry** - uma geometria (aceita (multi)linhas ou (multi)polígonos). • **distance** - intervalo máximo entre os vértices na geometria de saída
Exemplos	• `geom_to_wkt(densify_by_distance(geom_from_wkt('LINESTRING(1 1, 10 1)'), 4))` → LineString (1 1, 4 1, 7 1, 10 1)

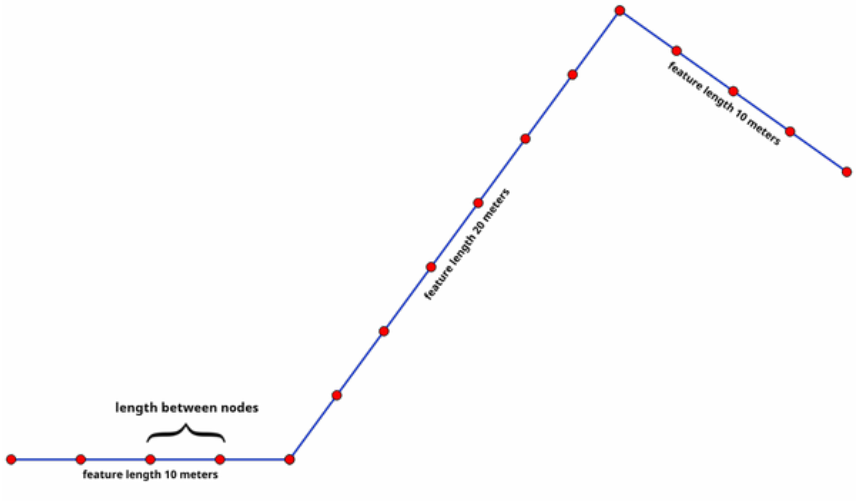


Figura9.13: Geometrias densificadas em um dado intervalo.

Leitura adicional: algoritmo *Densificar por intervalo*

difference

Retorna uma geometria que representa a parte da geometria1 que não faz interseção com geometria2.

Sintaxe	difference(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `geom_to_wkt( difference( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4)' ) ) )` → 'LINESTRING(4 4, 5 5)'

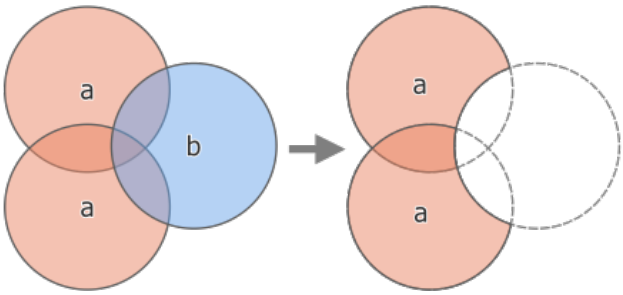


Figura9.14: Difference operation between a two-features input layer 'a' and a single feature overlay layer 'b' (left) - resulting in a new layer with the modified 'a' features (right)

Leitura adicional: algoritmo *Diferença*

disjoint

Testa quando as geometrias não intersectam espacialmente. Retorna VERDADEIRO se as geometrias não compartilham algum espaço juntas.

Sintaxe	disjoint(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `disjoint( geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0 ))' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `disjoint( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'POINT(4 4)' ))` → FALSE

Leitura adicional: *overlay_disjoint*

distance

Retorna a menor distância (baseada na referencia espacial) entre duas geometrias na unidade projetada.

Sintaxe	distance(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `distance( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(4 8)' ) )` → 4

distance_to_vertex

Retorna a distância ao longo da geometria para um vértice especificado.

Sintaxe	distance_to_vertex(geometry, vertex)
Argumentos	• **geometry** - Uma geometria de linha • **vertex** - índice do vértice, a partir de 0; se o valor for negativo, o índice do vértice selecionado será sua contagem total menos o valor absoluto
Exemplos	• `distance_to_vertex(geometry:=geom_from_wkt('LineString(0 0, 10 0, 10 10)'),vertex:=1)` → 10.0

end_point

Retorna o último nó de uma geometria.

Sintaxe	end_point(geometry)
Argumentos	• **geometry** - objeto geométrico
Exemplos	• `geom_to_wkt(end_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → 'Point (0 2)'

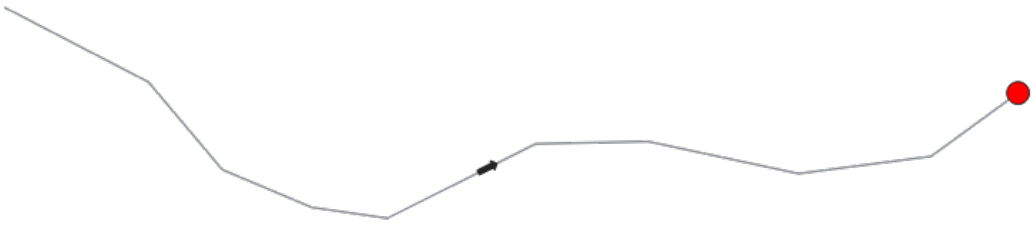


Figura9.15: End point of a line feature

Further reading: *start_point*, *Extrair vértices específicos* algorithm

exif_geotag

Cria uma geometria de pontos a partir da geotag do exif de um arquivo de imagem.

Sintaxe	exif_geotag(path)
Argumentos	• **path** - Um caminho de arquivo de imagem ou um valor de camada de mapa. Se um valor de camada de mapa for especificado, a origem do arquivo da camada será usada.
Exemplos	• `geom_to_wkt(exif_geotag('/my/photo.jpg'))` → 'Point (2 4)'

extend

Estende o início e o fim de uma geometria de linha em um valor especificado. As linhas são estendidas usando o rumo do primeiro e do último segmento da linha. Para uma multi linha, todas as partes são estendidas. As distâncias estão no Sistema de Referência Espacial desta geometria.

Sintaxe	extend(geometry, start_distance, end_distance)
Argumentos	• **geometry** - uma geometria (multi) linha • **start_distance** - distancia para extender o inicio da linha • **end_distance** - distancia para extender o final da linha
Exemplos	• `geom_to_wkt(extend(geom_from_wkt('LineString(0 0, 1 0, 1 1)'),1,2))` → 'LineString (-1 0, 1 0, 1 3)' • `geom_to_wkt(extend(geom_from_wkt('MultiLineString((0 0, 1 0, 1 1), (2 2, 0 2, 0 5))'),1,2))` → 'MultiLineString ((-1 0, 1 0, 1 3),(3 2, 0 2, 0 7))'

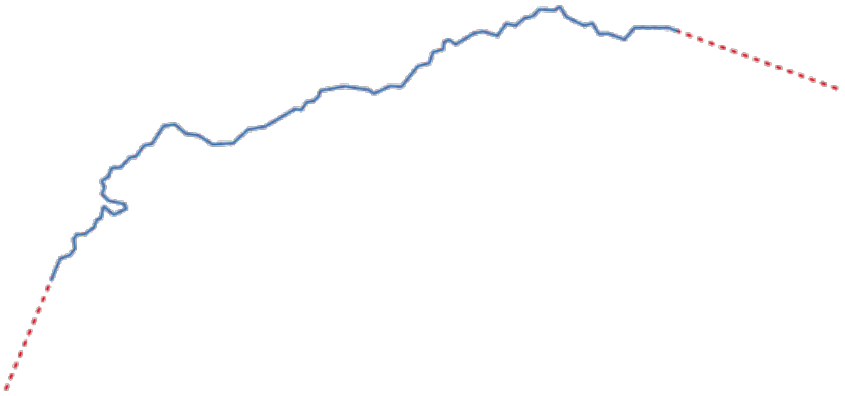


Figura9.16: Os traços vermelhos representam a extensão inicial e final da camada original

Leitura adicional: algoritmo *Linhas estendidas.*

exterior_ring

Retorna uma linha representando o anél exterior de uma geometria do tipo polígono. Se a geometria não for um polígono o resultado será NULL.

Sintaxe	exterior_ring(geometry)
Argumentos	• **geometry** - uma geometria de polígono
Exemplos	• `geom_to_wkt(exterior_ring(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),( 0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2, 0.1, 0.1 0.1))')))` → 'LineString (-1 -1, 4 0, 4 2, 0 2, -1 -1)'

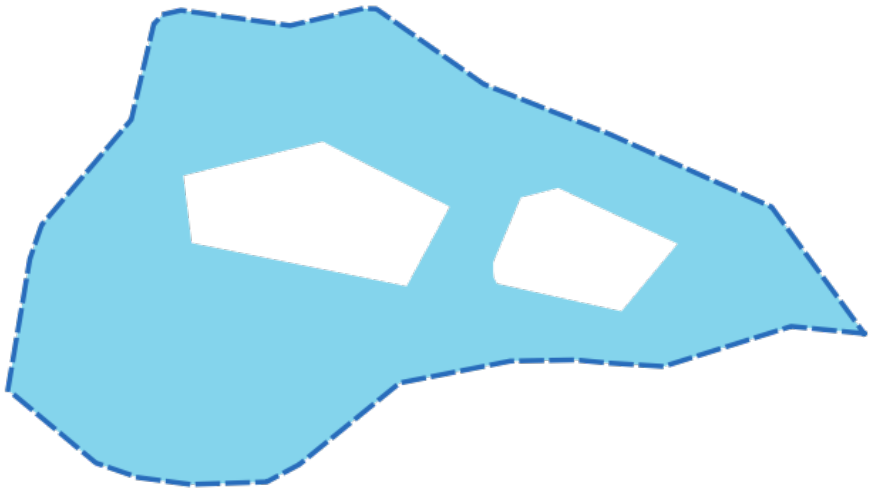


Figura9.17: The dashed line represents the exterior ring of the polygon

Further reading: *Limite* algorithm, *interior_ring_n*

extrude

Retorna uma versão extrudada da geometria de entrada (Multi-)Curve ou (Multi-)Linha com uma extensão especificada por x e y.

Sintaxe	extrude(geometry, x, y)
Argumentos	• **geometry** - uma geometria curva ou linha • **x** - extensão x, valor numérico • **y** - extensão y, valor numérico
Exemplos	• `geom_to_wkt(extrude(geom_from_wkt('LineString(1 2, 3 2, 4 3)'), 1, 2))` → 'Polygon ((1 2, 3 2, 4 3, 5 5, 4 4, 2 4, 1 2))' • `geom_to_wkt(extrude(geom_from_wkt('MultiLineString((1 2, 3 2), (4 3, 8 3))'), 1, 2))` → 'MultiPolygon (((1 2, 3 2, 4 4, 2 4, 1 2)),((4 3, 8 3, 9 5, 5 5, 4 3)))'

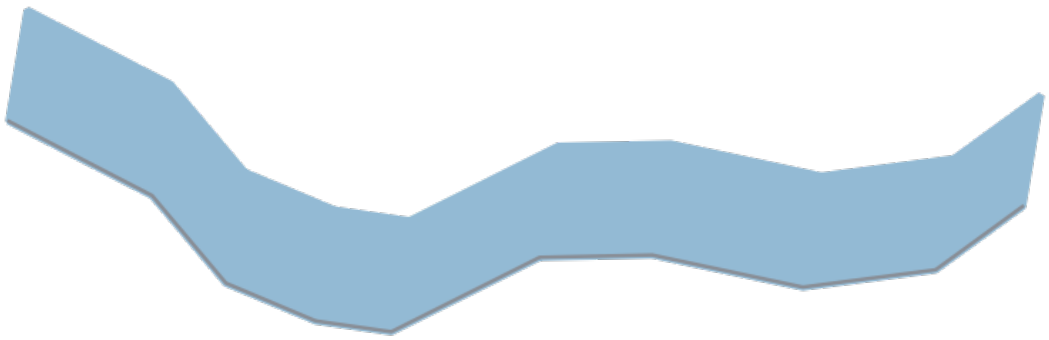


Figura9.18: Generating a polygon by extruding a line with offset in x and y directions

flip_coordinates

Retorna uma cópia da geometria com as coordenadas x e y trocadas. Útil para reparar geometrias que tiveram seus valores de latitude e longitude invertidos.

Sintaxe	flip_coordinates(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `geom_to_wkt(flip_coordinates(make_point(1, 2)))` → 'Point (2 1)' • `geom_to_wkt(flip_coordinates(geom_from_wkt('LineString(0 2, 1 0, 1 6)')))` → 'LineString (2 0, 0 1, 6 1)'

Leitura adicional: algoritmo *Trocar as coordenadas X e Y*

force_polygon_ccw

Força a geometria a respeitar a convenção onde os anéis exteriores são no sentido anti-horário, e anéis interiores são no sentido horário.

Sintaxe	force_polygon_ccw(geometry)
Argumentos	• **geometry** - Uma geometria. Qualquer geometria que não for polígono são retornadas sem modificações.
Exemplos	• `geom_to_wkt(force_polygon_ccw(geometry:=geom_from_wkt('Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))')))` → 'Polygon ((-1 -1, 4 0, 4 2, 0 2, -1 -1))'

Leitura adicional: *force_polygon_cw*, *force_rhr*

force_polygon_cw

Força a geometria a respeitar a convenção onde anéis externos são no sentido horário, e anéis internos são no sentido anti-horário.

Sintaxe	force_polygon_cw(geometry)
Argumentos	• **geometry** - Uma geometria. Qualquer geometria que não for polígono são retornadas sem modificações.
Exemplos	• `geom_to_wkt(force_polygon_cw(geometry:=geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))')))` → 'Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))'

Leitura adicional: *force_polygon_ccw*, *force_rhr*

force_rhr

Força uma geometria a respeitar a Regra da Mão Direita, na qual a área delimitada por um polígono está à direita do limite. Em particular, o anel externo é orientado no sentido horário e os anéis internos no sentido anti-horário. Devido à inconsistência na definição da Regra da Mão Direita em alguns contextos, recomenda-se o uso explícito force_polygon_cw function no lugar.

Sintaxe	force_rhr(geometry)
Argumentos	• **geometry** - Uma geometria. Qualquer geometria que não for polígono são retornadas sem modificações.
Exemplos	• `geom_to_wkt(force_rhr(geometry:=geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))')))` → 'Polygon ((-1 -1, 0 2, 4 2, 4 0, -1 -1))'

Leitura adicional: algoritmo *Regra de força da mão direita.*, *force_polygon_ccw*, *force_polygon_cw*

geom_from_gml

Retorna uma geometria a partir de uma representação GML de geometria.

Sintaxe	geom_from_gml(gml)
Argumentos	• **gml** - string representando a geometira em GML
Exemplos	• `geom_from_gml('<gml:LineString srsName="EPSG:4326"><gml:coordinates>4, 4 5,5 6,6</gml:coordinates></gml:LineString>')` → um objeto de geometria de linha

geom_from_wkb

Retorna uma geometria criada a apartir de um uma representação Well-Known Binary (WKB).

Sintaxe	geom_from_wkb(binary)
Argumentos	• **binary** - Representação Well-Known Binary (WKB) de uma geometria (como um binário blob)
Exemplos	• `geom_from_wkb( geom_to_wkb( make_point(4,5) ) )` → um objeto de geomeria ponto

geom_from_wkt

Retorna uma geometria criada a partir de uma representação Well-Known Text (WKT - Texto Bem Conhecido)

Sintaxe	geom_from_wkt(text)
Argumentos	• **text** - Representação de uma geometria em Well-Known Text (WKT)
Exemplos	• `geom_from_wkt( 'POINT(4 5)' )` → um objeto de geometria

geom_to_wkb

Retorna uma representação Well-known Binarye(WKB - Binário Bem Conhecido) da geometria

Sintaxe	geom_to_wkb(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `geom_to_wkb( @geometry )` → binary blob containing a geometry object

geom_to_wkt

Retorna a representação Well-Know Text (WKT - Texto Bem Conhecido) da geometria sem o metadado do SRID.

Sintaxe	geom_to_wkt(geometry, [precision=8]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **precision** - precisão numérica
Exemplos	• `geom_to_wkt( make_point(6, 50) )` → 'POINT(6 50)' • `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')))` → 'POINT(0 0.66666667)' • `geom_to_wkt(centroid(geom_from_wkt('Polygon((1 1, 0 0, -1 1, 1 1))')), 2)` → 'POINT(0 0.67)'

$geometry

Retorna a geometria da feição atual. Pode ser utilizada para processamento com outras funções. **ATENÇÃO: Esta função está obsoleta. É recomendado utilizar a variável de substituição @geometry.**

Sintaxe	$geometry
Exemplos	• `geom_to_wkt( $geometry )` → 'POINT(6 50)'

geometry

Retorna a geometria de uma feição.

Sintaxe	geometry(feature)
Argumentos	• **feature** - um objeto feição
Exemplos	• `geometry( @feature )` → the geometry of the current feature. Prefer using @geometry. • `geom_to_wkt( geometry( get_feature_by_id( 'streets', 1 ) ) )` → a geometria em WKT ta feição de id 1 da camada "streets", ex. 'POINT(6 50)' • `intersects( @geometry, geometry( get_feature( 'streets', 'name', 'Main St.' ) ) )` → TRUE if the current feature spatially intersects the 'Main St.' named feature in the "streets" layer

geometry_n

Retorna uma geometria específica de uma coleção de geometrias, ou NULL se a geometria de entrada não for uma coleção. Também retorna uma parte de uma geometria multi partes.

Sintaxe	geometry_n(geometry, index)
Argumentos	• **geometry** - coleção de geometria • **index** - indice da geometria a ser retornada, onde 1 é a primeira geometria na coleção
Exemplos	• `geom_to_wkt(geometry_n(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'),3))` → 'Point (1 0)'

geometry_type

Retorna uma valor em texto descrevendo o tipo de uma geometria (Point, Line ou Polygon)

Sintaxe	geometry_type(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `geometry_type( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 'Line' • `geometry_type( geom_from_wkt( 'MULTILINESTRING((2 5, 3 6, 4 8), (1 1, 0 0))') )` → 'Line' • `geometry_type( geom_from_wkt( 'POINT(2 5)') )` → 'Point' • `geometry_type( geom_from_wkt( 'POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))') )` → 'Polygon'

hausdorff_distance

Retorna a distância de Hausdorff entre duas geometrias. Este é básicamente a medição de quão similar ou diferentes duas geometrias são, quanto menor a distancia mais similares são as geometrias.

A função pode ser executada com um argumento opcional de fração densificar. Se não for especificado, é usada uma aproximação da distância padrão de Hausdorff. Essa aproximação é exata ou próxima o suficiente para um grande subconjunto de casos úteis. Exemplos destes são:

- distância computada entre Linestrings que são aproximadamente paralelas umas às outras e aproximadamente iguais em comprimento. Isso ocorre em redes lineares correspondentes.
- Teste de similaridade de geometrias.

Se a aproximação padrão fornecida por este método for insuficiente, especifique o argumento opcional de densificar fração. Especificar este argumento realiza uma densificação do segmento antes de calcular a distância discreta de Hausdorff. O parâmetro define a fração para densificar cada segmento. Cada segmento será dividido em vários subsegmentos de igual comprimento, cuja fração do comprimento total é a mais próxima da fração dada. Diminuir o parâmetro fração densificar fará com que a distância retornada se aproxime da verdadeira distância de Hausdorff para as geometrias.

Sintaxe	hausdorff_distance(geometry1, geometry2, [densify_fraction]) [] indica argumentos opcionais
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria • **densify_fraction** - densificar quantidade de fração
Exemplos	• `hausdorff_distance( geometry1:= geom_from_wkt('LINESTRING (0 0, 2 1)'),geometry2:=geom_from_wkt('LINESTRING (0 0, 2 0)'))` → 2 • `hausdorff_distance( geom_from_wkt('LINESTRING (130 0, 0 0, 0 150)'),geom_from_wkt('LINESTRING (10 10, 10 150, 130 10)'))` → 14.142135623 • `hausdorff_distance( geom_from_wkt('LINESTRING (130 0, 0 0, 0 150)'),geom_from_wkt('LINESTRING (10 10, 10 150, 130 10)'), 0.5)` → 70.0

inclination

Retorna a inclinação medida do zênite (0) ao nadir (180) no point_a ao point_b.

Sintaxe	inclination(point_a, point_b)
Argumentos	• **point_a** - geometria de ponto • **point_b** - geometria de ponto
Exemplos	• `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 5 ) )` → 0.0 • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, 0 ) )` → 90.0 • `inclination( make_point( 5, 10, 0 ), make_point( 50, 100, 0 ) )` → 90.0 • `inclination( make_point( 5, 10, 0 ), make_point( 5, 10, -5 ) )` → 180.0

interior_ring_n

Retorna um anél interior específico de uma geometria polígono, ou NULL se a geometria não for polígono.

Sintaxe	interior_ring_n(geometry, index)
Argumentos	• **geometry** - geometria do polígono • **index** - índice do interior a ser retornado, onde 1 é o primeiro anél interior
Exemplos	• `geom_to_wkt(interior_ring_n(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1),(-1 -1, 4 0, 4 2, 0 2, -1 -1))'),1))` → 'LineString (-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1)'

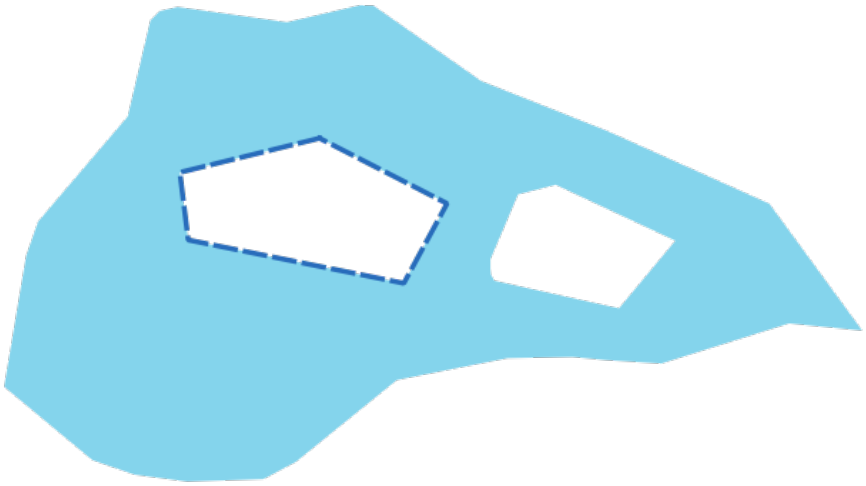


Figura9.19: The dashed line represents a specific interior ring of the polygon

Further reading: *exterior_ring*

intersection

Retorna uma geometria que representa a porção compartilhada de duas geometrias.

Sintaxe	intersection(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `geom_to_wkt( intersection( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4)' ) ) )` → 'LINESTRING(3 3, 4 4)' • `geom_to_wkt( intersection( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'MULTIPOINT(3.5 3.5, 4 5)' ) ) )` → 'POINT(3.5 3.5)'

Leitura adicional: algoritmo *Interseção*

intersects

Testa quando a geometria faz intercessão com outra. Retorna VERDADEIRO se as geometrias espacialmente intercessionam (compartilham alguma porção no espaço) e falso quando não.

Sintaxe	intersects(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `intersects( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `intersects( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'POINT(5 5)' ) )` → FALSE

Leitura adicional: *overlay_intersects*

intersects_bbox

Testa quando a geometria do retângulo envolvente sobrepõe outro retângulo envolvente de outra geometria. Retorna VERDADEIRO se as geometrias espacialmente fazem intercessão com o retângulo envolvente definido e falso se não fizerem.

Sintaxe	intersects_bbox(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `intersects_bbox( geom_from_wkt( 'POINT(4 5)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `intersects_bbox( geom_from_wkt( 'POINT(6 5)' ), geom_from_wkt( 'POLYGON((3 3, 4 4, 5 5, 3 3))' ) )` → FALSE

is_closed

Retorna VERDADEIRO se uma linha é fechada (pontos iniciais e finais são coincidentes), ou falso, se a linha não é fechada. Se a geometria não for uma linha então o resultado será NULO.

Sintaxe	is_closed(geometry)
Argumentos	• **geometry** - uma geometria de linha
Exemplos	• `is_closed(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE • `is_closed(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → TRUE

is_empty

Retorna VERDADEIRO se a geometria está vazia (sem coordenadas), falso caso a geometria não estiver vazia e NULLO se não houver geometria. Veja também is_empty_or_null.

Sintaxe	is_empty(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `is_empty(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE • `is_empty(geom_from_wkt('LINESTRING EMPTY'))` → TRUE • `is_empty(geom_from_wkt('POINT(7 4)'))` → FALSE • `is_empty(geom_from_wkt('POINT EMPTY'))` → TRUE

Leitura adicional: *is_empty_or_null*

is_empty_or_null

Returns TRUE if a geometry is NULL or empty (without coordinates) or false otherwise. This function is like the expression '@geometry IS NULL or is_empty(@geometry)'

Sintaxe	is_empty_or_null(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `is_empty_or_null(NULL)` → TRUE • `is_empty_or_null(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)'))` → FALSE • `is_empty_or_null(geom_from_wkt('LINESTRING EMPTY'))` → TRUE • `is_empty_or_null(geom_from_wkt('POINT(7 4)'))` → FALSE • `is_empty_or_null(geom_from_wkt('POINT EMPTY'))` → TRUE

Leitura adicional: *is_empty*, *NULL*

is_multipart

Retorna VERDADEIRO se a geometria não for Multi tipo.

Sintaxe	is_multipart(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `is_multipart(geom_from_wkt('MULTIPOINT ((0 0),(1 1),(2 2))'))` → TRUE • `is_multipart(geom_from_wkt('POINT (0 0)'))` → FALSE

is_valid

Retorna VERDADEIRO se a geometria for válida; se for bem formada em 2D de acordo com as regras OGC.

Sintaxe	is_valid(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `is_valid(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2, 0 0)'))` → TRUE • `is_valid(geom_from_wkt('LINESTRING(0 0)'))` → FALSE

Leintura adicional: *make_valid*, algoritmo *Verificar a validade* algorithm

$length

Retorna o comprimento de uma linha. Se você precisar do comprimento da borda de um polígono, use $perimeter. O comprimento calculado por esta função respeita tanto a configuração do elipsóide do projeto atual quanto as configurações da unidade de distância. Por exemplo, se um elipsóide foi definido para o projeto, o comprimento calculado será elipsoidal, e se nenhum elipsóide for definido, o comprimento calculado será planimétrico.

Sintaxe	$length
Exemplos	• `$comprimento` → 42.4711

length

Retorna o número de caracteres em uma string ou o tamanho de uma geometria linha.

Variante string

Retorna o número de caracteres em uma string.

Sintaxe	length(string)
Argumentos	• **string** - string para contar o comprimento de
Exemplos	• `length('hello')` → 5

Variante geometria

Calcule o comprimento de um objeto de geometria linha. Os cálculos são sempre planimétricos no Sistema de Referência Espacial (SRS) dessa geometria, e as unidades do comprimento retornado corresponderão às unidades do SRS. Isso difere dos cálculos realizados pela função $length, que realizará cálculos elipsoidais com base nas configurações de elipsóide e unidade de distância do projeto.

Sintaxe	length(geometry)
Argumentos	• **geometry** - objeto de geometria linha
Exemplos	• `length(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4.0

Leitura adicional: *straight_distance_2d*

length3D

Calcule o comprimento de um objeto de linha geométrica. Os cálculos são sempre planimétricos no Sistema de Referência Espacial (SRS) dessa geometria, e as unidades do comprimento retornado corresponderão às unidades do SRS. Isso difere dos cálculos realizados pela função $length, que realizará cálculos elipsoidais com base nas configurações de elipsóide e unidade de distância do projeto.

Sintaxe	length3D(geometry)
Argumentos	• **geometry** - objeto de geometria linha
Exemplos	• `length3D(geom_from_wkt('LINESTRINGZ(0 0 0, 3 0 4)'))` → 5.0

line_interpolate_angle

Retorna o ângulo paralelo à geometria a uma distância especificada ao longo de uma geometria de linhas. Os ângulos estão em graus no sentido horário do norte.

Sintaxe	line_interpolate_angle(geometry, distance)
Argumentos	• **geometry** - Uma geometria de linha • **distance** - distância ao longo da linha para interpolar
Exemplos	• `line_interpolate_angle(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),distance:=5)` → 90.0

line_interpolate_point

Retorna o ponto interpolado por uma distância específica ao longo de uma geometria de linha.

Sintaxe	line_interpolate_point(geometry, distance)
Argumentos	• **geometry** - Uma geometria de linha • **distance** - distância ao longo da linha para interpolar
Exemplos	• `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 8 0)'), distance:=5))` → 'Point (5 0)' • `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 1 1, 2 0)'), distance:=2.1))` → 'Point (1.48492424 0.51507576)' • `geom_to_wkt(line_interpolate_point(geometry:=geom_from_wkt('LineString(0 0, 1 0)'), distance:=2))` → NULL

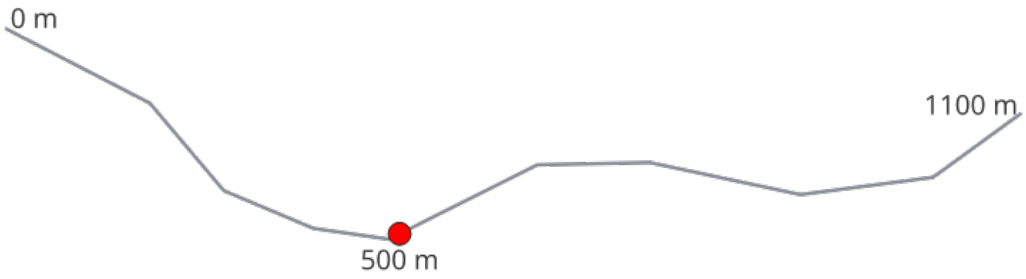


Figura9.20: Ponto interpolado a 500m do início da linha

Leitura adicional: algoritmo *Interpolar ponto na linha*

line_interpolate_point_by_m

Returns the point interpolated by a matching M value along a linestring geometry.

Sintaxe	line_interpolate_point_by_m(geometry, m, [use_3d_distance=false]) [] indica argumentos opcionais
Argumentos	• **geometry** - Uma geometria de linha • **m** - an M value • **use_3d_distance** - controls whether 2D or 3D distances between vertices should be used during interpolation (this option is only considered for lines with z values)
Exemplos	• `geom_to_wkt(line_interpolate_point_by_m(geom_from_wkt('LineStringM(0 0 0, 10 10 10)'), m:=5))` → 'Point (5 5)'

line_locate_m

Returns the distance along a linestring corresponding to the first matching interpolated M value.

Sintaxe	line_locate_m(geometry, m, [use_3d_distance=false]) [] indica argumentos opcionais
Argumentos	• **geometry** - Uma geometria de linha • **m** - an M value • **use_3d_distance** - controls whether 2D or 3D distances between vertices should be used during interpolation (this option is only considered for lines with z values)
Exemplos	• `line_locate_m(geometry:=geom_from_wkt('LineStringM(0 0 0, 10 10 10)'),m:=5)` → 7.07106

line_locate_point

Retorna a distância ao longo de uma linha correspondente à posição mais próxima da linha a uma geometria de ponto especificada.

Sintaxe	line_locate_point(geometry, point)
Argumentos	• **geometry** - Uma geometria de linha • **point** - geometria do ponto para localizar a posição mais próxima na cadeia de linhas para
Exemplos	• `line_locate_point(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),point:=geom_from_wkt('Point(5 0)'))` → 5.0

line_merge

Retorna uma geometria LineString ou MultiLineString, em que quaisquer LineStrings conectadas da geometria de entrada foram mescladas em uma única cadeia de linhas. Esta função retornará NULL se passar uma geometria que não seja uma LineString/MultiLineString.

Sintaxe	line_merge(geometry)
Argumentos	• **geometry** - uma geometria LineString/MultiLineString
Exemplos	• `geom_to_wkt(line_merge(geom_from_wkt('MULTILINESTRING((0 0, 1 1),(1 1, 2 2))')))` → 'LineString(0 0,1 1,2 2)' • `geom_to_wkt(line_merge(geom_from_wkt('MULTILINESTRING((0 0, 1 1),(11 1, 21 2))')))` → 'MultiLineString((0 0, 1 1),(11 1, 21 2)'

line_substring

Retorna a parte de uma geometria de linha (ou curva) que fica entre as distâncias inicial e final especificadas (medidas a partir do início da linha). Os valores Z e M são interpolados linearmente a partir dos valores existentes.

Sintaxe	line_substring(geometry, start_distance, end_distance)
Argumentos	• **geometry** - uma geometria de linha ou curva • **start_distance** - distância de inicio da sub linha • **end_distance** - distância para o fim da sub linha
Exemplos	• `geom_to_wkt(line_substring(geometry:=geom_from_wkt('LineString(0 0, 10 0)'),start_distance:=2,end_distance:=6))` → 'LineString (2 0,6 0)'

Figura9.21: Linha de substring com começo posto em 0 metros e final 250 metros.

Leitura adicional: algoritmo *Linha de substring.*

m

Retorna o valor m (medição) de uma geometria de ponto.

Sintaxe	m(geometry)
Argumentos	• **geometry** - uma geometria de ponto
Exemplos	• `m( geom_from_wkt( 'POINTM(2 5 4)' ) )` → 4

m_at

Retrieves a m coordinate of the geometry, or NULL if the geometry has no m value.

Sintaxe	m_at(geometry, vertex)
Argumentos	• **geometry** - objeto geométrico • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Exemplos	• `m_at(geom_from_wkt('LineStringZM(0 0 0 0, 10 10 0 5, 10 10 0 0)'), 1)` → 5

m_max

Retorna o valor máximo de m (medida) de uma geometria.

Sintaxe	m_max(geometry)
Argumentos	• **geometry** - uma geometria contendo valores m
Exemplos	• `m_max( make_point_m( 0,0,1 ) )` → 1 • `m_max(make_line( make_point_m( 0,0,1 ), make_point_m( -1,-1,2 ), make_point_m( -2,-2,0 ) ) )` → 2

m_min

Retorna o valor mínimo de m (medida) de uma geometria.

Sintaxe	m_min(geometry)
Argumentos	• **geometry** - uma geometria contendo valores m
Exemplos	• `m_min( make_point_m( 0,0,1 ) )` → 1 • `m_min(make_line( make_point_m( 0,0,1 ), make_point_m( -1, -1,2 ), make_point_m( -2,-2,0 ) ) )` → 0

main_angle

Retorna o ângulo do eixo longo (no sentido horário, em graus a partir do Norte) do retângulo envolvente mínimo orientado, que cobre completamente a geometria.

Sintaxe	main_angle(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `main_angle(geom_from_wkt('Polygon ((321577 129614, 321581 129618, 321585 129615, 321581 129610, 321577 129614))'))` → 38.66

make_circle

Cria um polígono circular.

Sintaxe	make_circle(center, radius, [segments=36]) [] indica argumentos opcionais
Argumentos	• **center** - ponto central do círculo • **radius** - raio do círculo • **segments** - argumento opcional para segmentação de polígonos. Por padrão esse valor é 36
Exemplos	• `geom_to_wkt(make_circle(make_point(10,10), 5, 4))` → 'Polygon ((10 15, 15 10, 10 5, 5 10, 10 15))' • `geom_to_wkt(make_circle(make_point(10,10,5), 5, 4))` → 'PolygonZ ((10 15 5, 15 10 5, 10 5 5, 5 10 5, 10 15 5))' • `geom_to_wkt(make_circle(make_point(10,10,5,30), 5, 4))` → 'PolygonZM ((10 15 5 30, 15 10 5 30, 10 5 5 30, 5 10 5 30, 10 15 5 30))'

make_ellipse

Cria um polígono elíptico.

Sintaxe	make_ellipse(center, semi_major_axis, semi_minor_axis, azimuth, [segments=36]) [] indica argumentos opcionais
Argumentos	• **center** - ponto central da elipse • **semi_major_axis** - eixo semi maior da elipse • **semi_minor_axis** - eixo semi menor da elipse • **azimuth** - orientação da elipse • **segments** - argumento opcional para segmentação de polígonos. Por padrão esse valor é 36
Exemplos	• `geom_to_wkt(make_ellipse(make_point(10,10), 5, 2, 90, 4))` → 'Polygon ((15 10, 10 8, 5 10, 10 12, 15 10))' • `geom_to_wkt(make_ellipse(make_point(10,10,5), 5, 2, 90, 4))` → 'PolygonZ ((15 10 5, 10 8 5, 5 10 5, 10 12 5, 15 10 5))' • `geom_to_wkt(make_ellipse(make_point(10,10,5,30), 5, 2, 90, 4))` → 'PolygonZM ((15 10 5 30, 10 8 5 30, 5 10 5 30, 10 12 5 30, 15 10 5 30))'

make_line

Cria uma geometria de linha a partir de uma série de geometrias de ponto.

Lista de variantes de argumentos

Os vértices de linha são especificados como argumentos separados para a função.

Sintaxe	make_line(point1, point2, …)
Argumentos	• **point** - uma geometria de ponto (ou array de pontos)
Exemplos	• `geom_to_wkt(make_line(make_point(2,4),make_point(3,5)))` → 'LineString (2 4, 3 5)' • `geom_to_wkt(make_line(make_point(2,4),make_point(3,5), make_point(9,7)))` → 'LineString (2 4, 3 5, 9 7)'

Variante array

Os vértices de linha são especificados como um array de pontos.

Sintaxe	make_line(array)
Argumentos	• **array** - array de pontos
Exemplos	• `geom_to_wkt(make_line(array(make_point(2,4),make_point(3, 5),make_point(9,7))))` → 'LineString (2 4, 3 5, 9 7)'

make_point

Cria uma geometria de ponto a partir de x e y (e valores z e m opcionais).

Sintaxe	make_point(x, y, [z], [m]) [] indica argumentos opcionais
Argumentos	• **x** - coordenada x do ponto • **y** - coordenada y do ponto • **z** - coordenada z opcional do ponto • **m** - valor m opcional do ponto
Exemplos	• `geom_to_wkt(make_point(2,4))` → 'Point (2 4)' • `geom_to_wkt(make_point(2,4,6))` → 'PointZ (2 4 6)' • `geom_to_wkt(make_point(2,4,6,8))` → 'PointZM (2 4 6 8)'

make_point_m

Cria uma geometria de ponto a partir de uma coordenada x, y e um valor m.

Sintaxe	make_point_m(x, y, m)
Argumentos	• **x** - coordenada x do ponto • **y** - coordenada y do ponto • **m** - valor m do ponto
Exemplos	• `geom_to_wkt(make_point_m(2,4,6))` → 'PointM (2 4 6)'

make_polygon

Cria uma geometria de polígono a partir de um anel externo e uma série opcional de geometrias de anel interno.

Sintaxe	make_polygon(outerRing, [innerRing1], [innerRing2], …) [] indica argumentos opcionais
Argumentos	• **outerRing** - geometria linha fechada para o anel externo do polígono • **innerRing** - linha fechada opcional para o anel interior
Exemplos	• `geom_to_wkt(make_polygon(geom_from_wkt('LINESTRING( 0 0, 0 1, 1 1, 1 0, 0 0 )')))` → 'Polygon ((0 0, 0 1, 1 1, 1 0, 0 0))' • `geom_to_wkt(make_polygon(geom_from_wkt('LINESTRING( 0 0, 0 1, 1 1, 1 0, 0 0 )'),geom_from_wkt('LINESTRING( 0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2 0.1, 0.1 0.1 )'), geom_from_wkt('LINESTRING( 0.8 0.8, 0.8 0.9, 0.9 0.9, 0.9 0.8, 0.8 0.8 )')))` → 'Polygon ((0 0, 0 1, 1 1, 1 0, 0 0),(0.1 0.1, 0.1 0.2, 0.2 0.2, 0.2 0.1, 0.1 0.1),(0.8 0.8, 0.8 0.9, 0.9 0.9, 0.9 0.8, 0.8 0.8))'

make_rectangle_3points

Cria um retângulo a partir de 3 pontos.

Sintaxe	make_rectangle_3points(point1, point2, point3, [option=0]) [] indica argumentos opcionais
Argumentos	• **point1** - Primeiro ponto. • **point2** - Segundo ponto. • **point3** - Terceiro ponto. • **option** - Um argumento opcional para construir o retângulo. Por padrão este valor é 0. O valor pode ser 0 (distância) ou 1 (projetado). Opção distância: A segunda distância é igual à distância entre o 2º e o 3º ponto. Opção projetada: A segunda distância é igual à distância da projeção perpendicular do 3º ponto sobre o segmento ou sua extensão.
Exemplos	• `geom_to_wkt(make_rectangle_3points(make_point(0, 0), make_point(0,5), make_point(5, 5), 0))` → 'Polygon ((0 0, 0 5, 5 5, 5 0, 0 0))' • `geom_to_wkt(make_rectangle_3points(make_point(0, 0), make_point(0,5), make_point(5, 3), 1))` → 'Polygon ((0 0, 0 5, 5 5, 5 0, 0 0))'

make_regular_polygon

Cria um polígono regular.

Sintaxe	make_regular_polygon(center, radius, number_sides, [circle=0]) [] indica argumentos opcionais
Argumentos	• **center** - centro do polígono regular • **radius** - segundo ponto. A primeira se o polígono regular estiver inscrito. O ponto médio do primeiro lado se o polígono regular for circunscrito. • **numero_lados** - Número de lados/bordas do polígono regular • **circle** - Argumento opcional para a construção do poligono regular. Por padrão o valor é 0. Valor pode ser 0 (inscrito) ou 1 (circunscrito)
Exemplos	• `geom_to_wkt(make_regular_polygon(make_point(0,0), make_point(0,5), 5))` → 'Polygon ((0 5, 4.76 1.55, 2.94 -4.05, -2.94 -4.05, -4.76 1.55, 0 5))' • `geom_to_wkt(make_regular_polygon(make_point(0,0), project(make_point(0,0), 4.0451, radians(36)), 5))` → 'Polygon ((0 5, 4.76 1.55, 2.94 -4.05, -2.94 -4.05, -4.76 1.55, 0 5))'

make_square

Cria um quadrado a partir da diagonal.

Sintaxe	make_square(point1, point2)
Argumentos	• **point1** - Primeiro ponto da diagonal • **point2** - Último ponto da diagonal
Exemplos	• `geom_to_wkt(make_square( make_point(0,0), make_point(5, 5)))` → 'Polygon ((0 0, -0 5, 5 5, 5 0, 0 0))' • `geom_to_wkt(make_square( make_point(5,0), make_point(5, 5)))` → 'Polygon ((5 0, 2.5 2.5, 5 5, 7.5 2.5, 5 0))'

make_triangle

Cria um polígono triangular.

Sintaxe	make_triangle(point1, point2, point3)
Argumentos	• **point1** - primeiro ponto do triângulo • **point2** - segundo ponto do triângulo • **point3** - terceiro ponto do triângulo
Exemplos	• `geom_to_wkt(make_triangle(make_point(0,0), make_point(5, 5), make_point(0,10)))` → 'Triangle ((0 0, 5 5, 0 10, 0 0))' • `geom_to_wkt(boundary(make_triangle(make_point(0,0), make_point(5,5), make_point(0,10))))` → 'LineString (0 0, 5 5, 0 10, 0 0)'

make_valid

Retorna uma geometria válida ou uma geometria vazia se a geometria não puder ser tornada valida.

Sintaxe	make_valid(geometry, [method=structure], [keep_collapsed=false]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **method** - algoritmo de reparo. Pode ser 'structure' ou 'linework'. A opção 'linework' combina todos os anéis em um conjunto de linhas com nós e então extrai polígonos válidos desse linework. O método 'structure' primeiro torna todos os anéis válidos e, em seguida, mescla as cascas e subtrai os buracos das cascas para gerar um resultado válido. Assume que os furos e cascas são categorizados corretamente. • **keep_collapsed** - se definido como true, os componentes que foram recolhidos em uma dimensionalidade inferior serão mantidos. Por exemplo, um anel colapsando em uma linha ou uma linha colapsando em um ponto.
Exemplos	• `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8, 3 2, 4 2))')))` → 'Polygon ((3 2, 5 8, 4 1, 3 2))' • `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8, 3 2, 4 2))'), 'linework'))` → 'GeometryCollection (Polygon ((5 8, 4 1, 3 2, 5 8)),LineString (3 2, 4 2))' • `geom_to_wkt(make_valid(geom_from_wkt('POLYGON((3 2, 4 1, 5 8))'), method:='linework'))` → 'Polygon ((3 2, 4 1, 5 8, 3 2))' • `make_valid(geom_from_wkt('LINESTRING(0 0)'))` → An empty geometry

Leitura adicional: algoritmo *is_valid*, *Corrigir geometrias*

minimal_circle

Retorna o círculo mínimo de uma geometria. Ele representa o círculo mínimo que envolve todas as geometrias dentro do conjunto.

Sintaxe	minimal_circle(geometry, [segments=36]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **segments** - argumento opcional para segmentação de polígonos. Por padrão esse valor é 36
Exemplos	• `geom_to_wkt( minimal_circle( geom_from_wkt( 'LINESTRING(0 5, 0 -5, 2 1)' ), 4 ) )` → 'Polygon ((0 5, 5 -0, -0 -5, -5 0, 0 5))' • `geom_to_wkt( minimal_circle( geom_from_wkt( 'MULTIPOINT(1 2, 3 4, 3 2)' ), 4 ) )` → 'Polygon ((3 4, 3 2, 1 2, 1 4, 3 4))'

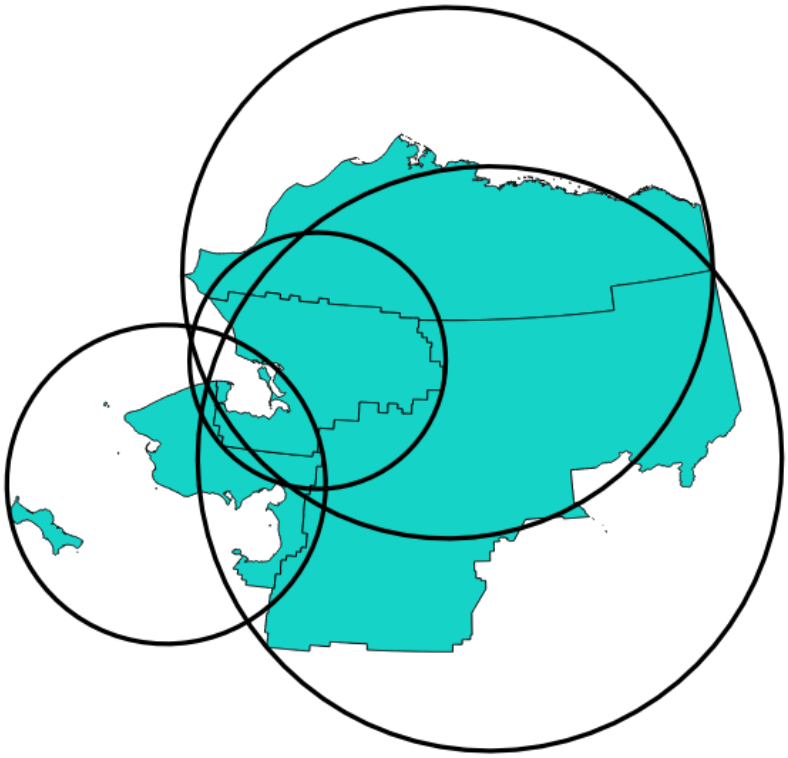


Figura9.22: Minimal enclosing circle of each feature

Leitura adicional: algoritmo *Círculos mínimos fechados.*

nodes_to_points

Retorna uma geometria multiponto que consiste em cada nó na geometria de entrada.

Sintaxe	nodes_to_points(geometry, [ignore_closing_nodes=false]) [] indica argumentos opcionais
Argumentos	• **geometry** - objeto geométrico • **ignore_closing_nodes** - argumento opcional que especifica se deve incluir nós duplicados que fecham linhas ou anéis de polígonos. O padrão é false, definido como true para evitar incluir esses nós duplicados na coleção de saída.
Exemplos	• `geom_to_wkt(nodes_to_points(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'MultiPoint ((0 0),(1 1),(2 2))' • `geom_to_wkt(nodes_to_points(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1))'),true))` → 'MultiPoint ((-1 -1),(4 0),(4 2),(0 2))'

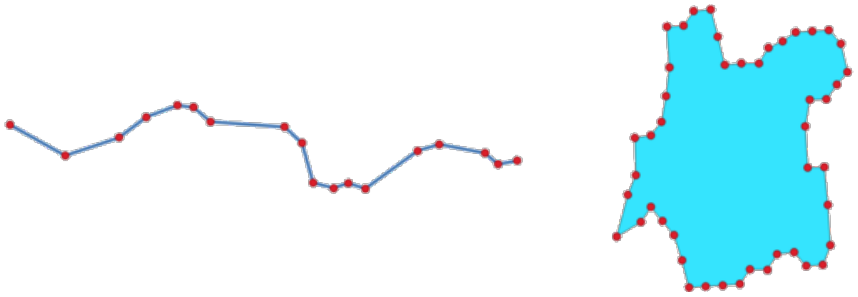


Figura9.23: Multi-point feature extracted from vertices

Leitura adicional: algoritmo *Extrair vértices*

num_geometries

Retorna o número de geometrias em uma coleção de geometrias, ou o número de partes em uma geometria multi parte. A função retorna NULO se a geometria de entrada não for uma coleção.

Sintaxe	num_geometries(geometry)
Argumentos	• **geometry** - uma coleção de geometria ou uma geometria multi parte
Exemplos	• `num_geometries(geom_from_wkt('GEOMETRYCOLLECTION(POINT(0 1), POINT(0 0), POINT(1 0), POINT(1 1))'))` → 4 • `num_geometries(geom_from_wkt('MULTIPOINT((0 1), (0 0), (1 0))'))` → 3

num_interior_rings

Retorna o número de anéis interiores em um polígono ou coleção de geometrias, ou NULL se a geometria de entrada não for polígono ou coleção.

Sintaxe	num_interior_rings(geometry)
Argumentos	• **geometry** - geometria de entrada
Exemplos	• `num_interior_rings(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1))'))` → 1

num_points

Retorna o número de vértices em uma geometria.

Sintaxe	num_points(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `num_points(@geometry)` → number of vertices in the current feature's geometry

num_rings

Retorna o número de anéis (incluindo anéis externos) em um polígono ou coleção de geometrias ou NULO se a geometria de entrada não for um polígono ou coleção.

Sintaxe	num_rings(geometry)
Argumentos	• **geometry** - geometria de entrada
Exemplos	• `num_rings(geom_from_wkt('POLYGON((-1 -1, 4 0, 4 2, 0 2, -1 -1),(-0.1 -0.1, 0.4 0, 0.4 0.2, 0 0.2, -0.1 -0.1))'))` → 2

offset_curve

Retorna uma geometria formada pelo deslocamento de uma geometria linha para o lado. As distâncias estão no Sistema de Referência Espacial desta geometria.

Sintaxe	offset_curve(geometry, distance, [segments=8], [join=1], [miter_limit=2.0]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria (multi) linha • **distance** - distância de deslocamento. Valores positivos serão gerados a esquerda das linhas, valores negativos para a direita • **segments** - número de segmentos a ser usado para representar um quarto de círculo quando um estilo de união arredondado é utilizado. Valores maiores resultam em linhas mais suaves com mais nós. • **join** - estilo de junção dos cantos, onde 1 = round, 2 = miter e 3 = bevel • **miter_limit** - limite na proporção de esquadria usada para cantos muito afiados (ao usar apenas junções de esquadria)
Exemplos	• `offset_curve(@geometry, 10.5)` → line offset to the left by 10.5 units • `offset_curve(@geometry, -10.5)` → line offset to the right by 10.5 units • `offset_curve(@geometry, 10.5, segments:=16, join:=1)` → line offset to the left by 10.5 units, using more segments to result in a smoother curve • `offset_curve(@geometry, 10.5, join:=3)` → line offset to the left by 10.5 units, using a beveled join

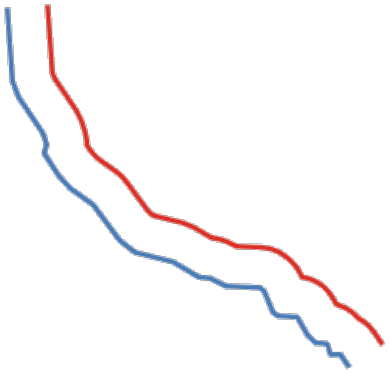


Figura9.24: Em azul a camada de origem, em vermelho a de deslocamento

Leitura adicional: algoritmo *Linhas de desvio.*

order_parts

Ordena as partes de um MultiGeometry por um determinado critério

Sintaxe	order_parts(geometry, orderby, [ascending=true]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria de vários tipos • **orderby** - uma expressão que define os critérios de ordenação • **ascending** - booleano, verdadeiro para ascendente, falso para descendente
Exemplos	• `geom_to_wkt(order_parts(geom_from_wkt('MultiPolygon (((1 1, 5 1, 5 5, 1 5, 1 1)),((1 1, 9 1, 9 9, 1 9, 1 1)))'), 'area(@geometry)', False))` → 'MultiPolygon (((1 1, 9 1, 9 9, 1 9, 1 1)),((1 1, 5 1, 5 5, 1 5, 1 1)))' • `geom_to_wkt(order_parts(geom_from_wkt('LineString(1 2, 3 2, 4 3)'), '1', True))` → 'LineString(1 2, 3 2, 4 3)'

oriented_bbox

Retorna uma geometria que representa o menor retângulo envolvente orientado de uma geometria de entrada.

Sintaxe	oriented_bbox(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `geom_to_wkt( oriented_bbox( geom_from_wkt( 'MULTIPOINT(1 2, 3 4, 3 2)' ) ) )` → 'Polygon ((3 2, 3 4, 1 4, 1 2, 3 2))'

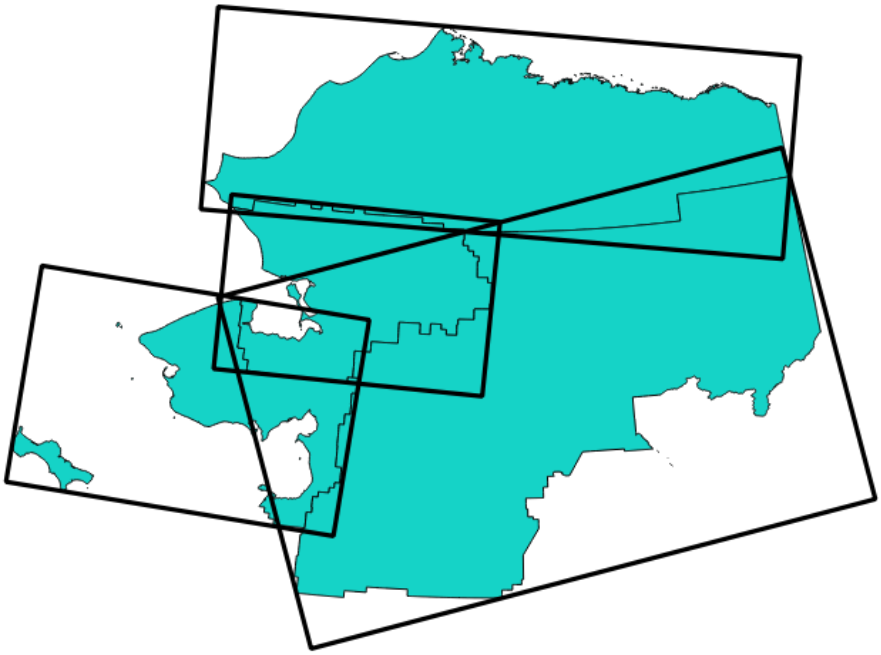


Figura9.25: Orientação mínima da caixa limitante.

Leitura adicional: algoritmo *Orientação mínima da caixa limitante.*

overlaps

Testa quando a geometria sobrepõe outra. Retorna VERDADEIRO se as geometrias compartilham espaço, são da mesma dimensão, mas não estão completamente contidas umas nas outras.

Sintaxe	overlaps(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `overlaps( geom_from_wkt( 'LINESTRING(3 5, 4 4, 5 5, 5 3)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `overlaps( geom_from_wkt( 'LINESTRING(0 0, 1 1)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → FALSE

overlay_contains

Retorna quando a feição atual contém espacialmente no mínimo uma feição de uma camada alvo, ou um lista baseada em expressão de resultados de feições na camada alvo contidas pela feição atual.

Leia mais sobre o predicado GEOS "Contains" , como descrito na função do PostGIS **`ST_Contains<https://postgis.net/docs/ST_Contains.html>`_**.

Sintaxe	overlay_contains(layer, [expression], [filter], [limit], [cache=false]) [] indica argumentos opcionais
Argumentos	• **layer*** - a camada cuja sobreposição é verificada • **expression** - uma expressão opcional para avaliar as feições da camada alvo. Se não fornecida, a função irá retornar um booleano indicando que possui ao menos uma combinação. • **filter** - uma expressão opcional para filtrar as feições alvo a serem verificadas. Se não fornecida, todas as feições serão verificadas. • **limit** - um inteiro opcional para limitar o número de feições combinadas. Se não fornecido, todas as combinações serão retornadas. • **cache** - configure como verdadeiro para construir um índice espacial local (na maioria das vezes, isso é indesejado, a menos que esteja trabalhando em um provedor de dados particularmente lento)
Exemplos	• `overlay_contains('regions')` → VERDADEIRO se a feição espacialmente contiver uma region • `overlay_contains('regions', filter:= population > 10000)` → VERDADEIRO se a feição atual espacialmente contém uma 'region' com 'population' maior que 10000 • `overlay_contains('regions', name)` → uma lista de 'name', para regiões contidas pela feição atual • `array_to_string(overlay_contains('regions', name))` → uma lista de nomes em uma string separada por vírgula, para as regiões contidas na feição atual. • `array_sort(overlay_contains(layer:='regions', expression:="name", filter:= population > 10000))` → uma lista ordenada de 'name', para as regiões contidas pela feição atual e com 'population' maior que 10000 • `overlay_contains(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions contained in the current feature

Leitura adicional: *contains*, *manipulação de arrays*, algoritmo *Selecionar pela localização*

overlay_crosses

Retorna quando a feição atual espacialmente cruza uma feição de uma camada alvo, ou uma lista de resultados baseados em expressão para as feições na camada alvo cruzadas pela feição atual.

Leia mais sobre o predicado GEOS "Crosses", conforme descrito na função PostGIS ST_Crosses.

Sintaxe	overlay_crosses(layer, [expression], [filter], [limit], [cache=false]) [] indica argumentos opcionais
Argumentos	• **layer*** - a camada cuja sobreposição é verificada • **expression** - uma expressão opcional para avaliar as feições da camada alvo. Se não fornecida, a função irá retornar um booleano indicando que possui ao menos uma combinação. • **filter** - uma expressão opcional para filtrar as feições alvo a serem verificadas. Se não fornecida, todas as feições serão verificadas. • **limit** - um inteiro opcional para limitar o número de feições combinadas. Se não fornecido, todas as combinações serão retornadas. • **cache** - configure como verdadeiro para construir um índice espacial local (na maioria das vezes, isso é indesejado, a menos que esteja trabalhando em um provedor de dados particularmente lento)
Exemplos	• `overlay_crosses('regions')` → VERDADEIRO se a feição atual espacialmente cruzar uma região. • `overlay_crosses('regions', filter:= population > 10000)` → VERDADEIRO se a feição atual espacialmente cruzar com uma 'region' com 'population' maior que 10000 • `overlay_crosses('regions', name)` → uma lista de nomes, para as 'regions' cruzadas pela feição atual • `array_to_string(overlay_crosses('regions', name))` → uma string separada por vírgulas da lista de 'name', para 'regions' cruzadas pela feição atual • `array_sort(overlay_crosses(layer:='regions', expression:="name", filter:= population > 10000))` → uma lista ordenada de 'name', para cada 'region' cruzada pela feição atual e com 'population' maior que 10000 • `overlay_crosses(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions crossed by the current feature

Leitura adicional: *crosses*, *manipulação de array*, algoritmo *Selecionar pela localização*

overlay_disjoint

Retorna se a feição atual está separada espacialmente de todos as feições de uma camada de destino ou de uma lista de resultados baseados em expressão para as feições na camada de destino que estão separados da feição atual.

Leia mais sobre o predicado GEOS "Disjoint", conforme descrito na função PostGIS ST_Disjoint.

Sintaxe	overlay_disjoint(layer, [expression], [filter], [limit], [cache=false]) [] indica argumentos opcionais
Argumentos	• **layer*** - a camada cuja sobreposição é verificada • **expression** - uma expressão opcional para avaliar as feições da camada alvo. Se não fornecida, a função irá retornar um booleano indicando que possui ao menos uma combinação. • **filter** - uma expressão opcional para filtrar as feições alvo a serem verificadas. Se não fornecida, todas as feições serão verificadas. • **limit** - um inteiro opcional para limitar o número de feições combinadas. Se não fornecido, todas as combinações serão retornadas. • **cache** - configure como verdadeiro para construir um índice espacial local (na maioria das vezes, isso é indesejado, a menos que esteja trabalhando em um provedor de dados particularmente lento)
Exemplos	• `overlay_disjoint('regions')` → VERDADEIRO se a feição atual for espacialmente separada de todas as 'regions' • `overlay_disjoint('regions', filter:= population > 10000)` → VERDADEIRO se a feição atual for espacialmente separada de todas 'regions' com 'population' maior que 10000 • `overlay_disjoint('regions', name)` → uma lista de 'name', para 'regions' que estão espacialmente separadas da feição atual • `array_to_string(overlay_disjoint('regions', name))` → uma string de valores separados por vírgula da lista de 'name', com cada 'regions' separadas espacialmente da feição atual • `array_sort(overlay_disjoint(layer:='regions', expression:="name", filter:= population > 10000))` → uma lista ordenada de 'name', de cada 'region' espacialmente separada da feição atual e com população maior que 10000 • `overlay_disjoint(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions spatially disjoint from the current feature

Leitura adicional: *disjoint*, *manipulação de array*, algoritimo *Selecionar pela localização*

overlay_equals

Returns whether the current feature's geometry is exactly equal to at least one feature's geometry from a target layer, or an array of expression-based results for the features in the target layer whose geometry is exactly equal to the current feature's geometry. Note that the order of vertices matters.

Sintaxe	overlay_equals(layer, [expression], [filter], [limit], [cache=false]) [] indica argumentos opcionais
Argumentos	• **layer*** - a camada cuja sobreposição é verificada • **expression** - uma expressão opcional para avaliar as feições da camada alvo. Se não fornecida, a função irá retornar um booleano indicando que possui ao menos uma combinação. • **filter** - uma expressão opcional para filtrar as feições alvo a serem verificadas. Se não fornecida, todas as feições serão verificadas. • **limit** - um inteiro opcional para limitar o número de feições combinadas. Se não fornecido, todas as combinações serão retornadas. • **cache** - configure como verdadeiro para construir um índice espacial local (na maioria das vezes, isso é indesejado, a menos que esteja trabalhando em um provedor de dados particularmente lento)
Exemplos	• `overlay_equals('regions')` → TRUE if the current feature's geometry is exactly the same as the geometry of a region • `overlay_equals('regions', filter:= population > 10000)` → TRUE if the current feature's geometry is exactly the same as the geometry of a region whose population is greater than 10000 • `overlay_equals('regions', name)` → an array of names, for the regions exactly equal to the current feature • `array_to_string(overlay_equals('regions', name))` → a string as a comma separated list of names, for the regions exactly equal to the current feature • `array_sort(overlay_equals(layer:='regions', expression:="name", filter:= population > 10000))` → an ordered array of names, for the regions exactly equal to the current feature and with a population greater than 10000 • `overlay_equals(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions exactly equal to the current feature

Nota

This function requires exact equality of geometries, meaning that the geometries 'LINESTRING(0 0, 1 1)' and 'LINESTRING(1 1, 0 0)' are different.

Leitura adicional: *manipulação de array*, algoritmo *Selecionar pela localização*

overlay_intersects

Retorna se feição atual intercepta espacialmente pelo menos uma feição de uma camada de destino ou uma matriz de resultados baseados em expressão para as feições na camada de destino interceptados pela feição atual.

Leia mais sobre o predicado GEOS “Intersects”, como descrito na função PostGIS ST_Intersects.

Sintaxe	overlay_intersects(layer, [expression], [filter], [limit], [cache=false], [min_overlap], [min_inscribed_circle_radius], [return_details], [sort_by_intersection_size]) [] indica argumentos opcionais
Argumentos	• **layer*** - a camada cuja sobreposição é verificada • **expression** - uma expressão opcional para avaliar as feições da camada alvo. Se não fornecida, a função irá retornar um booleano indicando que possui ao menos uma combinação. • **filter** - uma expressão opcional para filtrar as feições alvo a serem verificadas. Se não fornecida, todas as feições serão verificadas. • **limit** - um inteiro opcional para limitar o número de feições combinadas. Se não fornecido, todas as combinações serão retornadas. • **cache** - configure como verdadeiro para construir um índice espacial local (na maioria das vezes, isso é indesejado, a menos que esteja trabalhando em um provedor de dados particularmente lento) • **min_overlap** - define um filtro de exclusão opcional: – para polígonos, uma área mínima em unidades quadradas da feição atual para a interseção. Se a interseção resultar em vários polígonos a interseção será retornada se pelo menos um dos polígonos tiver uma área maior ou igual ao valor – para linhas, um comprimento mínimo em unidades da feição atual. Se a interseção resultar em várias linhas, a interseção será retornada se pelo menos uma das linhas tiver um comprimento maior ou igual ao valor. • **min_inscribed_circle_radius** - define um filtro de exclusão opcional (somente para polígonos): raio mínimo em unidades da feição atual para o círculo inscrito máximo da interseção. Se a interseção resultar em vários polígonos, a interseção será retornada se pelo menos um dos polígonos tiver um raio para o círculo máximo inscrito maior ou igual ao valor. Leia mais sobre predicado GEOS, como descrito na função PostGIS ST_MaximumInscribedCircle. Este argumento requer GEOS >= 3.9. • **return_details** - Set this to true to return a list of maps containing (key names in quotes) the feature 'id', the expression 'result' and the 'overlap' value (of the largest element in case of multipart). The 'radius' of the maximum inscribed circle is also returned when the target layer is a polygon. Only valid when used with the expression parameter • **sort_by_intersection_size** - somente válido quando utilizado em uma expressão, defina como 'des' para retornar os resultados ordenados pelo valor de sobreposição ou defina como 'asc' para ordenação ascendente.
Exemplos	• `overlay_intersects('regions')` → VERDADEIRO se a feição atual espacialmente fizer intercessão com uma região • `overlay_intersects('regions', filter:= population > 10000)` → VERDADEIRO se a feição atual espacialmente fizer intercessão com uma região com população maior que 10000 • `overlay_intersects('regions', name)` → um arrau de nomes, para as regiões interceptadas pela feição atual • `array_to_string(overlay_intersects('regions', name))` → uma lista separada por vírgula em string de nomes, para "regions" enterceptadas pela feição atual • `array_sort(overlay_intersects(layer:='regions', expression:="name", filter:= population > 10000))` → um array ordenada de "name" para "region" interceptadas pela feição atual e com "population" maior que 10000 • `overlay_intersects(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions intersected by the current feature • `overlay_intersects(layer:='regions', min_overlap:=0.54)` → VERDADEIRO se a feição atual espacialmente fizer intercessão com uma região e a área da intercessão (de pelo menos uma das partes no caso de multi poligonos) for maior ou igual a 0,54 • `overlay_intersects(layer:='regions', min_inscribed_circle_radius:=0.54)` → VERDADEIRO se a feição atual espacialmente fizer intercessão com uma região e o raio máximo do círculo inscrito da área de interseção (de pelo menos uma das partes no caso de multipartes) for maior ou igual a 0,54

Leitura adicional: *intersects*, *manipulação de array*, algoritmo *Selecionar pela localização*

overlay_nearest

Retorna se a feição atual tem feições de uma camada de destino dentro de uma determinada distância ou uma matriz de resultados baseados em expressões para as feições na camada de destino dentro de uma distância da feição atual.

Nota: Esta função pode ser lenta e consumir muita memória para camadas grandes.

Sintaxe	overlay_nearest(layer, [expression], [filter], [limit=1], [max_distance], [cache=false]) [] indica argumentos opcionais
Argumentos	• **layer** - a camada alvo • **expression** - uma expressão opcional para avaliar as feições da camada alvo. Se não fornecida, a função irá retornar um booleano indicando que possui ao menos uma combinação. • **filtro** - uma expressão opcional para filtrar as feições alvo a serem verificadas. Se não estiver definido, todas as feições da camada alvo serão usadas. • **limit** - um inteiro opcional para limitar o número de feições correspondentes. Se não for definido, apenas a feição mais próxima será retornada. Se definido como -1, retorna todas as feições correspondentes. • **max_distance** - uma distância opcional para limitar a busca de feições correspondentes. se não for preenchida, todas as feições na camada alvo serão utilizadas. • **cache** - configure como verdadeiro para construir um índice espacial local (na maioria das vezes, isso é indesejado, a menos que esteja trabalhando em um provedor de dados particularmente lento)
Exemplos	• `overlay_nearest('airports')` → VERDADEIRO se a camada "airports" possuir ao menos uma feição. • `overlay_nearest('airports', max_distance:= 5000)` → VERDADEIRO se houver um aeroporto dentro de uma distância de 5000 unidade do mapa da feição atual • `overlay_nearest('airports', name)` → "nome" do "airport" mais próximo à feição atual, como uma lista • `array_to_string(overlay_nearest('airports', name))` → nome do airport mais próximo à feição atual, como um texto. • `overlay_nearest(layer:='airports', expression:= name, max_distance:= 5000)` → nome do airport mais próximo dentro de uma distância de 5.000 unidades do mapa da feição atual, como uma lista • `overlay_nearest(layer:='airports', expression:="name", filter:= "Use"='Civilian', limit:=3)` → uma lista de nomes, para até três aeroportos civís próximos ordenados pela distância • `overlay_nearest(layer:='airports', expression:="name", limit:= -1, max_distance:= 5000)` → um a lista de nomes, de todos os aeroportos dentro de uma distância de 5000 unidades do mapa a partir da feição atual, ordenado pela distância

Leitura adicional: *manipulação de array*, algoritmo *Unir atributos por mais próximo*

overlay_touches

Retorna se a feição atual toca espacialmente pelo menos uma feição de uma camada de destino ou uma matriz de resultados baseados em expressão para as feições na camada de destino tocadas pela feição atual.

Leia mais sobre o predicado subjacente "Touches" do GEOS, conforme descrito na função ST_Touches do PostGIS.

Sintaxe	overlay_touches(layer, [expression], [filter], [limit], [cache=false]) [] indica argumentos opcionais
Argumentos	• **layer*** - a camada cuja sobreposição é verificada • **expression** - uma expressão opcional para avaliar as feições da camada alvo. Se não fornecida, a função irá retornar um booleano indicando que possui ao menos uma combinação. • **filter** - uma expressão opcional para filtrar as feições alvo a serem verificadas. Se não fornecida, todas as feições serão verificadas. • **limit** - um inteiro opcional para limitar o número de feições combinadas. Se não fornecido, todas as combinações serão retornadas. • **cache** - configure como verdadeiro para construir um índice espacial local (na maioria das vezes, isso é indesejado, a menos que esteja trabalhando em um provedor de dados particularmente lento)
Exemplos	• `overlay_touches('regions')` → VERDADEIRO se a feição atual espacialmente tocar uma região • `overlay_touches('regions', filter:= population > 10000)` → TRUE se a feição atual espacialmente tocar uma região com uma população maior que 10000 • `overlay_touches('regions', name)` → uma lista de nomes, das regiões tocadas pela feição atual. • `array_to_string(overlay_touches('regions', name))` → a string as a comma separated list of names, for the regions touched by the current feature • `array_sort(overlay_touches(layer:='regions', expression:="name", filter:= population > 10000))` → uma lista de nomes ordenados, das regiões tocadas pela feição atual e com população maior que 10000 • `overlay_touches(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions touched by the current feature

Leitura adicional: *touches*, *manipulação de array*, algoritmo *Selecionar pela localização*

overlay_within

Retorna se a feição atual está espacialmente dentro de pelo menos uma feição da camada de destino ou uma matriz de resultados baseados em expressão para as feições na camada de destino que contém a feição atual.

Leia mais sobre o predicado GEOS "dentro" subjacente, conforme descrito na função ST_Within do PostGIS.

Sintaxe	overlay_within(layer, [expression], [filter], [limit], [cache=false]) [] indica argumentos opcionais
Argumentos	• **layer*** - a camada cuja sobreposição é verificada • **expression** - uma expressão opcional para avaliar as feições da camada alvo. Se não fornecida, a função irá retornar um booleano indicando que possui ao menos uma combinação. • **filter** - uma expressão opcional para filtrar as feições alvo a serem verificadas. Se não fornecida, todas as feições serão verificadas. • **limit** - um inteiro opcional para limitar o número de feições combinadas. Se não fornecido, todas as combinações serão retornadas. • **cache** - configure como verdadeiro para construir um índice espacial local (na maioria das vezes, isso é indesejado, a menos que esteja trabalhando em um provedor de dados particularmente lento)
Exemplos	• `overlay_within('regions')` → VERDADEIRO se a feição atual for espacialmente contida por uma região • `overlay_within('regions', filter:= population > 10000)` → VERDADEIRO se a feição atual é espacialmente contida por uma região com uma população maior que 10000 • `overlay_within('regions', name)` → uma lista de nomes, das regiões que contém a feição atual • `array_to_string(overlay_within('regions', name))` → uma lista em texto separado por vírgula dos nomes das regiões que contém a feição atual • `array_sort(overlay_within(layer:='regions', expression:="name", filter:= population > 10000))` → uma lista ordenada de nomes das regiões que contém a feição atual e que possuem uma população maior que 10000 • `overlay_within(layer:='regions', expression:= geom_to_wkt(@geometry), limit:=2)` → an array of geometries (in WKT), for up to two regions containing the current feature

Leitura adicional: *within*, *manipulação de array*, algoritmo *Selecionar pela localização*

$perimeter

Retorna o comprimento do perímetro da feição atual. O perímetro calculado por esta função respeita a configuração do elipsóide do projeto atual e as configurações da unidade de distância. Por exemplo, se um elipsóide foi definido para o projeto, o perímetro calculado será elipsoidal e, se nenhum elipsóide for definido, o perímetro calculado será planimétrico.

Sintaxe	$perimeter
Exemplos	• `$perimeter` → 42

perimeter

Retorna o perímetro de um objeto de polígono geométrico. Os cálculos são sempre planimétricos no Spatial Reference System (SRS) dessa geometria, e as unidades do perímetro retornado corresponderão às unidades do SRS. Isso difere dos cálculos executados pela função $perimeter, que realizará cálculos elipsoidais com base nas configurações do elipsóide e da unidade de distância do projeto.

Sintaxe	perimeter(geometry)
Argumentos	• **geometry** - objeto de geometria de polígono
Exemplos	• `perimeter(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'))` → 12.0

point_n

Retorna um nó específico de uma geometria.

Sintaxe	point_n(geometry, index)
Argumentos	• **geometry** - objeto geométrico • **index** - índice do nó a ser retornado, onde 1 é o primeiro nó; se o valor for negativo, o índice do vértice selecionado será sua contagem total menos o valor absoluto
Exemplos	• `geom_to_wkt(point_n(geom_from_wkt('POLYGON((0 0, 4 0, 4 2, 0 2, 0 0))'),2))` → 'Point (4 0)'

Leitura adicional: algoritmo *Extrair vértices específicos*

point_on_surface

Retorna um ponto garantido estar na superfície de uma geometria.

Sintaxe	point_on_surface(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `point_on_surface(@geometry)` → a point geometry

Leitura adicional: algoritmo *Ponto na Superfície*

pole_of_inaccessibility

Calcula o pólo aproximado de inacessibilidade para uma superfície, que é o ponto interno mais distante do limite da superfície. Essa função usa o algoritmo 'polylabel' (Vladimir Agafonkin, 2016), que é uma abordagem iterativa garantida para encontrar o verdadeiro pólo de inacessibilidade dentro de uma tolerância especificada. Tolerâncias mais precisas requerem mais iterações e demoram mais para serem calculadas.

Sintaxe	pole_of_inaccessibility(geometry, tolerance)
Argumentos	• **geometry** - uma geometria • **tolerance** - distância máxima entre o ponto retornado e a localização do pólo verdadeiro
Exemplos	• `geom_to_wkt(pole_of_inaccessibility( geom_from_wkt('POLYGON((0 1, 0 9, 3 10, 3 3, 10 3, 10 1, 0 1))'), 0.1))` → 'Point(1.546875 2.546875)'

Figura9.26: Pólo de inacessibilidade

Leitura adicional: algoritmo *Pólo de inacessibilidade*

project

Retorna um ponto projetado de um ponto inicial usando uma distância, um rumo (azimute) e uma elevação em radianos.

Sintaxe	project(point, distance, azimuth, [elevation]) [] indica argumentos opcionais
Argumentos	• **point** - ponto inicial • **distance** - distância a projetar • **azimuth** - azimute em radianos no sentido horário, onde 0 corresponde ao norte • **elevation** - ângulo de inclinação em radianos
Exemplos	• `geom_to_wkt(project(make_point(1, 2), 3, radians(270)))` → 'Point(-2, 2)'

Leitura adicional: algoritmo *Projetar pontos (Cartesiano)*

relate

Testa a representação do modelo de interseção 9 estendido dimensional (DE-9IM) da relação entre duas geometrias.

Variante de relação

Retorna a representação Dimensional Extended 9 Intersection Model (DE-9IM) da relação entre duas geometrias.

Sintaxe	relate(geometry, geometry)
Argumentos	• **geometry** - uma geometria • **geometry** - uma geometria
Exemplos	• `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ) )` → 'FF1F00102'

Variante de correspondência de padrão

Testa se a relação DE-9IM entre duas geometrias corresponde a um padrão especificado.

Sintaxe	relate(geometry, geometry, pattern)
Argumentos	• **geometry** - uma geometria • **geometry** - uma geometria • **pattern** - DE-9IM padrão a combinar
Exemplos	• `relate( geom_from_wkt( 'LINESTRING(40 40,120 120)' ), geom_from_wkt( 'LINESTRING(40 40,60 120)' ), '**1F001**' )` → TRUE

reverse

Reverses the direction of a line string or reverses a string of text.

Variante string

Reverses the order of characters in a string.

Sintaxe	reverse(string)
Argumentos	• **string** - string to reverse
Exemplos	• `reverse('hello')` → 'olleh'

Variante geometria

Inverte a direção de uma cadeia de linha invertendo a ordem de seus vértices.

Sintaxe	reverse(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `geom_to_wkt(reverse(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'LINESTRING(2 2, 1 1, 0 0)'

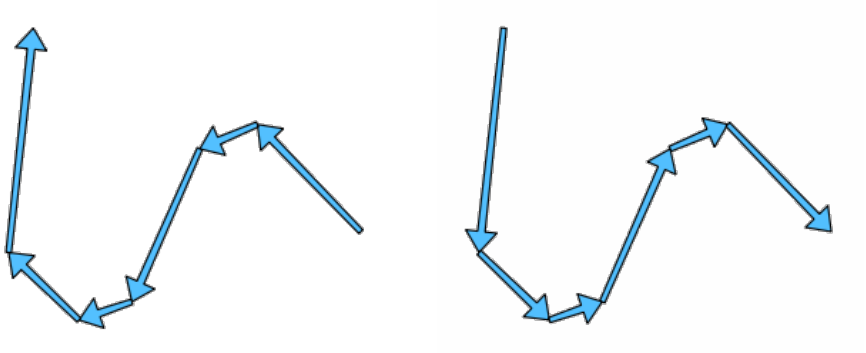


Figura9.27: Reversing line direction

Leitura adicional: algoritmo *Direção reversa da linha*

rotate

Retorna uma versão rotacionada de uma geometria. Os cálculos estão no Sistema de Referência Espacial desta geometria.

Sintaxe	rotate(geometry, rotation, [center=NULL], [per_part=false]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **rotation** - rotação no sentido horário em graus • **center** - ponto central de rotação. Se não for especificado, o centro da caixa delimitadora da geometria será usado. • **per_part** - aplicar rotação por parte. Se for verdadeiro, a rotação será aplicada em torno do centro do retângulo envolvente de cada peça quando a geometria de entrada for multipartes e um ponto central de rotação explícito não for especificado.
Exemplos	• `rotate(@geometry, 45, make_point(4, 5))` → geometry rotated 45 degrees clockwise around the (4, 5) point • `rotate(@geometry, 45)` → geometry rotated 45 degrees clockwise around the center of its bounding box

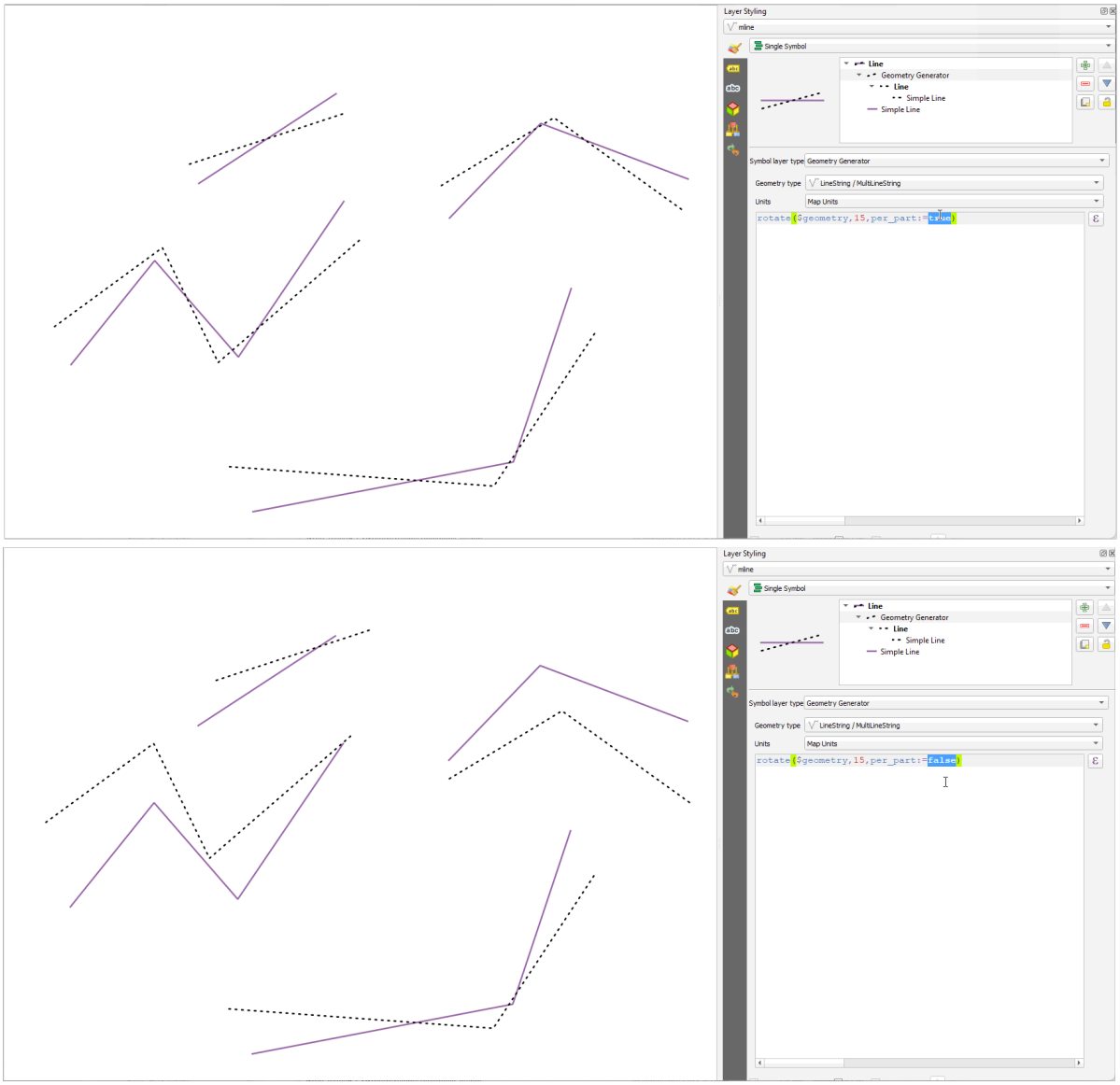


Figura9.28: Rotação de feições

roundness

Calculates how close a polygon shape is to a circle. The function returns 1 when the polygon shape is a perfect circle and 0 when it is completely flat.

Sintaxe	roundness(geometry)
Argumentos	• **geometry** - um polígono
Exemplos	• `round(roundness(geom_from_wkt('POLYGON(( 0 0, 0 1, 1 1, 1 0, 0 0))')), 3)` → 0.785 • `round(roundness(geom_from_wkt('POLYGON(( 0 0, 0 0.1, 1 0.1, 1 0, 0 0))')), 3)` → 0.260

Leitura adicional: algoritmo *Grau de arredondamento*

scale

Retorna uma versão em escala de uma geometria. Os cálculos estão no Sistema de Referência Espacial desta geometria.

Sintaxe	scale(geometry, x_scale, y_scale, [center]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **x_scale** - fator de escala do eixo x • **y_scale** - fator de escala do eixo y • **center** - ponto central de escala. Se não for especificado, será usado o centro do retângulo envolvente da geometria.
Exemplos	• `scale(@geometry, 2, 0.5, make_point(4, 5))` → geometry scaled twice horizontally and halved vertically, around the (4, 5) point • `scale(@geometry, 2, 0.5)` → geometry twice horizontally and halved vertically, around the center of its bounding box

segments_to_lines

Retorna uma geometria de várias linhas que consiste em uma linha para cada segmento na geometria de entrada.

Sintaxe	segments_to_lines(geometry)
Argumentos	• **geometry** - objeto geométrico
Exemplos	• `geom_to_wkt(segments_to_lines(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'MultiLineString ((0 0, 1 1),(1 1, 2 2))'

Leitura adicional: algoritmo *Explodir linhas*

shared_paths

Returns a collection containing paths shared by the two input geometries. Those going in the same direction are in the first element of the collection, those going in the opposite direction are in the second element. The paths themselves are given in the direction of the first geometry.

Sintaxe	shared_paths(geometry1, geometry2)
Argumentos	• **geometry1** - a LineString/MultiLineString geometry • **geometry2** - a LineString/MultiLineString geometry
Exemplos	• `geom_to_wkt(shared_paths(geom_from_wkt('MULTILINESTRING((26 125,26 200,126 200,126 125,26 125),(51 150,101 150,76 175,51 150))'),geom_from_wkt('LINESTRING(151 100,126 156.25,126 125,90 161, 76 175)')))` → 'GeometryCollection (MultiLineString ((126 156.25, 126 125),(101 150, 90 161),(90 161, 76 175)),MultiLineString EMPTY)' • `geom_to_wkt(shared_paths(geom_from_wkt('LINESTRING(76 175,90 161,126 125,126 156.25,151 100)'), geom_from_wkt('MULTILINESTRING((26 125,26 200,126 200,126 125,26 125),(51 150,101 150,76 175,51 150))')))` → 'GeometryCollection (MultiLineString EMPTY,MultiLineString ((76 175, 90 161),(90 161, 101 150),(126 125, 126 156.25)))'

shortest_line

Retorna a linha mais curta que une geometria1 a geometria2. A linha resultante começará em geometria1 e terminará em geometria2.

Sintaxe	shortest_line(geometry1, geometry2)
Argumentos	• **geometry1** - geometria para encontrar a linha mais curta de • **geometry2** - geometria para encontrar a linha mais curta para
Exemplos	• `geom_to_wkt(shortest_line(geom_from_wkt('LINESTRING (20 80, 98 190, 110 180, 50 75 )'),geom_from_wkt('POINT(100 100)')))` → 'LineString(73.0769 115.384, 100 100)'

simplify

Simplifica uma geometria removendo nós usando um limite baseado em distância (ou seja, o algoritmo de Douglas Peucker). O algoritmo preserva grandes desvios nas geometrias e reduz o número de vértices em segmentos quase retos.

Sintaxe	simplify(geometry, tolerance)
Argumentos	• **geometry** - uma geometria • **tolerance** - desvio máximo de segmentos retos para pontos a serem removidos
Exemplos	• `geom_to_wkt(simplify(geometry:=geom_from_wkt('LineString(0 0, 5 0.1, 10 0)'),tolerance:=5))` → 'LineString(0 0, 10 0)'

Figura9.29: From left to right, source layer and increasing simplification tolerances

Leitura adicional: algoritmo *Simplificar*

simplify_vw

Simplifica uma geometria removendo nós usando um limite baseado em área (ou seja, o algoritmo de Visvalingam-Whyatt). O algoritmo remove vértices que criam pequenas áreas em geometrias, por exemplo, picos estreitos ou segmentos quase retos.

Sintaxe	simplify_vw(geometry, tolerance)
Argumentos	• **geometry** - uma geometria • **tolerance** - uma medida da área máxima criada por um nó para que o nó seja removido
Exemplos	• `geom_to_wkt(simplify_vw(geometry:=geom_from_wkt('LineString(0 0, 5 0, 5.01 10, 5.02 0, 10 0)'),tolerance:=5))` → 'LineString(0 0, 10 0)'

Leitura adicional: algoritmo *Simplificar*

single_sided_buffer

Retorna uma geometria formada pelo buffer de apenas um lado de uma geometria linestring. As distâncias estão no Sistema de Referência Espacial desta geometria.

Sintaxe	single_sided_buffer(geometry, distance, [segments=8], [join=1], [miter_limit=2.0]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria (multi) linha • **distance** - distância do buffer. Os valores positivos serão armazenados à esquerda das linhas, os valores negativos à direita • **segments** - número de segmentos a serem usados para representar um quarto de círculo quando um estilo de junção arredondada é usado. Um número maior resulta em um buffer mais suave com mais nós. • **join** - estilo de junção dos cantos, onde 1 = round, 2 = miter e 3 = bevel • **miter_limit** - limite na proporção de esquadria usada para cantos muito afiados (ao usar apenas junções de esquadria)
Exemplos	• `single_sided_buffer(@geometry, 10.5)` → line buffered to the left by 10.5 units • `single_sided_buffer(@geometry, -10.5)` → line buffered to the right by 10.5 units • `single_sided_buffer(@geometry, 10.5, segments:=16, join:=1)` → line buffered to the left by 10.5 units, using more segments to result in a smoother buffer • `single_sided_buffer(@geometry, 10.5, join:=3)` → line buffered to the left by 10.5 units, using a beveled join

Figura9.30: Buffer do lado esquerdo versus direito na mesma camada de linha vetorial

Leitura adicional: algoritmo *Buffer de lado único*

sinuosity

Retorna a sinuosidade de uma curva, que é a razão entre o comprimento da curva e a distância reta (2D) entre suas extremidades.

Sintaxe	sinuosity(geometry)
Argumentos	• **geometry** - curva de entrada (circularstring, linestring)
Exemplos	• `round(sinuosity(geom_from_wkt('LINESTRING(2 0, 2 2, 3 2, 3 3)')), 3)` → 1.265 • `sinuosity(geom_from_wkt('LINESTRING( 3 1, 5 1)'))` → 1.0

smooth

Suaviza uma geometria adicionando nós extras que arredondam os cantos na geometria. Se as geometrias de entrada contiverem valores Z ou M, eles também serão suavizados e a geometria de saída manterá a mesma dimensionalidade da geometria de entrada.

Sintaxe	smooth(geometry, [iterations=1], [offset=0.25], [min_length=-1], [max_angle=180]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **iterations** - número de iterações de suavização a serem aplicadas. Números maiores resultam em geometrias mais suaves, porém mais complexas. • **offset** - valor entre 0 e 0,5, que controla o quanto a geometria suavizada segue a geometria original. Valores menores resultam em uma suavização mais apertada, valores maiores resultam em uma suavização mais frouxa. • **min_length** - comprimento mínimo de segmentos para aplicar a suavização. Este parâmetro pode ser usado para evitar colocar nós adicionais em excesso em segmentos mais curtos da geometria. • **max_angle** - ângulo máximo no nó para suavização a ser aplicada (0-180). Ao diminuir o ângulo máximo, cantos agudos intencionalmente na geometria podem ser preservados. Por exemplo, um valor de 80 graus manterá os ângulos retos na geometria.
Exemplos	• `geom_to_wkt(smooth(geometry:=geom_from_wkt('LineString(0 0, 5 0, 5 5)'),iterations:=1,offset:=0.2,min_length:=-1, max_angle:=180))` → 'LineString (0 0, 4 0, 5 1, 5 5)'

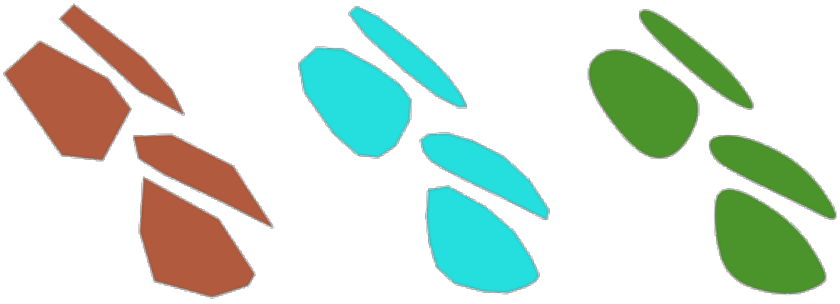


Figura9.31: Um número crescente de iterações terá como resultado geometrias mais suaves.

Leitura adicional: algoritmo *Suave*

square_wave

Constructs square/rectangular waves along the boundary of a geometry.

Sintaxe	square_wave(geometry, wavelength, amplitude, [strict=False]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **wavelength** - wavelength of square waveform • **amplitude** - amplitude of square waveform • **strict** - By default the wavelength argument is treated as a "maximum wavelength", where the actual wavelength will be dynamically adjusted so that an exact number of square waves are created along the boundaries of the geometry. If the strict argument is set to true then the wavelength will be used exactly and an incomplete pattern may be used for the final waveform.
Exemplos	• `square_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Square waves with wavelength 3 and amplitude 1 along the linestring

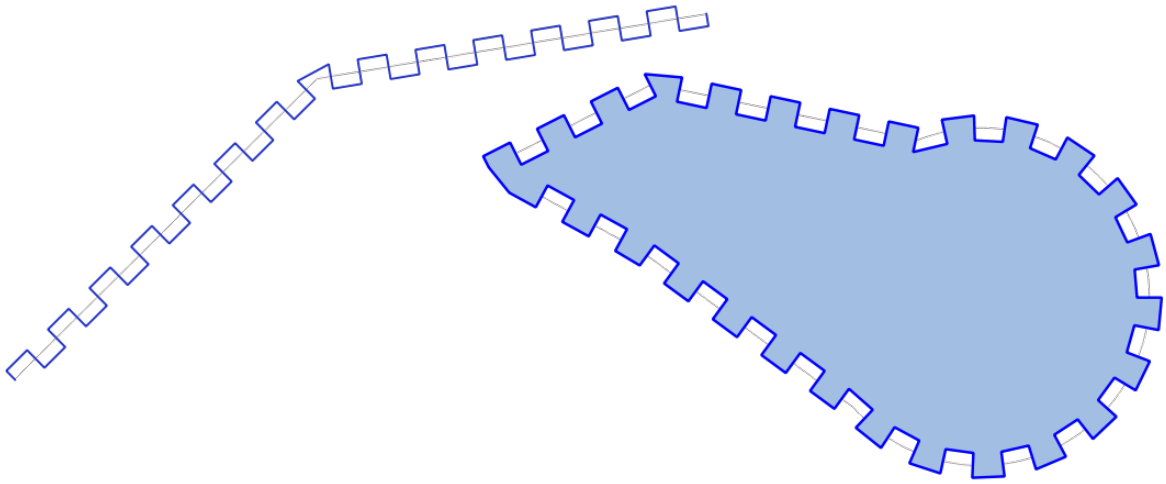


Figura9.32: Symbolizing features with square waves

square_wave_randomized

Constructs randomized square/rectangular waves along the boundary of a geometry.

Sintaxe	square_wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **min_wavelength** - minimum wavelength of waves • **max_wavelength** - maximum wavelength of waves • **min_amplitude** - minimum amplitude of waves • **max_amplitude** - maximum amplitude of waves • **seed** - specifies a random seed for generating waves. If the seed is 0, then a completely random set of waves will be generated.
Exemplos	• `square_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Randomly sized square waves with wavelengths between 2 and 3 and amplitudes between 0.1 and 0.2 along the linestring

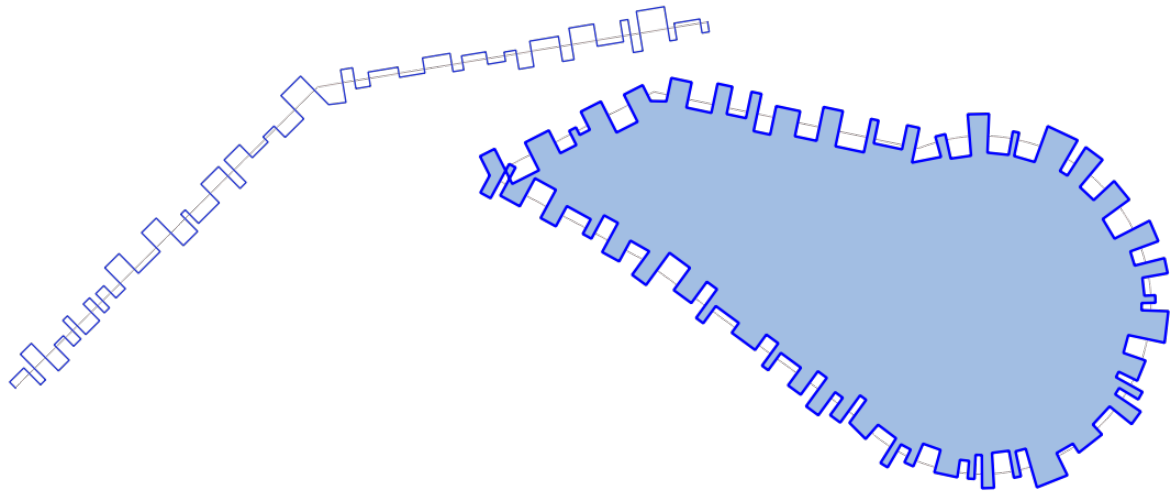


Figura9.33: Symbolizing features with square randomized waves

start_point

Retorna o primeiro nó de uma geometria.

Sintaxe	start_point(geometry)
Argumentos	• **geometry** - objeto geométrico
Exemplos	• `geom_to_wkt(start_point(geom_from_wkt('LINESTRING(4 0, 4 2, 0 2)')))` → 'Point (4 0)'

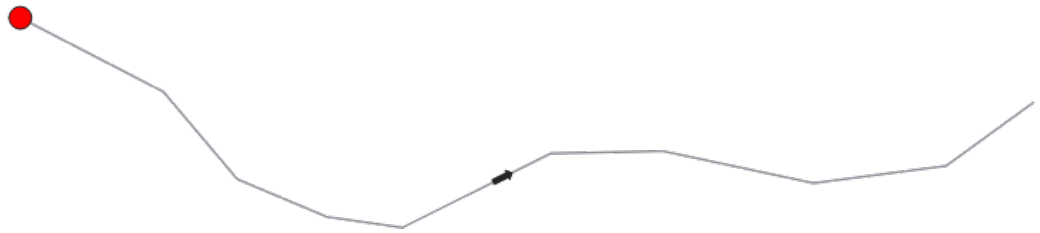


Figura9.34: Starting point of a line feature

Further reading: *end_point*, *Extrair vértices específicos* algorithm

straight_distance_2d

Retorna a distância direta/euclidiana entre o primeiro e o último vértice de uma geometria. A geometria deve ser uma curva (sequência circular, sequência de linhas).

Sintaxe	straight_distance_2d(geometry)
Argumentos	• **geometry** - A geometria.
Exemplos	• `straight_distance_2d(geom_from_wkt('LINESTRING(1 0, 1 1)'))` → 1 • `round(straight_distance_2d(geom_from_wkt('LINESTRING(1 4, 3 5, 5 0)')), 3)` → 5.657

Leitura adicional: *length*

sym_difference

Retorna uma geometria que representa as partes de duas geometrias que não se intersectam.

Sintaxe	sym_difference(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `geom_to_wkt( sym_difference( geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ), geom_from_wkt( 'LINESTRING(3 3, 8 8)' ) ) )` → 'LINESTRING(5 5, 8 8)'

Leitura adicional: algoritmo *Diferença simétrica*

tapered_buffer

Cria um buffer ao longo de uma geometria de linha onde o diâmetro do buffer varia uniformemente ao longo do comprimento da linha.

Sintaxe	tapered_buffer(geometry, start_width, end_width, [segments=8]) [] indica argumentos opcionais
Argumentos	• **geometry** - geometria de entrada. Deve ser uma geometria de (multi)linha. • **start_width** - largura do buffer no início da linha, • **end_width** - largura do buffer no final da linha. • **segments** - número de segmentos para aproximar as curvas de um quarto de círculo no buffer.
Exemplos	• `tapered_buffer(geometry:=geom_from_wkt('LINESTRING(1 2, 4 2)'),start_width:=1,end_width:=2,segments:=8)` → Um buffer cônico começando com um diâmetro de 1 e terminando com um diâmetro de 2 ao longo da geometria linestring.

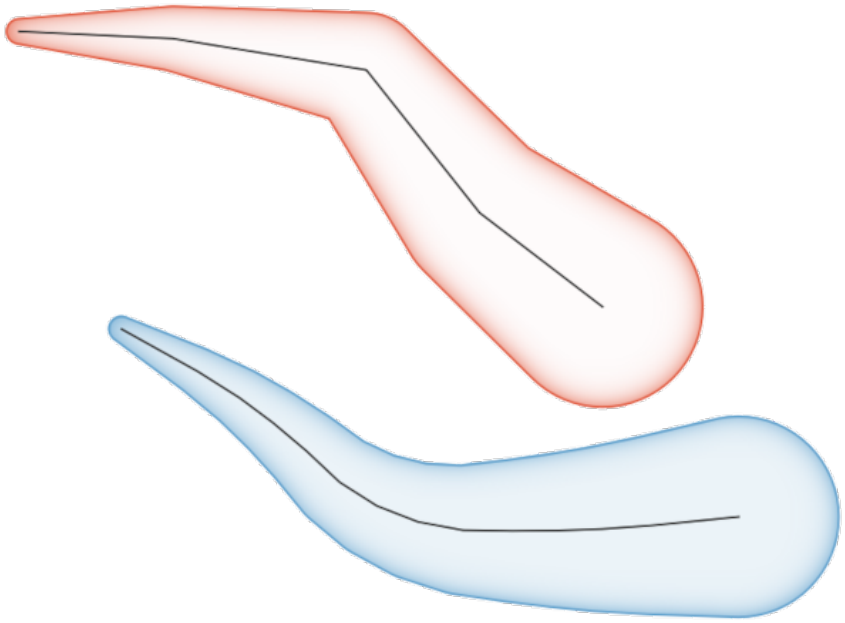


Figura9.35: Buffer cônico em feições de linha

Leitura adicional: algoritmo *Buffers cônicos.*

touches

Tests whether a geometry touches another. Returns TRUE if the geometries have at least one point in common, but their interiors do not intersect.

Sintaxe	touches(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `touches( geom_from_wkt( 'LINESTRING(5 3, 4 4)' ), geom_from_wkt( 'LINESTRING(3 3, 4 4, 5 5)' ) )` → TRUE • `touches( geom_from_wkt( 'POINT(4 4)' ), geom_from_wkt( 'POINT(5 5)' ) )` → FALSE

Leitura adicional: *overlay_touches*

transform

Retorna a geometria transformada de um CRS de origem para um CRS de destino.

Sintaxe	transform(geometry, source_auth_id, dest_auth_id)
Argumentos	• **geometry** - uma geometria • **source_auth_id** - the source CRS definition or CRS object • **dest_auth_id** - the destination CRS definition or CRS object
Exemplos	• `geom_to_wkt( transform( make_point(488995.53240249, 7104473.38600835), 'EPSG:2154', 'EPSG:4326' ) )` → 'POINT(0 51)'

Leitura adicional: algoritmo *Reprojetar camada*

translate

Retorna uma versão traduzida de uma geometria. Os cálculos estão no Sistema de Referência Espacial desta geometria.

Sintaxe	translate(geometry, dx, dy)
Argumentos	• **geometry** - uma geometria • **dx** - delta x • **dy** - delta y
Exemplos	• `translate(@geometry, 5, 10)` → a geometry of the same type like the original one

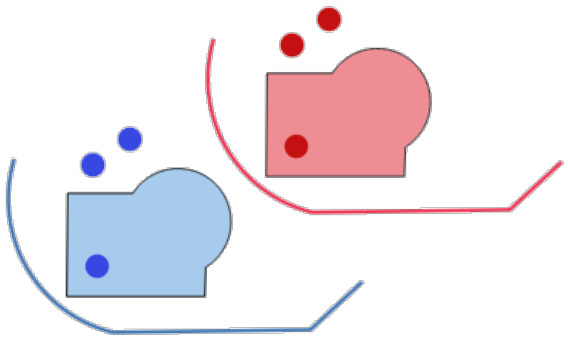


Figura9.36: Tradução de Feições

Leitura adicional: algoritmo *Transladar*

triangular_wave

Constructs triangular waves along the boundary of a geometry.

Sintaxe	triangular_wave(geometry, wavelength, amplitude, [strict=False]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **wavelength** - wavelength of triangular waveform • **amplitude** - amplitude of triangular waveform • **strict** - By default the wavelength argument is treated as a "maximum wavelength", where the actual wavelength will be dynamically adjusted so that an exact number of triangular waves are created along the boundaries of the geometry. If the strict argument is set to true then the wavelength will be used exactly and an incomplete pattern may be used for the final waveform.
Exemplos	• `triangular_wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Triangular waves with wavelength 3 and amplitude 1 along the linestring

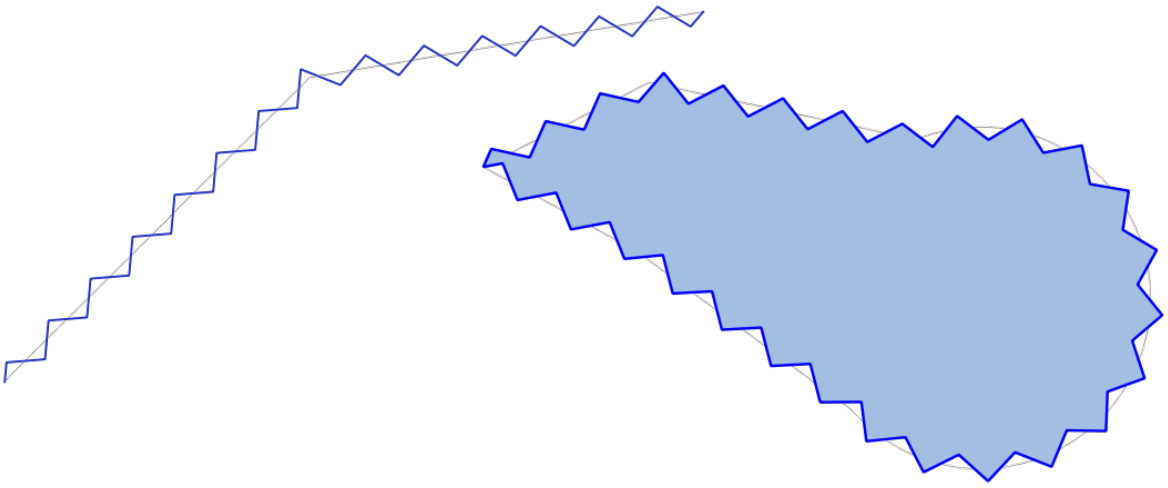


Figura9.37: Symbolizing features with triangular waves

triangular_wave_randomized

Constructs randomized triangular waves along the boundary of a geometry.

Sintaxe	triangular_wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **min_wavelength** - minimum wavelength of waves • **max_wavelength** - maximum wavelength of waves • **min_amplitude** - minimum amplitude of waves • **max_amplitude** - maximum amplitude of waves • **seed** - specifies a random seed for generating waves. If the seed is 0, then a completely random set of waves will be generated.
Exemplos	• `triangular_wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Randomly sized triangular waves with wavelengths between 2 and 3 and amplitudes between 0.1 and 0.2 along the linestring

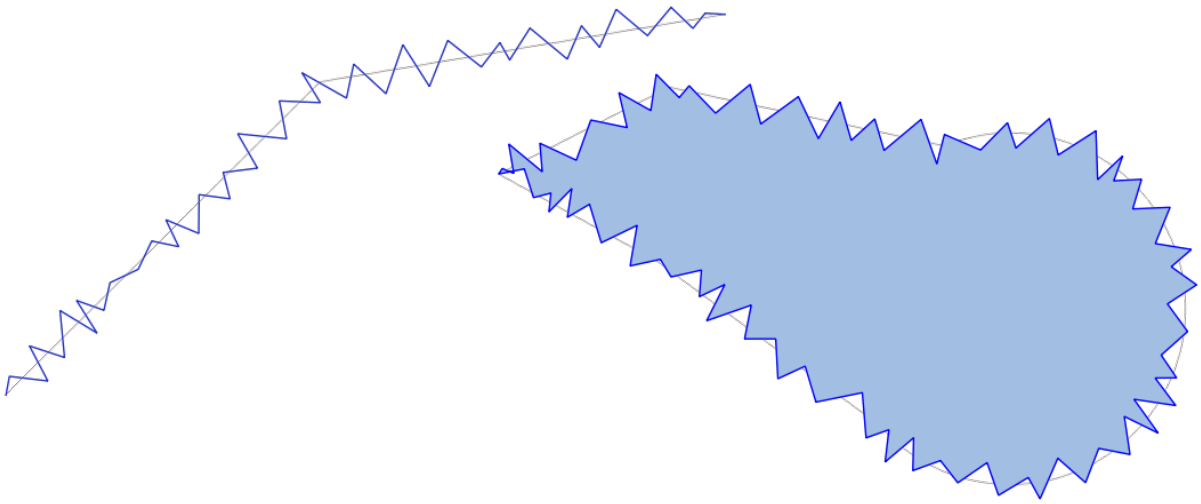


Figura9.38: Symbolizing features with triangular randomized waves

union

Retorna uma geometria que representa a união do conjunto de pontos das geometrias.

Sintaxe	union(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `geom_to_wkt( union( make_point(4, 4), make_point(5, 5) ) )` → 'MULTIPOINT(4 4, 5 5)'

wave

Constructs rounded (sine-like) waves along the boundary of a geometry.

Sintaxe	wave(geometry, wavelength, amplitude, [strict=False]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **wavelength** - wavelength of sine-like waveform • **amplitude** - amplitude of sine-like waveform • **strict** - By default the wavelength argument is treated as a "maximum wavelength", where the actual wavelength will be dynamically adjusted so that an exact number of waves are created along the boundaries of the geometry. If the strict argument is set to true then the wavelength will be used exactly and an incomplete pattern may be used for the final waveform.
Exemplos	• `wave(geom_from_wkt('LineString(0 0, 10 0)'), 3, 1)` → Sine-like waves with wavelength 3 and amplitude 1 along the linestring

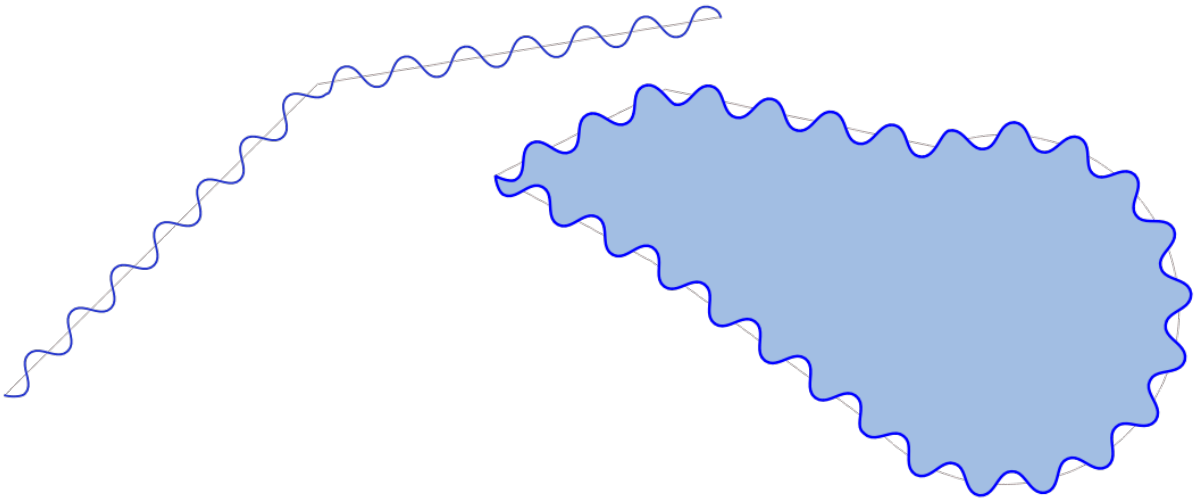


Figura9.39: Symbolizing features with waves

wave_randomized

Constructs randomized curved (sine-like) waves along the boundary of a geometry.

Sintaxe	wave_randomized(geometry, min_wavelength, max_wavelength, min_amplitude, max_amplitude, [seed=0]) [] indica argumentos opcionais
Argumentos	• **geometry** - uma geometria • **min_wavelength** - minimum wavelength of waves • **max_wavelength** - maximum wavelength of waves • **min_amplitude** - minimum amplitude of waves • **max_amplitude** - maximum amplitude of waves • **seed** - specifies a random seed for generating waves. If the seed is 0, then a completely random set of waves will be generated.
Exemplos	• `wave_randomized(geom_from_wkt('LineString(0 0, 10 0)'), 2, 3, 0.1, 0.2)` → Randomly sized curved waves with wavelengths between 2 and 3 and amplitudes between 0.1 and 0.2 along the linestring

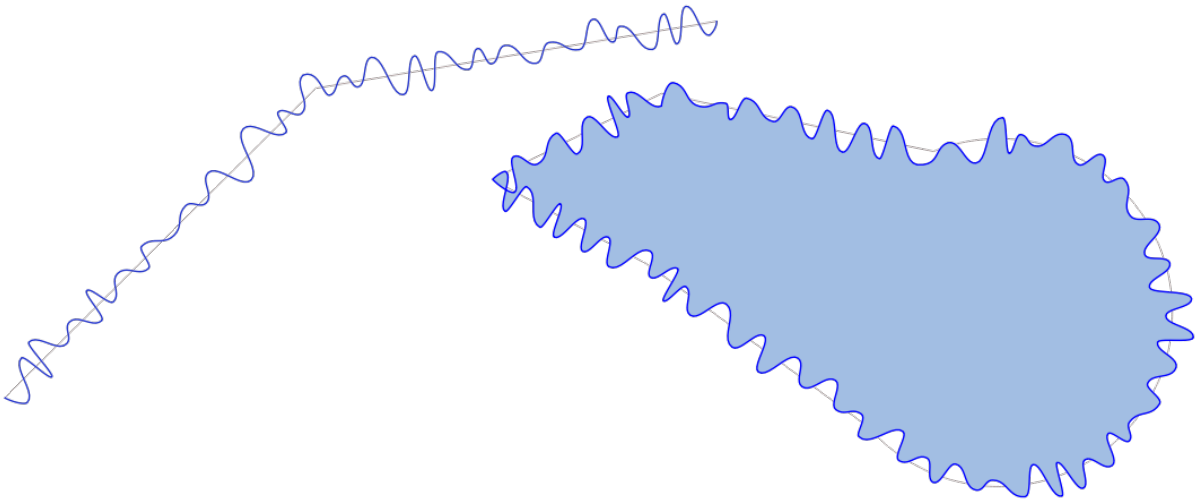


Figura9.40: Symbolizing features with randomized waves

wedge_buffer

Retorna um buffer em forma de cunha originário de uma geometria de ponto.

Sintaxe	wedge_buffer(center, azimuth, width, outer_radius, [inner_radius=0.0]) [] indica argumentos opcionais
Argumentos	• **center** - ponto central (origem) do buffer. Deve ser uma geometria de ponto. • **azimuth** - ângulo (em graus) para o meio da cunha apontar. • **width** - largura do buffer (em graus). Observe que a cunha se estenderá até a metade da largura angular de cada lado da direção do azimute. • **outer_radius** - raio externo para buffers • **inner_radius** - raio interno opcional para buffers
Exemplos	• `wedge_buffer(center:=geom_from_wkt('POINT(1 2)'), azimuth:=90,width:=180,outer_radius:=1)` → Um buffer em forma de cunha centrado no ponto (1,2), voltado para o leste, com largura de 180 graus e raio externo de 1.

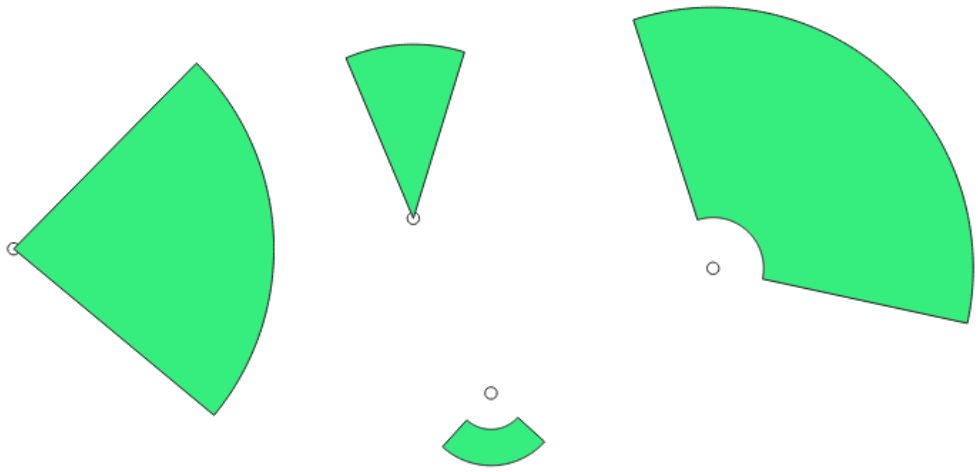


Figura9.41: Wedge buffering features

Leitura adicional: algoritmo *Crie buffers de cunha (wedge buffer).*

within

Tests whether a geometry is within another. Returns TRUE if the geometry1 is completely within geometry2.

Sintaxe	within(geometry1, geometry2)
Argumentos	• **geometry1** - uma geometria • **geometry2** - uma geometria
Exemplos	• `within( geom_from_wkt( 'POINT( 0.5 0.5)' ), geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))' ) )` → TRUE • `within( geom_from_wkt( 'POINT( 5 5 )' ), geom_from_wkt( 'POLYGON((0 0, 0 1, 1 1, 1 0, 0 0 ))' ) )` → FALSE

Leitura adicional: *overlay_within*

$x

Returns the x coordinate of the current point feature. If the feature is a multipoint feature, then the x-coordinate of the first point will be returned. **WARNING: This function is deprecated. It is recommended to use the replacement x() function with @geometry variable instead.**

Sintaxe	$x
Exemplos	• `$x` → 42

Further reading: *x*

x

Retorna a coordenada x de uma geometria de ponto ou a coordenada x do centroide para uma geometria sem ponto.

Sintaxe	x(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `x( geom_from_wkt( 'POINT(2 5)' ) )` → 2 • `x( @geometry )` → x coordinate of the current feature's centroid

$x_at

Retrieves a x coordinate of the current feature's geometry. **WARNING: This function is deprecated. It is recommended to use the replacement x_at function with @geometry variable instead.**

Sintaxe	$x_at(vertex)
Argumentos	• **vertex** - index of the vertex of the current geometry (indices start at 0; negative values apply from the last index, starting at -1)
Exemplos	• `$x_at(1)` → 5

Further reading: *x_at*

x_at

Retrieves a x coordinate of the geometry.

Sintaxe	x_at(geometry, vertex)
Argumentos	• **geometry** - objeto geométrico • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Exemplos	• `x_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 4

x_max

Retorna a coordenada x máxima de uma geometria. Os cálculos estão no sistema de referência espacial desta geometria.

Sintaxe	x_max(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `x_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 4

x_min

Retorna a coordenada x mínima de uma geometria. Os cálculos estão no sistema de referência espacial desta geometria.

Sintaxe	x_min(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `x_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 2

$y

Returns the y coordinate of the current point feature. If the feature is a multipoint feature, then the y-coordinate of the first point will be returned. **WARNING: This function is deprecated. It is recommended to use the replacement y() function with @geometry variable instead.**

Sintaxe	$y
Exemplos	• `$y` → 42

Further reading: *y*

y

Retorna a coordenada y de uma geometria de ponto ou a coordenada y do centroide para uma geometria sem ponto.

Sintaxe	y(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `y( geom_from_wkt( 'POINT(2 5)' ) )` → 5 • `y( @geometry )` → y coordinate of the current feature's centroid

$y_at

Retrieves a y coordinate of the current feature's geometry. **WARNING: This function is deprecated. It is recommended to use the replacement y_at function with @geometry variable instead.**

Sintaxe	$y_at(vertex)
Argumentos	• **vertex** - index of the vertex of the current geometry (indices start at 0; negative values apply from the last index, starting at -1)
Exemplos	• `$y_at(1)` → 2

Further reading: *y_at*

y_at

Retrieves a y coordinate of the geometry.

Sintaxe	y_at(geometry, vertex)
Argumentos	• **geometry** - objeto geométrico • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Exemplos	• `y_at( geom_from_wkt( 'POINT(4 5)' ), 0 )` → 5

y_max

Retorna a coordenada y máxima de uma geometria. Os cálculos estão no sistema de referência espacial desta geometria.

Sintaxe	y_max(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `y_max( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 8

y_min

Retorna a coordenada y mínima de uma geometria. Os cálculos estão no sistema de referência espacial desta geometria.

Sintaxe	y_min(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `y_min( geom_from_wkt( 'LINESTRING(2 5, 3 6, 4 8)') )` → 5

$z

Returns the z value of the current point feature if it is 3D. If the feature is a multipoint feature, then the z value of the first point will be returned. **WARNING: This function is deprecated. It is recommended to use the replacement z() function with @geometry variable instead.**

Sintaxe	$z
Exemplos	• `$z` → 123

z

Retorna a coordenada z de uma geometria de ponto, ou NULO se a geometria não tiver valor z.

Sintaxe	z(geometry)
Argumentos	• **geometry** - uma geometria de ponto
Exemplos	• `z( geom_from_wkt( 'POINTZ(2 5 7)' ) )` → 7

z_at

Retrieves a z coordinate of the geometry, or NULL if the geometry has no z value.

Sintaxe	z_at(geometry, vertex)
Argumentos	• **geometry** - objeto geométrico • **vertex** - index of the vertex of the geometry (indices start at 0; negative values apply from the last index, starting at -1)
Exemplos	• `z_at(geom_from_wkt('LineStringZ(0 0 0, 10 10 5, 10 10 0)'), 1)` → 5

z_max

Retorna a coordenada z máxima de uma geometria, ou NULO se a geometria não tiver valor z.

Sintaxe	z_max(geometry)
Argumentos	• **geometry** - uma geometria com coordenada z
Exemplos	• `z_max( geom_from_wkt( 'POINT ( 0 0 1 )' ) )` → 1 • `z_max( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 3 • `z_max( make_line( make_point( 0,0,0 ), make_point( -1,-1,-2 ) ) )` → 0 • `z_max( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → 2 • `z_max( geom_from_wkt( 'POINT ( 0 0 )' ) )` → NULL

z_min

Retorna a coordenada z mínima de uma geometria, ou NULO se a geometria não tiver valor z.

Sintaxe	z_min(geometry)
Argumentos	• **geometry** - uma geometria com coordenada z
Exemplos	• `z_min( geom_from_wkt( 'POINT ( 0 0 1 )' ) )` → 1 • `z_min( geom_from_wkt( 'MULTIPOINT ( 0 0 1 , 1 1 3 )' ) )` → 1 • `z_min( make_line( make_point( 0,0,0 ), make_point( -1,-1,-2 ) ) )` → -2 • `z_min( geom_from_wkt( 'LINESTRING( 0 0 0, 1 0 2, 1 1 -1 )' ) )` → -1 • `z_min( geom_from_wkt( 'POINT ( 0 0 )' ) )` → NULL

9.2.15 Funções de Layout

Este grupo contém funções para manipular as propriedades dos itens de layout de impressão.

item_variables

Retorna um mapa de variáveis de um item de layout dentro deste layout de impressão.

Sintaxe	item_variables(id)
Argumentos	• **id** - ID do item do layout
Exemplos	• `map_get( item_variables('Map 0'), 'map_scale')` → escala do item 'Map 0' no layout de impressão atual

Leitura adicional: Lista de *variáveis* padrão

map_credits

Returns a list of credit (usage rights) strings for the layers shown in a layout, or specific layout map item.

Sintaxe	map_credits([id], [include_layer_names=false], [layer_name_separator=': ']) [] indica argumentos opcionais
Argumentos	• **id** - Map item ID. If not specified, the layers from all maps in the layout will be used. • **include_layer_names** - Defina como verdadeiro para incluir os nomes das camadas antes de suas cadeias de crédito • **layer_name_separator** - String a ser inserida entre os nomes de camadas e seus créditos, se include_layer_names for verdadeiro
Exemplos	• `array_to_string( map_credits() )` → comma separated list of layer credits for all layers shown in all map items in the layout, e.g 'CC-BY-NC, CC-BY-SA' • `array_to_string( map_credits( 'Main Map' ) )` → lista de créditos das camadas separada por vírgula das camadas exibidas no item do layout "Main Map", E.x.: 'CC-BY-NC, CC-BY-SA' • `array_to_string( map_credits( 'Main Map', include_layer_names := true, layer_name_separator := ': ' ) )` → lista separada por vírgulas de nomes de camadas e seus créditos para camadas mostradas no item de layout 'Main Map', por exemplo 'Linhas ferroviárias: CC-BY-NC, Basemap: CC-BY-SA'

Esta função precisa que *Propriedades de metadados de acesso* das camadas tenham sido preenchidos.

9.2.16 Camadas do mapa

Este grupo contém a lista de camadas disponíveis no projeto atual e, para cada camada, seus campos (armazenados no grupo de dados, virtual ou auxiliares bem como os provenientes de uniões). Os campos podem ser interagidos da mesma maneira mencionada em *Campos e Valores*, exceto que um clique-duplo adicionará o nome como uma string (entre aspas simples) à expressão em vez de como uma referência de campo, visto que eles não pertencem à camada ativa. Isto oferece uma forma conveniente de escrever expressões referenciando diferentes camadas, com quando criamos *agregações*, *atributos* ou consultas *espaciais*.

Também fornece algumas funções convenientes para manipular camadas.

decode_uri

Pega uma camada e decodifica o uri do provedor de dados subjacente. Depende do provedor de dados, quais dados estão disponíveis.

Sintaxe	decode_uri(layer, [part]) [] indica argumentos opcionais
Argumentos	• **layer** - Camada a qual a uri deve ser decodificada. • **part** - Parte da uri a ser retornada. Se não for especificada, um mapa com todas as parts da url será retornado.
Exemplos	• `decode_uri(@layer)` → {'layerId': '0', 'layerName': '', 'path': '/home/qgis/shapefile.shp'} • `decode_uri(@layer)` → {'layerId': NULL, 'layerName': 'layer', 'path': '/home/qgis/geopackage.gpkg'} • `decode_uri(@layer, 'path')` → 'C:\my_data\qgis\shape.shp'

layer_property

Retorna uma propriedade de camada correspondente ou valor de metadados.

Sintaxe	layer_property(layer, property)
Argumentos	• **layer** - uma string, representando um nome de camada ou um ID de camada • **property** - uma string correspondente à propriedade a ser retornada. As opções válidas são: – name: nome da camada – ID: ID da camada – title: string titulo do metadado – abstract: string do metadado resumo – keywords: metadado palavras-chave – data_url: metadado URL – attribution: string do metadado atribuição – attribution_url: metadado URL atribuição – source: camada origem – min_scale: escala de exibição mínima para camada – max_scale: escala de exibição máxima para camada – is_editable: se a camada estiver no modo de edição – crs: camada SRC – crs_definition: definição completa do SRC da camada – crs_description: descrição do SRC da camada – crs_ellipsoid: acronym of the layer CRS ellipsoid – extent: extensão da camada (como um objeto de geometria) – ditance_units: unidades de distância da camada – type: tipo de camada, ex., Vetorial ou Raster – storage_type: formato de armazenamento (apenas camadas vetoriais) – geometry_type: tipo da geometria, e.x.:, Point (apenas camadas vetoriais) – feature_count: contagem aproximada de feição da camada (apenas camadas vetoriais) – path: caminho para o arquivo fonte da camada. Disponível apenas para camadas baseadas em arquivos.
Exemplos	• `layer_property('streets','title')` → 'Basemap Streets' • `layer_property('airports','feature_count')` → 120 • `layer_property('landsat','crs')` → 'EPSG:4326'

Leitura adicional: propriedades de camada *vector*, *raster* e *mesh*

load_layer

Loads a layer by source URI and provider name.

Sintaxe	load_layer(uri, provider)
Argumentos	• **uri** - layer source URI string • **provider** - layer data provider name
Exemplos	• `layer_property(load_layer('c:/data/roads.shp', 'ogr'), 'feature_count')` → count of features from the c:/data/roads.shp vector layer

9.2.17 Funções de mapeamento

Este grupo contém funções para criar ou manipular chaves e valores de estruturas de dados de mapas (também conhecidos como objetos de dicionário, pares de chave-valor ou arrays associativos). Ao contrário de *estrutura de dados lista* em que a ordem dos valores importa, a ordem dos pares de chave-valor no objeto de mapa não é relevante e os valores são identificados por suas chaves.

from_json

Carrega um texto formatado como JSON

Sintaxe	from_json(string)
Argumentos	• **string** - string JSON
Exemplos	• `from_json('{"1":"one","2":"two"}')` → { '1': 'one', '2': 'two' } • `from_json('[1,2,3]')` → [1,2,3]

hstore_to_map

Cria um mapa a parir de uma string formatada como hstore

Sintaxe	hstore_to_map(string)
Argumentos	• **string** - string de entrada
Exemplos	• `hstore_to_map('qgis=>rocks')` → { 'qgis': 'rocks' }

map

Retorna um mapa contendo todas as chaves e valores passados como par de parâmetros.

Sintaxe	map(key1, value1, key2, value2, ...)
Argumentos	• **key** - uma chave (string) • **valor** - um valor
Exemplos	• `map('1','one','2', 'two')` → { '1': 'one', '2': 'two' } • `map('1','one','2', 'two')['1']` → 'one'

map_akeys

Retorna um array contendo todas as chaves de um mapa

Sintaxe	map_akeys(map)
Argumentos	• **map** - um mapa
Exemplos	• `map_akeys(map('1','one','2','two'))` → ['1', '2']

map_avals

Retorna todos os valores de um map como um array.

Sintaxe	map_avals(map)
Argumentos	• **map** - um mapa
Exemplos	• `map_avals(map('1','one','2','two'))` → ['one', 'two']

map_concat

Retorna um mapa contendo todas as entradas dos mapas fornecidos. Se dois mapas contiverem a mesma chave, o valor do segundo mapa será usado.

Sintaxe	map_concat(map1, map2, …)
Argumentos	• **map** - um mapa
Exemplos	• `map_concat(map('1','one', '2','overridden'),map('2','two', '3','three'))` → { '1': 'one', '2': 'two', '3': 'three' }

map_delete

Retorna um mapa com a chave fornecida e seu valor correspondente excluído.

Sintaxe	map_delete(map, key)
Argumentos	• **map** - um mapa • **key** - a chave a ser excluída
Exemplos	• `map_delete(map('1','one','2','two'),'2')` → { '1': 'one' }

map_exist

Returns TRUE if the given key exists in the map.

Sintaxe	map_exist(map, key)
Argumentos	• **map** - um mapa • **key** - chave a ser pesquisada
Exemplos	• `map_exist(map('1','one','2','two'),'3')` → FALSE

map_get

Retorna o valor de um mapa, dada sua chave. Retorna NULO se a chave não existir.

Sintaxe	map_get(map, key)
Argumentos	• **map** - um mapa • **key** - chave a ser pesquisada
Exemplos	• `map_get(map('1','one','2','two'),'2')` → 'two' • `map_get( item_variables('Map 0'), 'map_scale')` → escala do item 'Map 0' (se existir) no layout de impressão atual

Dica

Você pode utilizar também o *operador de índice ([])* para pegar um valor de um mapa.

map_insert

Retorna um mapa com uma chave/valor adicionado. Se a chave já existir, seu valor será substituído.

Sintaxe	map_insert(map, key, value)
Argumentos	• **map** - um mapa • **key** - a chave a ser adicionada • **value** - o valor a ser adicionado
Exemplos	• `map_insert(map('1','one'),'3','three')` → { '1': 'one', '3': 'three' } • `map_insert(map('1','one','2','overridden'),'2','two')` → { '1': 'one', '2': 'two' }

map_prefix_keys

Returns a map with all keys prefixed by a given string.

Sintaxe	map_prefix_keys(map, prefix)
Argumentos	• **map** - um mapa • **prefix** - a string
Exemplos	• `map_prefix_keys(map('1','one','2','two'), 'prefix-')` → { 'prefix-1': 'one', 'prefix-2': 'two' }

map_to_hstore

Mescla os elementos de um mapa em uma string formatada em hstore.

Sintaxe	map_to_hstore(map)
Argumentos	• **map** - mapa de entrada
Exemplos	• `map_to_hstore(map('1','one','2','two'))` → '"1"=>"one","2"=>"two"'

map_to_html_dl

Merge map elements into a HTML definition list string.

Sintaxe	map_to_html_dl(map)
Argumentos	• **map** - mapa de entrada
Exemplos	• `map_to_html_dl(map('1','one','2','two'))` → <dl><dt>1</dt><dd>one</dd><dt>2</dt><dd>two</dd></dl>

map_to_html_table

Merge map elements into a HTML table string.

Sintaxe	map_to_html_table(map)
Argumentos	• **map** - mapa de entrada
Exemplos	• `map_to_html_table(map('1','one','2','two'))` → <table><thead><tr><th>1</th><th>2</th></tr></thead><tbody><tr><td>one</td><td>two</td></tr></tbody

to_json

Cria uma string formada em JSON a partir de um mapa, matriz ou outro valor.

Sintaxe	to_json(value)
Argumentos	• **value** - O valor de entrada
Exemplos	• `to_json(map('1','one','2','two'))` → {“1”:”one”,”2”:”two”} • `to_json(array(1,2,3))` → [1,2,3]

url_encode

Returns an URL encoded string from a map. Transforms all characters in their properly-encoded form producing a fully-compliant query string.

Note that the plus sign ‘+’ is not converted.

Sintaxe	url_encode(map)
Argumentos	• **map** - a map.
Exemplos	• `url_encode(map('a&+b', 'a and plus b', 'a=b', 'a equals b'))` → ‘a%26+b=a%20and%20plus%20b&a%3Db=a%20equals%20b’

9.2.18 Funções Matemáticas

Este grupo contém funções matemáticas (por exemplo, raiz quadrada, sen e cos).

abs

Retorna o valor absoluto de um número.

Sintaxe	abs(value)
Argumentos	• **value** - um número
Exemplos	• `abs(-2)` → 2

acos

Retorna o cosseno inverso de um valor em radianos.

Sintaxe	acos(value)
Argumentos	• **value** - cosseno de um ângulo em radianos
Exemplos	• `acos(0.5)` → 1.0471975511966

asin

Retorna o seno inverso de um valor em radianos.

Sintaxe	asin(value)
Argumentos	• **value** - seno de um ângulo em radianos
Exemplos	• `asen(1.0)` → 1.5707963267949

atan

Retorna a tangente inversa de um valor em radianos.

Sintaxe	atan(value)
Argumentos	• **value** - tangente de um ângulo em radianos
Exemplos	• `atan(0.5)` → 0.463647609000806

atan2

Retorna a tangente inversa de dy/dx usando os sinais dos dois argumentos para determinar o quadrante do resultado.

Sintaxe	atan2(dy, dx)
Argumentos	• **dy** - diferença na coordenada y • **dx** - diferença na coordenada x
Exemplos	• `atan2(1.0, 1.732)` → 0.523611477769969

ceil

Arredonda um número para cima.

Sintaxe	ceil(value)
Argumentos	• **value** - um número
Exemplos	• `ceil(4.9)` → 5 • `ceil(-4.9)` → -4

clamp

Restringe um valor de entrada a um intervalo especificado.

Sintaxe	clamp(minimum, input, maximum)
Argumentos	• **minimum** - o menor valor que *input* pode assumir • **input** - um valor que será restrito ao intervalo especificado por *minimun* e *maximun* • **maximum** - o maior valor que *input* pode assumir
Exemplos	• `clamp(1,5,10)` → 5 *input* está entre 1 e 10, então retorna inalterada • `clamp(1,0,10)` → 1 *input* é menor que o valor mínimo de 1, então a função retorna 1 • `clamp(1,11,10)` → 10 se o valor de *input* for maior que o valor máximo de 10, então a função retornará 10

cos

Retorna o cosseno de um ângulo.

Sintaxe	cos(angle)
Argumentos	• **angle** - ângulo em radianos
Exemplos	• `cos(1.571)` → 0.000796326710733263

degrees

Converte de radianos para graus.

Sintaxe	degrees(radians)
Argumentos	• **radians** - valor numérico
Exemplos	• `degrees(3.14159)` → 180 • `degrees(1)` → 57.2958

exp

Retorna o exponencial de um valor.

Sintaxe	exp(value)
Argumentos	• **value** - número para retornar o expoente
Exemplos	• `exp(1.0)` → 2.71828182845905

floor

Arredonda um número para baixo.

Sintaxe	floor(value)
Argumentos	• **value** - um número
Exemplos	• `floor(4.9)` → 4 • `floor(-4.9)` → -5

ln

Retorna o logaritmo natural de um valor.

Sintaxe	ln(value)
Argumentos	• **value** - valor numérico
Exemplos	• `ln(1)` → 0 • `ln(2.7182818284590452354)` → 1

log

Retorna o valor do logaritmo do valor passado e da base.

Sintaxe	log(base, value)
Argumentos	• **base** - qualquer número positivo • **value** - qualquer número positivo
Exemplos	• `log(2, 32)` → 5 • `log(0.5, 32)` → -5

log10

Retorna o valor do logaritmo de base 10 da expressão passada.

Sintaxe	log10(value)
Argumentos	• **value** - qualquer número positivo
Exemplos	• `log10(1)` → 0 • `log10(100)` → 2

max

Retorna o maior valor em um conjunto de valores.

Sintaxe	max(value1, value2, …)
Argumentos	• **value** - um número
Exemplos	• `max(2,10.2,5.5)` → 10.2 • `max(20.5,NULL,6.2)` → 20.5

min

Retorna o menor valor em um conjunto de valores.

Sintaxe	min(value1, value2, …)
Argumentos	• **value** - um número
Exemplos	• `min(20.5,10,6.2)` → 6.2 • `min(2,-10.3,NULL)` → -10.3

pi

Retorna o valor de pi para cálculos.

Sintaxe	pi()
Exemplos	• `pi()` → 3.14159265358979

radians

Converte de graus para radianos.

Sintaxe	radians(degrees)
Argumentos	• **degrees** - valor numérico
Exemplos	• `radians(180)` → 3.14159 • `radians(57.2958)` → 1

rand

Retorna um inteiro aleatório dentro do intervalo especificado pelo argumento mínimo e máximo (inclusive). Se uma seed for fornecida, o retorno será sempre o mesmo, dependendo da seed.

Sintaxe	rand(min, max, [seed=NULL]) [] indica argumentos opcionais
Argumentos	• **min** - um número inteiro que representa o menor número aleatório possível desejado • **max** - um inteiro que representa o maior número aleatório possível desejado • **seed** - qualquer valor usado como seed
Exemplos	• `rand(1, 10)` → 8

randf

Retorna um float aleatório dentro do intervalo especificado pelo argumento mínimo e máximo (inclusive). Se uma seed for fornecida, o retorno será sempre o mesmo, dependendo da seed.

Sintaxe	randf([min=0.0], [max=1.0], [seed=NULL]) [] indica argumentos opcionais
Argumentos	• **min** - um float representando o menor número aleatório possível desejado • **max** - um float representando o maior número aleatório possível desejado • **seed** - qualquer valor usado como seed
Exemplos	• `randf(1, 10)` → 4.59258286403147

round

Arredonda um número para número de casas decimais.

Sintaxe	round(value, [places=0]) [] indica argumentos opcionais
Argumentos	• **value** - número decimal a ser arredondado • **places** - Número inteiro opcional que representa o número de casas para arredondar os decimais. Pode ser negativo.
Exemplos	• `round(1234.567, 2)` → 1234.57 • `round(1234.567)` → 1235 • `round(1234.567, -1)` → 1230

scale_exponential

Transforma um determinado valor de um domínio de entrada para um intervalo de saída usando uma curva exponencial. Esta função pode ser usada para atenuar valores dentro ou fora do intervalo de saída especificado.

Sintaxe	scale_exponential(value, domain_min, domain_max, range_min, range_max, exponent)
Argumentos	• **value** - Um valor no domínio de entrada. A função retornará um valor escalado correspondente no intervalo de saída. • **domain_min** - Especifica o valor mínimo no domínio de entrada, o menor valor que o valor de entrada deve assumir. • **domain_max** - Especifica o valor maior no domínio de entrada, o maior valor que o valor de entrada deve assumir. • **range_min** - Especifica um valor mínimo no intervalo de saída, o menor valor que a função deve retornar. • **range_max** - Especifica o valor máximo do intervalor de saída, o maior valor que a função deverá retornar. • **exponent** - Um valor positivo (maior que 0), que determina a forma como os valores de entrada são mapeados para o intervalo de saída. Grandes expoentes farão com que os valores de saída 'acelerem', iniciando lentamente antes de acelerar à medida que os valores de entrada se aproximam do máximo do domínio. Expoentes menores (menos de 1) farão com que os valores de saída 'desacelerem', onde o mapeamento começa rapidamente, mas diminui conforme se aproxima do máximo do domínio.
Exemplos	• `scale_exponential(5,0,10,0,100,2)` → 3.030 facilitando o ln, usando um expoente de 2 • `scale_exponential(3,0,10,0,100,0.5)` → 87.585 easing out, usando um expoente de 0.5

scale_linear

Transforma um determinado valor de um domínio de entrada em um intervalo de saída usando interpolação linear.

Sintaxe	scale_linear(value, domain_min, domain_max, range_min, range_max)
Argumentos	• **value** - Um valor no domínio de entrada. A função retornará um valor escalado correspondente no intervalo de saída. • **domain_min** - Especifica o valor mínimo no domínio de entrada, o menor valor que o valor de entrada deve assumir. • **domain_max** - Especifica o valor maior no domínio de entrada, o maior valor que o valor de entrada deve assumir. • **range_min** - Especifica um valor mínimo no intervalo de saída, o menor valor que a função deve retornar. • **range_max** - Especifica o valor máximo do intervalor de saída, o maior valor que a função deverá retornar.
Exemplos	• `scale_linear(5,0,10,0,100)` → 50 • `scale_linear(0.2,0,1,0,360)` → 72 dimensionar um valor entre 0 e 1 para um ângulo entre 0 e 360 • `scale_linear(1500,1000,10000,9,20)` → 9.6111111 escalar uma população que varia entre 1000 e 10000 para um tamanho de fonte entre 9 e 20

scale_polynomial

Transforms a given value from an input domain to an output range using a polynomial curve. This function can be used to ease values in or out of the specified output range.

Sintaxe	scale_polynomial(value, domain_min, domain_max, range_min, range_max, exponent)
Argumentos	• **value** - Um valor no domínio de entrada. A função retornará um valor escalado correspondente no intervalo de saída. • **domain_min** - Especifica o valor mínimo no domínio de entrada, o menor valor que o valor de entrada deve assumir. • **domain_max** - Especifica o valor maior no domínio de entrada, o maior valor que o valor de entrada deve assumir. • **range_min** - Especifica um valor mínimo no intervalo de saída, o menor valor que a função deve retornar. • **range_max** - Especifica o valor máximo do intervalor de saída, o maior valor que a função deverá retornar. • **exponent** - Um valor positivo (maior que 0), que determina a forma como os valores de entrada são mapeados para o intervalo de saída. Grandes expoentes farão com que os valores de saída 'acelerem', iniciando lentamente antes de acelerar à medida que os valores de entrada se aproximam do máximo do domínio. Expoentes menores (menos de 1) farão com que os valores de saída 'desacelerem', onde o mapeamento começa rapidamente, mas diminui conforme se aproxima do máximo do domínio.
Exemplos	• `scale_polynomial(5,0,10,0,100,2)` → 25 facilitando o ln, usando um expoente de 2 • `scale_polynomial(3,0,10,0,100,0.5)` → 54.772 easing out, usando um expoente de 0.5

sin

Retorna o seno de um ângulo.

Sintaxe	sin(angle)
Argumentos	• **angle** - ângulo em radianos
Exemplos	• `sin(1.571)` → 0.999999682931835

sqrt

Retorna a raiz quadrada de um valor.

Sintaxe	sqrt(value)
Argumentos	• **value** - um número
Exemplos	• `sqrt(9)` → 3

tan

Retorna a tangente de um ângulo.

Sintaxe	tan(angle)
Argumentos	• **angle** - ângulo em radianos
Exemplos	• `tan(1.0)` → 1.5574077246549

9.2.19 Funções de Malhas

Este grupo contém funções que calculam ou retornam valores relacionados à mesh.

$area_face

Retorna a área da face da malha atual. A área calculada por esta função respeita tanto a configuração do elipsóide do projeto atual quanto as configurações da unidade de área. Por exemplo, se um elipsóide foi definido para o projeto, a área calculada será elipsoidal, e se nenhum elipsóide for definido, a área calculada será planimétrica.

Sintaxe	$area_face
Exemplos	• `$face_area` → 42

$face_index

Retorna o índice da face da malha atual.

Sintaxe	$face_index
Exemplos	• `$face_index` → 4581

$vertex_as_point

Retorna o vértice atual como uma geometria de ponto.

Sintaxe	$vertex_as_point
Exemplos	• `geom_to_wkt( $vertex_as_point )` → 'POINT(800 1500 41)'

$vertex_index

Retorna o índice do vértice do mesh atual

Sintaxe	$vertex_index
Exemplos	• `$vertex_index` → 9874

$vertex_x

Retorna a coordenada X do vértice da malha atual.

Sintaxe	$vertex_x
Exemplos	• `$vertex_x` → 42.12

$vertex_y

Retorna a coordenada Y do vértice da malha atual.

Sintaxe	$vertex_y
Exemplos	• `$vertex_y` → 12.24

$vertex_z

Retorna o valor Z do vértice do mesh atual.

Sintaxe	$vertex_z
Exemplos	• `$vertex_z` → 42

9.2.20 Operadores

Este grupo contém operadores (por exemplo, +, -, *). Observe que, para a maioria das funções matemáticas abaixo, se uma das entradas for NULL, o resultado será NULL.

%

Remainder of division. Takes the sign of the dividend.

Sintaxe	a % b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `9 % 2` → 1 • `9 % -2` → 1 • `-9 % 2` → -1 • `5 % NULO` → NULO

*

Multiplicação de dois valores

Sintaxe	a * b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 * 4` → 20 • `5 * NULO` → NULO

+

Adição de dois valores. Se um dos valores for NULO, o resultado será NULO.

Sintaxe	a + b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 + 4` → 9 • `5 + NULO` → NULO • `'QGIS ' + 'ROCKS'` → 'QGIS ROCKS' • `to_datetime('2020-08-01 12:00:00') + '1 day 2 hours'` → 2020-08-02T14:00:00

Leitura adicional: *concat*, ||

-

Subtração de dois valores. Se um dos valores for NULO, o resultado será NULO.

Sintaxe	a - b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 - 4` → 1 • `5 - NULO` → NULO • `to_datetime('2012-05-05 12:00:00') - to_interval('1 day 2 hours')` → 2012-05-04T10:00:00

/

Divisão de dois valores

Sintaxe	a / b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 / 4` → 1.25 • *5 / NULO`* → NULO

//

Floor division of two values

Sintaxe	a // b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `9 // 2` → 4

<

Compara dois valores e avalia como 1 se o valor da esquerda for menor que o valor da direita.

Sintaxe	a < b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 < 4` → FALSE • `5 < 5` → FALSE • `4 < 5` → TRUE

<=

Compara dois valores e avalia como 1 se o valor da esquerda for menor ou igual ao valor da direita.

Sintaxe	a <= b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 <= 4` → FALSE • `5 <= 5` → TRUE • `4 <= 5` → TRUE

<>

Compara dois valores e avalia com 1 se eles não forem iguais.

Sintaxe	a <> b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 <> 4` → TRUE • `4 <> 4` → FALSE • `5 <> NULO` → NULO • `NULO <> NULO` → NULO

=

Compara dois valores e avalia com 1 se forem iguais.

Sintaxe	a = b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 = 4` → FALSE • `4 = 4` → TRUE • `5 = NULO` → NULO • `NULO = NULO` → NULO

>

Compara dois valores e avalia como 1 se o valor esquerdo for maior que o valor direito.

Sintaxe	a > b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 > 4` → TRUE • `5 > 5` → FALSE • `4 > 5` → FALSE

>=

Compara dois valores e avalia como 1 se o valor da esquerda for maior ou igual ao valor da direita.

Sintaxe	a >= b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 >= 4` → TRUE • `5 >= 5` → TRUE • `4 >= 5` → FALSE

AND

Returns TRUE when conditions a and b are true.

Sintaxe	a AND b
Argumentos	• **a** - condição • **b** - condição
Exemplos	• `TRUE AND TRUE` → TRUE • `TRUE AND FALSE` → FALSE • `4 = 2+2 AND 1 = 1` → TRUE • `4 = 2+2 AND 1 = 2` → FALSE

BETWEEN

Returns TRUE if value is within the specified range. The range is considered inclusive of the bounds. To test for exclusion NOT BETWEEN can be used.

Sintaxe	value BETWEEN lower_bound AND higher_bound
Argumentos	• **value** - the value to compare with a range. It can be a string, a number or a date. • **lower_bound AND higher_bound** - range bounds
Exemplos	• `'B' BETWEEN 'A' AND 'C'` → TRUE • `2 BETWEEN 1 AND 3` → TRUE • `2 BETWEEN 2 AND 3` → TRUE • `'B' BETWEEN 'a' AND 'c'` → FALSE • `lower('B') BETWEEN 'a' AND 'b'` → TRUE

Nota

value BETWEEN lower_bound AND higher_bound is the same as "*value >= lower_bound AND value <= higher_bound*".

Leitura adicional: *NOT BETWEEN*

ILIKE

Returns TRUE if the first parameter matches case-insensitive the supplied pattern. LIKE can be used instead of ILIKE to make the match case-sensitive. Works with numbers also.

Sintaxe	string/número padrão ILIKE
Argumentos	• **string/number** - string para pesquisar • **pattern** - padrão a ser procurado, você pode usar '%' como curinga, '_' como um único caractere e '\\' para escapar caracteres especiais.
Exemplos	• `'A' ILIKE 'A'` → TRUE • `'A' ILIKE 'a'` → TRUE • `'A' ILIKE 'B'` → FALSE • `'ABC' ILIKE 'b'` → FALSE • `'ABC' ILIKE 'B'` → FALSE • `'ABC' ILIKE '_b_'` → TRUE • `'ABC' ILIKE '_B_'` → TRUE • `'ABCD' ILIKE '_b_'` → FALSE • `'ABCD' ILIKE '_B_'` → FALSE • `'ABCD' ILIKE '_b%'` → TRUE • `'ABCD' ILIKE '_B%'` → TRUE • `'ABCD' ILIKE '%b%'` → TRUE • `'ABCD' ILIKE '%B%'` → TRUE • `'ABCD%' ILIKE 'abcd\\%'` → TRUE • `'ABCD' ILIKE '%B\\%'` → FALSE

IN

Returns TRUE if value is found within a list of values.

Sintaxe	a IN b
Argumentos	• **a** - valor • **b** - lista de valores
Exemplos	• `'A' IN ('A','B')` → TRUE • `'A' IN ('C','B')` → FALSE

IS

Returns TRUE if a is the same as b.

Sintaxe	a IS b
Argumentos	• **a** - qualquer valor • **b** - qualquer valor
Exemplos	• `'A' IS 'A'` → TRUE • `'A' IS 'a'` → FALSE • `4 IS 4` → TRUE • `4 IS 2+2` → TRUE • `4 IS 2` → FALSE • `@geometry IS NULL` → 0, if your geometry is not NULL

IS NOT

Returns TRUE if a is not the same as b.

Sintaxe	a IS NOT b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `'a' IS NOT 'b'` → TRUE • `'a' IS NOT 'a'` → FALSE • `4 IS NOT 2+2` → FALSE

LIKE

Returns TRUE if the first parameter matches the supplied pattern. Works with numbers also.

Sintaxe	texto/número padrão LIKE
Argumentos	• **string/number** - valor • **pattern** - padrão a ser comparado com o valor, você pode usar '%' como curinga, '_' como caractere único e '\\' para escapar esses caracteres especiais.
Exemplos	• `'A' LIKE 'A'` → TRUE • `'A' LIKE 'a'` → FALSE • `'A' LIKE 'B'` → FALSE • `'ABC' LIKE 'B'` → FALSE • `'ABC' LIKE '_B_'` → TRUE • `'ABCD' LIKE '_B_'` → FALSE • `'ABCD' LIKE '_B%'` → TRUE • `'ABCD' LIKE '%B%'` → TRUE • `'1%' LIKE '1\\%'` → TRUE • `'1_' LIKE '1\\%'` → FALSE

NOT

Nega uma condição.

Sintaxe	NOT a
Argumentos	• **a** - condição
Exemplos	• `NOT 1` → FALSE • `NOT 0` → TRUE

NOT BETWEEN

Returns TRUE if value is not within the specified range. The range is considered inclusive of the bounds.

Sintaxe	value NOT BETWEEN lower_bound AND higher_bound
Argumentos	• **value** - the value to compare with a range. It can be a string, a number or a date. • **lower_bound AND higher_bound** - range bounds
Exemplos	• `'B' NOT BETWEEN 'A' AND 'C'` → FALSE • `1.0 NOT BETWEEN 1.1 AND 1.2` → TRUE • `2 NOT BETWEEN 2 AND 3` → FALSE • `'B' NOT BETWEEN 'a' AND 'c'` → TRUE • `lower('B') NOT BETWEEN 'a' AND 'b'` → FALSE

Nota

value NOT BETWEEN lower_bound AND higher_bound is the same as "*value < lower_bound OR value > higher_bound*".

Leitura adicional: *BETWEEN*

OR

Returns TRUE when condition a or b is true.

Sintaxe	a OR b
Argumentos	• **a** - condição • **b** - condição
Exemplos	• `4 = 2+2 OR 1 = 1` → TRUE • `4 = 2+2 OR 1 = 2` → TRUE • `4 = 2 OR 1 = 2` → FALSE

[]

Operador de índice. Retorna um elemento de uma matriz ou valor de mapa.

Sintaxe	[index]
Argumentos	• **index** - índice do array ou chave de um valor num mapa
Exemplos	• `array(1,2,3)[0]` → 1 • `array(1,2,3)[2]` → 3 • `array(1,2,3)[-1]` → 3 • `map('a',1,'b',2)['a']` → 1 • `map('a',1,'b',2)['b']` → 2

Leitura adicional: *array_get*, *map_get*

^

Potência de dois valores.

Sintaxe	a ^ b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `5 ^ 4` → 625 • `5 ^ NULO` → NULO

||

Une dois valores em uma string.

Se um dos valores for NULL o resultado será NULL. Veja a função CONCAT para um comportamento diferente

Sintaxe	a || b
Argumentos	• **a** - valor • **b** - valor
Exemplos	• `'Aqui' || ' e ' || 'ali'` → 'Aqui e ali' • `'Nothing' || NULL` → NULL • `'Dia: ' || "Diameter"` → 'Dia: 25' • `1 || 2` → '12'

Leitura adicional: *concat*, *+*

~

Executa uma correspondência de expressão regular em um valor de string. Os caracteres de barra invertida devem ter escape duplo (por exemplo, "\\s" para corresponder a um caractere de espaço em branco).

Sintaxe	string ~ regex
Argumentos	• **string** - Um valor de string • **regex** - Uma expressão regular. As barras devem ser escapadas, e.x.: \\d.
Exemplos	• `'hello' ~ 'll'` → TRUE • `'hello' ~ '^ll'` → FALSE • `'hello' ~ 'llo$'` → TRUE • `'abc123' ~ '\\d+'` → TRUE

Leitura adicional: *regexp_match*

9.2.21 Funções de Processamento

Este grupo contém funções que operam em algoritmos de processamento.

parameter

Retorna o valor de um parâmetro de entrada do algoritmo de processamento.

Sintaxe	parameter(name)
Argumentos	• **name** - nome do parâmetro de entrada correspondente
Exemplos	• `parameter('BUFFER_SIZE')` → 5.6

9.2.22 Funções de Rasters

Este grupo contém funções para operar na camada raster.

raster_attributes

Returns a map with the fields names as keys and the raster attribute table values as values from the attribute table entry that matches the given raster value.

Sintaxe	raster_attributes(layer, band, value)
Argumentos	• **layer** - o nome ou id de uma camada raster • **band** - the band number for the associated attribute table lookup. • **value** - raster value
Exemplos	• `raster_attributes('vegetation', 1, raster_value('vegetation', 1, make_point(1,1)))` → {'class': 'Vegetated', 'subclass': 'Trees'} • `raster_attributes('vegetation', 1, raster_value('vegetation', 1, @layer_cursor_point))` → {'class': 'Vegetated', 'subclass': 'Trees'}

raster_statistic

Retorna as estatísticas de uma camada raster.

Sintaxe	raster_statistic(layer, band, property)
Argumentos	• **camada** - uma string, representando um nome de camada raster ou um ID de camada • **band** - inteiro representando o número da banda da camada raster, começando em 1 • **property** - uma string correspondente à propriedade a ser retornada. As opções válidas são: – min: valor mínimo – max: valor máximo – avg: valor médio (média) – stdev: desvio padrão dos valores – range: intervalo de valores (max. - min.) – sum: soma de todos os valores do raster
Exemplos	• `raster_statistic('lc',1,'avg')` → Valor médio da banda 1 da camada raster 'lc' • `raster_statistic('ac2010',3,'min')` → Menor valor da banda 3 da camada raster '2010'

raster_value

Retorna o valor raster encontrado no ponto fornecido.

Sintaxe	raster_value(layer, band, point)
Argumentos	• **layer** - o nome ou id de uma camada raster • **banda** - o número da banda para amostrar o valor. • **point** - geometria de ponto (para geometrias multipartes com mais de uma parte, um valor NULO será retornado)
Exemplos	• `raster_value('dem', 1, make_point(1,1))` → 25 • `raster_value('ndvi', 2, @layer_cursor_point)` → 25

9.2.23 Funções de Registro e Atributos

Este grupo contém funções que operam na identificação da registros.

attribute

Retorna um atributo de uma feição.

** Variante 1**

Retorna o valor de um atributo da feição atual.

Sintaxe	attribute(attribute_name)
Argumentos	• **attribute_name** - nome do atributo a ser retornado
Exemplos	• `attribute( 'nome' )` → valor armazenado no atributo 'nome' para a feição atual

Variante 2

Permite que a feição de destino e o nome do atributo sejam especificados.

Sintaxe	attribute(feature, attribute_name)
Argumentos	• **feature** - uma feição • **attribute_name** - nome do atributo a ser retornado
Exemplos	• `attribute( @atlas_feature, 'name' )` → valor armazenado no atributo 'name' na feição atual do atlas

attributes

Retorna um mapa contendo todos os atributos de uma feição, onde os nomes de campo são as chaves do mapa

** Variante 1**

Retorna um mapa de todos os atributos da feição atual.

Sintaxe	attributes()
Exemplos	• `attributes()['name']` → valor armazenado no atributo 'name' da feição atual

Variante 2

Permite que a feição de destino seja especificada.

Sintaxe	attributes(feature)
Argumentos	• **feature** - uma feição
Exemplos	• `attributes( @atlas_feature )['name']` → valor armazenado no atributo 'name' na feição atual do do atlas

Leitura adicional: *Funções de mapeamento*

$currentfeature

Returns the current feature being evaluated. This can be used with the 'attribute' function to evaluate attribute values from the current feature. **WARNING: This function is deprecated. It is recommended to use the replacement @feature variable instead.**

Sintaxe	$currentfeature
Exemplos	• `attribute( $currentfeature, 'name' )` → valor armazenado no atributo 'name' da feição atual

display_expression

Retorna a expressão de exibição para uma determinada feição em uma camada. A expressão é avaliada por padrão. Pode ser usado com zero, um ou mais argumentos, veja abaixo para detalhes.

Sem parâmetros

Se for chamada sem nenhum parâmetro, a função irá avaliar a expressão de exibição da feição atual na camada atual.

Sintaxe	display_expression()
Exemplos	• `display_expression()` → Expressão de exibição da feição atual na camada atual.

Um parâmetro 'feição'

Se chamada apenas com um parâmetro 'feição', a função avaliará a feição especificada da camada atual.

Sintaxe	display_expression(feature)
Argumentos	• **feature** - Feição que deverá ser avaliada.
Exemplos	• `display_expression(@atlas_feature)` → Expressão de exibição da feição atual do atlas.

Parâmetros de camada e feição

Se a função for chamada com uma camada e uma feição, ela avaliará a feição especificada da camada especificada.

Sintaxe	display_expression(layer, feature, [evaluate=true]) [] indica argumentos opcionais
Argumentos	• **layer** - A camada (ou ID ou nome) • **feature** - Feição que deverá ser avaliada. • **evaluate** - Se a expressão deve ser avaliada. Se false, a expressão será retornada apenas como uma string literal (que pode ser avaliada posteriormente usando a função 'eval').
Exemplos	• `display_expression( 'streets', get_feature_by_id('streets', 1))` → Expressão de exibição da feição com o ID 1 na camada 'streets'. • `display_expression('a_layer_id', @feature, 'False')` → The display expression of the given feature not evaluated.

feature_id

Returns a feature's unique ID, or NULL if the feature is not valid.

Sintaxe	feature_id(feature)
Argumentos	• **feature** - um objeto feição
Exemplos	• `feature_id( @feature )` → the ID of the current feature

Further reading: *get_feature_by_id*

get_feature

Retorna a primeia feição de uma camada correspondente a um determinado valor de atributo.

Single value variant

Along with the layer ID, a single column and value are specified.

Sintaxe	get_feature(layer, attribute, value)
Argumentos	• **layer** - nome ou ID da camada • **attribute** - attribute name to use for the match • **value** - valor do atributo a ser correspondido
Exemplos	• `get_feature('streets','name','main st')` → primeira feição encontrada na camada "streets" com o valor "main st" no campo "name"

Variação por mapeamento

Along with the layer ID, a map containing the columns (key) and their respective value to be used.

Sintaxe	get_feature(layer, attribute)
Argumentos	• **layer** - nome ou ID da camada • **attribute** - Map containing the column and value pairs to use
Exemplos	• `get_feature('streets',map('name','main st','lane_num','4'))` → first feature found in "streets" layer with "main st" value in the "name" field and "4" value in the "lane_num" field

get_feature_by_id

Retorna a feição com um id em uma camada.

Sintaxe	get_feature_by_id(layer, feature_id)
Argumentos	• **layer** - camada, nome da camada ou ID da camada • **feature_id** - id da feição que será retornada
Exemplos	• `get_feature_by_id('streets', 1)` → feição de id 1 na camada "streets"

Further reading: *feature_id*

$id

Returns the feature id of the current row. **WARNING: This function is deprecated. It is recommended to use the replacement @id variable instead.**

Sintaxe	$id
Exemplos	• `$id` → 42

Further reading: *feature_id*, *get_feature_by_id*

is_attribute_valid

Returns TRUE if a specific feature attribute meets all constraints.

Sintaxe	is_attribute_valid(attribute, [feature], [layer], [strength]) [] indica argumentos opcionais
Argumentos	• **attribute** - an attribute name • **feature** - A feature. If not set, the current feature will be used. • **layer** - A vector layer. If not set, the current layer will be used. • **strength** - Set to 'hard' or 'soft' to narrow down to a specific constraint type. If not set, the function will return FALSE if either a hard or a soft constraint fails.
Exemplos	• `is_attribute_valid('HECTARES')` → TRUE if the current feature's value in the "HECTARES" field meets all constraints. • `is_attribute_valid('HOUSES',get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE if the value in the "HOUSES" field from the feature with "FID"=10 in 'my_layer' fails to meet all constraints.

Further reading: *Restrições*

is_feature_valid

Returns TRUE if a feature meets all field constraints.

Sintaxe	is_feature_valid([feature], [layer], [strength]) [] indica argumentos opcionais
Argumentos	• **feature** - A feature. If not set, the current feature will be used. • **layer** - A vector layer. If not set, the current layer will be used. • **strength** - Set to 'hard' or 'soft' to narrow down to a specific constraint type. If not set, the function will return FALSE if either a hard or a soft constraint fails.
Exemplos	• `is_feature_valid(strength:='hard')` → TRUE if all fields from the current feature meet their hard constraints. • `is_feature_valid(get_feature('my_layer', 'FID', 10), 'my_layer')` → FALSE if all fields from feature with "FID"=10 in 'my_layer' fails to meet all constraints.

Further reading: *Restrições*

is_selected

Returns TRUE if a feature is selected. Can be used with zero, one or two arguments, see below for details.

Sem parâmetros

If called with no parameters, the function will return TRUE if the current feature in the current layer is selected.

Sintaxe	is_selected()
Exemplos	• `is_selected()` → TRUE if the current feature in the current layer is selected.

Um parâmetro 'feição'

If called with a 'feature' parameter only, the function returns TRUE if the specified feature from the current layer is selected.

Sintaxe	is_selected(feature)
Argumentos	• **feature** - A feição que deve ser verificada para seleção.
Exemplos	• `is_selected(@atlas_feature)` → TRUE if the current atlas feature is selected. • `is_selected(get_feature('streets', 'name', 'Main St.'))` → TRUE if the unique named "Main St." feature on the active "streets" layer is selected. • `is_selected(get_feature_by_id('streets', 1))` → TRUE if the feature with the id 1 on the active "streets" layer is selected.

Dois parâmetros

If the function is called with both a layer and a feature, it will return TRUE if the specified feature from the specified layer is selected.

Sintaxe	is_selected(layer, feature)
Argumentos	• **layer** - A camada (seu ID ou nome) na qual a seleção será verificada. • **feature** - A feição que deve ser verificada para seleção.
Exemplos	• `is_selected( 'streets', get_feature('streets', 'name', "street_name"))` → TRUE if the current building's street is selected (assuming the building layer has a field named 'street_name' and the 'streets' layer has a field called 'name' with unique values). • `is_selected( 'streets', get_feature_by_id('streets', 1))` → TRUE if the feature with the id 1 on the "streets" layer is selected.

maptip

Retorna o maptip de uma feição de uma camada. A expressão é avaliada por padrão. Pode ser usada com zero, um ou mais argumentos, veja abaixo para mais detalhes.

Sem parâmetros

Se chamada sem parâmetros, a função avaliará a dica de mapa da feição atual na camada atual.

Sintaxe	maptip()
Exemplos	• `maptip()` → Maptip da feição atual da camada atual.

Um parâmetro 'feição'

Se chamada apenas com um parâmetro 'feição', a função avaliará a feição especificada da camada atual.

Sintaxe	maptip(feature)
Argumentos	• **feature** - Feição que deverá ser avaliada.
Exemplos	• `maptip(@atlas_feature)` → Maptip da feição atual da feição do atlas.

Parâmetros de camada e feição

Se a função for chamada com uma camada e uma feição, ela avaliará a feição especificada da camada especificada.

Sintaxe	maptip(layer, feature, [evaluate=true]) [] indica argumentos opcionais
Argumentos	• **layer** - A camada (ou ID ou nome) • **feature** - Feição que deverá ser avaliada. • **evaluate** - Se a expressão precisa ser avaliada, Se falso, a expressão será retornada como um texto literal (que pode ser potencialmente avaliada futuramente utilizando a função 'eval_template')
Exemplos	• `maptip('streets', get_feature_by_id('streets', 1))` → Maptip da feição de ID 1 na camada 'streets'. • `maptip('a_layer_id', @feature, 'False')` → The maptip of the given feature not evaluated.

num_selected

Retorna o número de feições selecionadas em uma determinada camada. Por padrão, funciona na camada na qual a expressão é avaliada.

Sintaxe	num_selected([layer=current layer]) [] indica argumentos opcionais
Argumentos	• **layer** - Camada (ou seu id ou nome) onde a seleção deverá ser verificada.
Exemplos	• `num_selected()` → O número de feições selecionadas na feição atual. • `num_selected('streets')` → Número de feições selecionadas na camada 'streets'

represent_attributes

Returns a map with the attribute names as keys and the configured representation values as values. The representation value for the attributes depends on the configured widget type for each attribute. Can be used with zero, one or more arguments, see below for details.

Sem parâmetros

If called with no parameters, the function will return the representation of the attributes of the current feature in the current layer.

Sintaxe	represent_attributes()
Exemplos	• `represent_attributes()` → The representation of the attributes for the current feature.

Um parâmetro 'feição'

If called with a 'feature' parameter only, the function will return the representation of the attributes of the specified feature from the current layer.

Sintaxe	represent_attributes(feature)
Argumentos	• **feature** - Feição que deverá ser avaliada.
Exemplos	• `represent_attributes(@atlas_feature)` → The representation of the attributes for the specified feature from the current layer.

Parâmetros de camada e feição

If called with a 'layer' and a 'feature' parameter, the function will return the representation of the attributes of the specified feature from the specified layer.

Sintaxe	represent_attributes(layer, feature)
Argumentos	• **layer** - The layer (or its ID or name). • **feature** - Feição que deverá ser avaliada.
Exemplos	• `represent_attributes('atlas_layer', @atlas_feature)` → The representation of the attributes for the specified feature from the specified layer.

Leitura adicional: *represent_value*

represent_value

Retorna o valor de representação configurado para um valor de campo. Depende do tipo de widget configurado. Frequentemente, isso é útil para widgets de 'Mapa de valores'.

Sintaxe	represent_value(value, [fieldName]) [] indica argumentos opcionais
Argumentos	• **value** - O valor que deve ser resolvido. Provavelmente um campo. • **fieldName** - The field name for which the widget configuration should be loaded.
Exemplos	• `represent_value("field_with_value_map")` → Descrição do valor • `represent_value('static value', 'field_name')` → Descrição do valor estático

Leitura adicional: *widget types*, *represent_attributes*

sqlite_fetch_and_increment

Gerencia valores auto incrementais em bancos de dados sqlite

Valores SQLIte padrões podem apenas serem aplicados durante a inserção e não pré-buscados.

Isso torna impossível adquirir uma chave primária incrementada via AUTO_INCREMENT antes de criar a linha no banco de dados. Observação: com postgres, isso funciona por meio da opção *avaliar valores padrão*.

Ao adicionar novas feições com relações, é muito bom poder adicionar filhos para um pai, enquanto o formulário dos pais ainda está aberto e equanto a feição pai não estiver commitada.

Para contornar essa limitação, esta função pode ser usada para gerenciar valores de sequência em uma tabela separada em formatos baseados em sqlite como gpkg.

A tabela de sequência será filtrada para um id de sequência (filter_attribute e filter_value) e o valor atual do id_field será incrementado em 1 e o valor incrementado retornado.

Se colunas adicionais exigirem a especificação de valores, o mapa default_values poderá ser usado para essa finalidade.

Observação

Esta função modifica a tabela sqlite de destino. Destina-se ao uso com configurações de valor padrão para atributos.

Quando o parâmetro do banco de dados for uma camada e a camada estiver no modo de transação, o valor será recuperado apenas uma vez durante o tempo de vida de uma transação e armazenado em cache e incrementado. Isso torna inseguro trabalhar no mesmo banco de dados de vários processos em paralelo.

Sintaxe	sqlite_fetch_and_increment(database, table, id_field, filter_attribute, filter_value, [default_values]) [] indica argumentos opcionais
Argumentos	• **database** - Caminho para o arquivo sqlite ou camada geopackage • **table** - Nome da tabela que gerencia as sequências • **id_field** - Nome do campo que contém o valor atual • **filter_attribute** - Nome do campo que contém o identificador único para essa sequência. Precisa ter o indice UNIQUE. • **filter_value** - Nome da sequencia a ser usada. • **default_values** - Mapa com valores padrão para colunas adicionais na tabela. Os valores precisam ser informados integralmente. Funções são permitidas.
Exemplos	• `sqlite_fetch_and_increment(@layer, 'sequence_table', 'last_unique_id', 'sequence_id', 'global', map('last_change', 'date(''now'')', 'user', '''' || @user_account_name || ''''))` → 0 • `sqlite_fetch_and_increment(layer_property(@layer, 'path'), 'sequence_table', 'last_unique_id', 'sequence_id', 'global', map('last_change', 'date(''now'')', 'user', '''' || @user_account_name || ''''))` → 0

Leitura adicional: *Data Sources Properties*, *Setting relations between multiple layers*

uuid

Gera um Idenrificador Universal Único (UUID) para cada linha utilizando o método Qt QUuid::createUuid.

Sintaxe	uuid([format='WithBraces']) [] indica argumentos opcionais
Argumentos	• **format** - Formato, que irá formatar o UUID. 'WithBraces', 'WithoutBraces' ou 'Id128'.
Exemplos	• `uuid()` → '{0bd2f60f-f157-4a6d-96af-d4ba4cb366a1}' • `uuid('WithoutBraces')` → '0bd2f60f-f157-4a6d-96af-d4ba4cb366a1' • `uuid('Id128')` → '0bd2f60ff1574a6d96afd4ba4cb366a1'

9.2.24 Relações

Este grupo contém a liste de *relações* disponíveis no projeto atual, junto com suas descrições. Ele permite acesso rápido ao ID da relação para a escrita de uma expressção (como por exemplo a função *relation_aggregate*) ou customização de um formulário.

9.2.25 Sensors Functions

This group contains functions to interact with sensors.

sensor_data

Returns the last captured value (or values as a map for sensors which report multiple values) from a registered sensor.

Sintaxe	sensor_data(name, [expiration]) [] indica argumentos opcionais
Argumentos	• **name** - the sensor name • **expiration** - maximum millisecond since last captured value allowed
Exemplos	• `sensor_data('geiger_1')` → '2000'

9.2.26 Funções de String

Este grupo contém funções que operam em textos (por exemplo, quer substituir, converter para maiúsculas).

ascii

Retorna o código unicode associado ao primeiro caractere de uma string.

Sintaxe	ascii(string)
Argumentos	• **string** - string a ser convertida para código unicode
Exemplos	• `ascii('Q')` → 81

char

Retorna um caractere associal a um código unicode.

Sintaxe	char(code)
Argumentos	• **code** - número de um código unicode
Exemplos	• `char(81)` → 'Q'

concat

Concatena várias strings em uma. Valores nulos são convertidos em strings vazias. Outros valores (como números) são convertidos para string).

Sintaxe	concat(string1, string2, …)
Argumentos	• **string** - um valor string
Exemplos	• `concat('sun', 'set')` → 'sunset' • `concat('a','b','c','d','e')` → 'abcde' • `concat('Anno ', 1984)` → 'Anno 1984' • `concat('The Wall', NULL)` → 'The Wall'

Sobre a concatenação de campos

Você pode também concatenar strings ou valores de campos utilizando também os operadores `||` ou +, com algumas características:

- O operador + também significa somar expressão; portanto, se você tiver um operando inteiro (valor numérico ou de campo), isso pode ser propenso a erros e é melhor usar os outros:

```
'My feature id is: ' + "gid" => triggers an error as gid returns an integer
```

- When any of the arguments is a NULL value, either `||` or + will return a NULL value. To return the other arguments regardless the NULL value, you may want to use the `concat` or `array_to_string` function:

```
'My feature id is: ' + NULL ==> NULL
'My feature id is: ' || NULL => NULL
concat('My feature id is: ', NULL) => 'My feature id is: '
array_to_string( array('My feature id is: ', NULL) ) => 'My feature id is: '
```

further reading: *array_to_string*, ||, +

format

Formate uma string usando os argumentos fornecidos.

Sintaxe	format(string, arg1, arg2, …)
Argumentos	• **string** - Uma string com espaços reservados %1, %2, etc., para os argumentos. Os espaços reservados podem ser repetidos. O espaço reservado de menor número é substituído por arg1, o próximo por arg2, etc. • **arg** - qualquer tipo. Qualquer número de argumentos.
Exemplos	• `format('Isso %1 um %2','é', 'teste')` → 'Isso é um teste' • `format('Isso é %2',' um pouco inesperado mas o 2 é o menor número na string','normal')` → 'Isso é um pouco inesperado mas o 2 é o menor número na string'

format_date

Fomata o tipo date ou string em uma string formadata customizada. Usa o formato de string do Qt date/time . Veja QDateTime::toString.

Sintaxe	format_date(datetime, format, [language]) [] indica argumentos opcionais
Argumentos	• **datetime** - valor data, hora ou datetime • **format** - String do modelo usado para formatar a string.

Expressão	Output
d	o dia como número sem zero à esquerda (1 a 31)
dd	o dia como número com um zero à esquerda (01 a 31)
ddd	localização abreveada do nome do dia (ex. 'Seg' a 'Dom')
dddd	o nome longo do dia localizado (ex., 'Segunda-feira' a 'Domingo')
M	o mês como número sem zero à esquerda (1-12)
MM	o mês como número com um zero à esquerda (01-12)
MMM	o nome abreviado do mês localizado (ex., 'Jan' a 'Dez')
MMMM	o nome longo do mês localizado (ex., 'janeiro' a 'dezembro')
yy	o ano como número de dois dígitos (00-99)
yyyy	o ano como número de quatro dígitos

Estas expressões podem ser usadas para a parte da hora da string de formatação:

Expressão	Output
h	a hora sem zero à esquerda (0 a 23 ou 1 a 12 se for exibido AM/PM)
hh	a hora com um zero à esquerda (00 a 23 ou 01 a 12 se for exibido AM/PM)
H	a hora sem zero à esquerda (0 a 23, mesmo com exibição AM/PM)
HH	a hora com um zero à esquerda (00 a 23, mesmo com exibição AM/PM)
m	o minuto sem zero à esquerda (0 a 59)
mm	o minuto com um zero à esquerda (00 a 59)
s	o segundo sem zero à esquerda (0 a 59)
ss	o segundo com um zero à esquerda (00 a 59)
z	os milissegundos sem zeros à direita (0 a 999)
zzz	os milissegundos com zeros à direita (000 a 999)
AP or A	Iinterprete como uma hora AM/PM. *AP* precisa ser ambos 'AM' ou 'PM'.
ap or a	Interprete como uma hora AM/PM. *ap* precisa ser ambos 'am' ou 'pm'.

- **language** - lingua (caixa baixa, duas ou três letras, código de linguagem ISO 639) usada para formatar a data em uma string customizada. Por padrão a localização do usuário atual do QGIS é utilizada.

Exemplos

- `format_date('2012-05-15','dd.MM.yyyy')` → '15.05.2012'
- `format_date('2012-05-15','d MMMM yyyy','fr')` → '15 mai 2012'
- `format_date('2012-05-15','dddd')` → 'Terça-feira', se a localização é uma variante do Português.
- `format_date('2012-05-15 13:54:20','dd.MM.yy')` → '15.05.12'
- `format_date('13:54:20','hh:mm AP')` → '01:54 PM'

format_number

Retonra um número formatado com o separador de milhar de acordo com a localização, Por padrão a localização atual do QGIS é utilizada. Também permite truncar as casas decimais para a quantidade de casas fornecidas.

Sintaxe	format_number(number, [places=0], [language], [omit_group_separators=false], [trim_trailing_zeroes=false]) [] indica argumentos opcionais
Argumentos	• **number** - número a ser formatado • **places** - número inteiro que representa o número de casas decimais para truncar a cadeia. • **language** - lingua (minúscula, duas- ou três letras, ISO 639 código de lingua) Utilizado para formatar o número em uma string. Por padrão a localização atual do usuário no QGIS é utilizada. • **omit_group_separators** - if set to true then group separators will not be included in the string • **trim_trailing_zeroes** - if set to true then trailing zeros following the decimal point will be trimmed from the string
Exemplos	• `format_number(10000000.332,2)` → '10,000,000.33' se, por exemplo, a localização atual for uma variante do Inglês • `format_number(10000000.332,2,'fr')` → '10 000 000,33'

left

Retorna uma parte de uma string que contém os *enésimos* caractere mais a esquerda de uma string.

Sintaxe	left(string, length)
Argumentos	• **string** - a string • **length** - inteiro. O número de caracteres à esquerda da cadeia a ser retornada.
Exemplos	• `left('Hello World',5)` → 'Hello'

Further reading: *right*

length

Retorna o número de caracteres em uma string ou o tamanho de uma geometria linha.

Variante string

Retorna o número de caracteres em uma string.

Sintaxe	length(string)
Argumentos	• **string** - string para contar o comprimento de
Exemplos	• `length('hello')` → 5

Variante geometria

Calcule o comprimento de um objeto de geometria linha. Os cálculos são sempre planimétricos no Sistema de Referência Espacial (SRS) dessa geometria, e as unidades do comprimento retornado corresponderão às unidades do SRS. Isso difere dos cálculos realizados pela função $length, que realizará cálculos elipsoidais com base nas configurações de elipsóide e unidade de distância do projeto.

Sintaxe	length(geometry)
Argumentos	• **geometry** - objeto de geometria linha
Exemplos	• `length(geom_from_wkt('LINESTRING(0 0, 4 0)'))` → 4.0

lower

Converte uma cadeia em letras minúsculas.

Sintaxe	lower(string)
Argumentos	• **string** - a string a ser convertida para letras minúsculas
Exemplos	• `lower('HELLO World')` → 'hello world'

Further reading: *upper*, *title*

lpad

Retorna uma cadeia preenchida à esquerda até a largura especificada, usando um caractere de preenchimento. Se a largura de destino for menor que o comprimento da cadeia, a cadeia, será truncada.

Sintaxe	lpad(string, width, fill)
Argumentos	• **string** - string para preencher • **width** - comprimento da nova cadeia • **fill** - caractere que irá preencher o espaço restante
Exemplos	• `lpad('Hello', 10, 'x')` → 'xxxxxHello' • `lpad('Hello', 3, 'x')` → 'Hel'

Further reading: *rpad*

ltrim

Removes the longest string containing only the specified characters (a space by default) from the start of string.

Sintaxe	ltrim(string, [characters=' ']) [] indica argumentos opcionais
Argumentos	• **string** - string para aparar • **characters** - characters to trim
Exemplos	• `ltrim(' hello world ')` → 'hello world ' • `ltrim('zzzytest', 'xyz')` → 'test'

Further reading: *rtrim*, *trim*

regexp_match

Retorna a primeira posição que corresponda com a expressão regular em uma string unicode, ou 0 se a sub string não for encontrada.

Sintaxe	regexp_match(input_string, regex)
Argumentos	• **input_string** - a string a ser usada contra a expressão regular • **regex** - A expressão regular a ser testada contra a string, os caracteres de barra invertida precisam ser escapados (Ex., "\\s" para combinar com espaço em branco ou "\\b" para corresponder a um limite de palavra).
Exemplos	• `regexp_match('QGIS ROCKS','\\sROCKS')` → 5 • `regexp_match('Budač','udač\\b')` → 2

Further reading: *strpos*

regexp_replace

Retorna uma string com Retorna uma string com a expressão regular fornecida substituída.

Sintaxe	regexp_replace(input_string, regex, replacement)
Argumentos	• **input_string** - a string para substituir as correspondências • **regex** - Expressão regular para substituir. caracteres com barra invertida devem ser escapados (e.x.: "\\s" para corresponder a um caractere de espaço em branco). • **replacement** - String que irá subistituir quaisquer ocorrências da expressão regular fornecida. captura de grupos pode ser inserida na string de substituição utilizando \\1, \\2, etc.
Exemplos	• `regexp_replace('QGIS SHOULD ROCK','\\sSHOULD\\s',' DOES ')` → 'QGIS DOES ROCK' • `regexp_replace('ABC123','\\d+','')` → 'ABC' • `regexp_replace('my name is John','(.*) is (.*)','\\2 is \\1')` → 'John is my name'

Further reading: *replace*, *array_replace*

regexp_substr

Retorna a parte de uma cadeia que corresponde a uma expressão regular fornecida.

Sintaxe	regexp_substr(input_string, regex)
Argumentos	• **input_string** - string onde irá procurar ocorrências • **regex** - A expressão regular a ser comparada. Os caracteres de barra invertida devem ter escape duplo (e.x.:, "\\s" para corresponder a um caractere de espaço em branco).
Exemplos	• `regexp_substr('abc123','(\\d+)')` → '123'

Further reading: *substr*, *regexp_matches*

repeat

Repeats a string a specified number of times.

Sintaxe	repeat(string, number)
Argumentos	• **string** - string to repeat • **number** - number of times to repeat string
Exemplos	• `repeat('Hello', 3)` → 'HelloHelloHello'

replace

Retorna uma string com a string, array ou mapa de strings fornecido substituído.

Variante String & array

Retorna uma string com a string ou matriz de strings fornecida substituída por uma string ou uma matriz de strings.

Sintaxe	replace(string, before, after)
Argumentos	• **string** - string de entrada • **before** - string ou matriz de strings para substituir • **after** - string ou matriz de strings a ser utilizadas como substitutas
Exemplos	• `replace('QGIS SHOULD ROCK','SHOULD','DOES')` → 'QGIS DOES ROCK' • `replace('QGIS ABC',array('A','B','C'),array('X','Y','Z'))` → 'QGIS XYZ' • `replace('QGIS',array('Q','S'),'')` → 'GI'

Variação por mapeamento

Retorna a string com as chaves de mapas substituidas pelos valores pareados. As chaves mais longas são avaliadas primeiro.

Sintaxe	replace(string, map)
Argumentos	• **string** - string de entrada • **map** - map que contém chaves e valores
Exemplos	• `replace('APP SHOULD ROCK',map('APP','QGIS','SHOULD', 'DOES'))` → 'QGIS DOES ROCK' • `replace('forty two',map('for','4','two','2','forty two', '42'))` → '42'

Further reading: *regexp_replace*, *array_replace*

reverse

Reverses the direction of a line string or reverses a string of text.

Variante string

Reverses the order of characters in a string.

Sintaxe	reverse(string)
Argumentos	• **string** - string to reverse
Exemplos	• `reverse('hello')` → 'olleh'

Variante geometria

Inverte a direção de uma cadeia de linha invertendo a ordem de seus vértices.

Sintaxe	reverse(geometry)
Argumentos	• **geometry** - uma geometria
Exemplos	• `geom_to_wkt(reverse(geom_from_wkt('LINESTRING(0 0, 1 1, 2 2)')))` → 'LINESTRING(2 2, 1 1, 0 0)'

right

Retorna uma sub string que contém os caracteres *n* mais à direita da string.

Sintaxe	right(string, length)
Argumentos	• **string** - a string • **length** - inteiro. O número de caracteres à direita da string a ser retornado.
Exemplos	• `right('Hello World',5)` → 'World'

Further reading: *left*

rpad

Retorna uma string preenchida à direita na largura especificada, usando um caractere de preenchimento. Se a largura de destino for menor que o comprimento da string, a string será truncada.

Sintaxe	rpad(string, width, fill)
Argumentos	• **string** - string para preencher • **width** - comprimento da nova cadeia • **fill** - caractere que irá preencher o espaço restante
Exemplos	• `rpad('Hello', 10, 'x')` → 'Helloxxxxx' • `rpad('Hello', 3, 'x')` → 'Hel'

Further reading: *lpad*

rtrim

Removes the longest string containing only the specified characters (a space by default) from the end of string.

Sintaxe	rtrim(string, [characters=' ']) [] indica argumentos opcionais
Argumentos	• **string** - string para aparar • **characters** - characters to trim
Exemplos	• `rtrim(' hello world ')` → ' hello world' • `rtrim('testxxzx', 'xyz')` → 'test'

Further reading: *ltrim*, *trim*

strpos

Retorna a primeira posição encontrada de uma substring em outra string, ou 0 se a substring não for encontrada.

Sintaxe	strpos(haystack, needle)
Argumentos	• **haystack** - string that is to be searched • **needle** - string to search for
Exemplos	• `strpos('HELLO WORLD','WORLD')` → 7 • `strpos('HELLO WORLD','GOODBYE')` → 0

Leitura adicional: *regexp_match*

substr

Retorna uma parte de uma string.

Sintaxe	substr(string, start, [length]) [] indica argumentos opcionais
Argumentos	• **string** - string de entrada inteira • **start** - inteiro representando a posição inicial para extrair começando com 1; se start for negativo, a string de retorno começará no final da string menos o valor inicial • **length** - inteiro representando o tamanho da string a ser extraída; se length for negativo, a string retornada irá omitir a quantidade de caracteres do final da string
Exemplos	• `substr('HELLO WORLD',3,5)` → 'LLO W' • `substr('HELLO WORLD',6)` → ' WORLD' • `substr('HELLO WORLD',-5)` → 'WORLD' • `substr('HELLO',3,-1)` → 'LL' • `substr('HELLO WORLD',-5,2)` → 'WO' • `substr('HELLO WORLD',-5,-1)` → 'WORL'

Further reading: *regexp_substr*, *regexp_matches*

title

Converte todas as palavras de uma string de título (todas as palavras com a primeira letra maiúscula).

Sintaxe	title(string)
Argumentos	• **string** - string a ser convertida
Exemplos	• `title('hello WOrld')` → 'Hello World'

Further reading: *lower*, *upper*

to_string

Converts a number to string. The conversion is not locale-aware, see 'format_number' for a locale-aware alternative.

Sintaxe	to_string(number)
Argumentos	• **number** - Interio ou valor real. Número a ser convertido para string.
Exemplos	• `to_string(1.23)` → '1.23'

Further reading: *format_number*

trim

Remove todos os espaços em branco do início e do final (espaços vazios, tabulações, etc) de uma string.

Sintaxe	trim(string)
Argumentos	• **string** - string para aparar
Exemplos	• `trim(' hello world ')` → 'hello world'

Further reading: *ltrim*, *rtrim*

upper

Converte uma string em letras maiúsculas.

Sintaxe	upper(string)
Argumentos	• **string** - string a ser convertida para maiúscula
Exemplos	• `upper('hello WOrld')` → 'HELLO WORLD'

Further reading: *lower*, *title*

wordwrap

Retorna uma string agrupada em um número máximo/mínimo de caracteres.

Sintaxe	wordwrap(string, wrap_length, [delimiter_string]) [] indica argumentos opcionais
Argumentos	• **string** - string a ser agrupada • **wrap_length** - um inteiro. Se wrap_length for positivo, o número representará o número máximo ideal de caracteres a serem agrupados; se negativo, o número representa o número mínimo de caracteres a serem agrupados. • **delimiter_string** - String elimitadora opcional para agrupar em uma nova linha.
Exemplos	• `wordwrap('UNIVERSITY OF QGIS',13)` → 'UNIVERSITY OF QGIS' • `wordwrap('UNIVERSITY OF QGIS',-3)` → 'UNIVERSITY OF QGIS'

9.2.27 Expressões do usuário

Este grupo contém as expressões salvas como *expressões do usuário*.

9.2.28 Variáveis

Este grupo contém variáveis dinâmicas relacionadas a aplicação, ao arquivo do projeto e outras configurações. A disponibilidade das variáveis depende do contexto:

- da caixa de diálogo Selecionar por expressão
- a partir da caixa de diálogo Calculadora de campo
- na caixa de diálogo de propriedades da camada
- do layout de impressão

Para usar essas variáveis em uma expressão, elas devem ser precedidas pelo caractere `@` (por exemplo, `@row_number`).

Variável	Descrição
algorithm_id	O ID exclusivo de um algoritmo
animation_end_time	Fim do intervalo de tempo geral da animação (como um valor de data e hora)
animation_interval	Duração do intervalo temporal geral da animação (como um valor de intervalo)
animation_start_time	Início do intervalo temporal geral da animação (como um valor de data e hora)
atlas_feature	A feição atual do atlas (como objeto de feição)
atlas_featureid	O ID da feição do atlas atual
atlas_featurenumber	O número atual da feição do atlas no layout
atlas_filename	O nome do arquivo do atlas atual
atlas_geometry	A geometria da feição do atlas atual
atlas_layerid	O ID da camada de cobertura do atlas atual
atlas_layername	O nome atual da camada de cobertura do atlas
atlas_pagename	O nome da página atual do atlas
atlas_totalfeatures	O número total de feições no atlas
band	The number of the band in the raster layer
band_description	The description of the band in the raster layer
band_name	The name of the band in the raster layer
canvas_cursor_point	A última posição do cursor na tela nas coordenadas geográficas do projeto
cluster_color	A cor dos símbolos em um cluster ou NULL se os símbolos tiverem cores misturadas
cluster_size	O número de símbolos contidos em um cluster
current_feature	A feição atualmente sendo editada no formulário de atributo ou na linha da tabela
current_geometry	A geometria da feição atualmente sendo editada no formulário ou na linha da tabela
current_parent_feature	representa a feição que está sendo editada no formulário pai. Apenas utilizável em um formulário incorporado pelo contexto.
current_parent_geometry	representa a geometria da feição que está sendo editada no formulário pai. Apenas utilizável em formulário incorporado pelo contexto.
form_mode	Para que o formulário é usado, como AddFeatureMode, SingleEditMode, MultiEditMode, SearchMode, AggregateSearchMode ou IdentifyMode como string.
feature	The current feature being evaluated. This can be used with the 'attribute' function to evaluate attribute values from the current feature.
frame_duration	Duração temporal de cada quadro de animação (como um valor de intervalo)
frame_number	Número do quadro atual durante a reprodução da animação
frame_rate	Número de quadros por segundo durante a reprodução da animação

continua na próxima página

Tabela 9.1 – continuação da página anterior

Variável	Descrição
fullextent_maxx	Valor máximo de x da extensão completa da tela (incluindo todas as camadas)
fullextent_maxy	Valor y máximo da extensão total da tela (incluindo todas as camadas)
fullextent_minx	Valor mínimo de x da extensão completa da tela (incluindo todas as camadas)
fullextent_miny	Valor y mínimo da extensão completa da tela (incluindo todas as camadas)
geometry	The geometry of the current feature being evaluated
geometry_part_count	O número de partes na geometria da feição renderizada
geometry_part_num	O número da parte da geometria atual para a feiição que está sendo renderizada
geometry_point_count	O número de pontos na parte da geometria renderizada
geometry_point_num	O número do ponto atual na parte da geometria renderizada
geometry_ring_num	Número do anel da geometria atual da feição sendo renderizado (para feições do tipo polígonos apenas). O anel exterior tem um valor de 0;
grid_axis	O eixo de anotação da grade atual (por exemplo, 'x' para longitude, 'y' para latitude)
grid_number	O valor atual da anotação da grade
id	The ID of the current feature being evaluated
item_id	O ID do usuário do item de layout (não necessariamente exclusivo)
item_uuid	O ID exclusivo do item de layout
layer	A camada atual
layer_crs	Autoridade do Sistema de Referência de Coordenadas da camada atual
layer_crs_ellipsoid	The ellipsoid Authority ID of the current layer CRS
layer_cursor_point	Point geometry under the mouse position in map canvas (or the GetFeatureInfo position in context of QGIS Server), in active layer's CRS
layer_id	O ID da camada atual
layer_ids	Uma lista com os ID de todas as camadas de mapas no projeto atual
layer_name	O nome da camada atual
layer_vertical_crs	The Identifier for the vertical coordinate reference system of the layer (e.g., 'EPSG:5703')
layer_vertical_crs_definition	The full definition of the vertical Coordinate reference system of the layer
layer_vertical_crs_description	The name of the vertical Coordinate reference system of the layer
layer_vertical_crs_wkt	The WKT definition of the vertical Coordinate reference system of the current layer
layers	Lista com todas as camadas no projeto atual
layout_dpi	A resolução da composição (DPI)
layout_name	O nome do layout
layout_numpages	O número de páginas no layout
layout_page	O número da página do item atual no layout
layout_pageheight	Altura da página atual no layout (em mm para tamanhos de papéis padrão, ou qualquer outra unidade sendo usada pelo tamanho de papel customizado)
layout_pageoffsets	Array de coordenadas do eixo Y de cada página. Permite posicionar itens dinamicamente nas páginas em um contexto em que os tamanhos das páginas podem mudar
layout_pagewidth	Largura da página atual no layout (em mm para tamanhos de papel padrão, ou outra unidade utilizada em tamanhos de papel customizado)
legend_column_count	O número de colunas na legenda
legend_filter_by_map	Indica se o conteúdo da legenda é filtrado pelo mapa
legend_filter_out_atlas	Indica se o atlas é filtrado para fora da legenda
legend_split_layers	Indica se as camadas podem ser divididas na legenda
legend_title	O título da lenda
legend_wrap_string	O(s) caractere(s) usados para quebrar o texto da legenda
map_crs	O Sistema de referência de coordenadas do mapa atual
map_crs_acronym	A sigla do Sistema de referência de coordenadas do mapa atual
map_crs_definition	A definição completa do Sistema de referência de coordenadas do mapa atual
map_crs_description	O nome do Sistema de referência de coordenadas do mapa atual
map_crs_ellipsoid	A sigla do elipsóide do Sistema de referência de coordenadas do mapa atual
map_crs_proj4	A definição Proj4 do Sistema de referência de coordenadas do mapa atual

continua na próxima página

Tabela 9.1 – continuação da página anterior

Variável	Descrição
map_crs_projection	O nome descritivo do método de projeção usado pelo sistema de referência de coordenadas do mapa (por exemplo, 'Albers Área Igual')
map_crs_wkt	A definição WKT do Sistema de referência de coordenadas do mapa atual
map_end_time	O fim do intervalo de tempo temporal do mapa (como um valor de data e hora)
map_extent	A geometria que representa a extensão atual do mapa
map_extent_center	A feição de ponto no centro do mapa
map_extent_height	A altura atual do mapa
map_extent_width	A largura atual do mapa
map_id	O ID do destino atual do mapa. Será 'canvas' para renderizações de tela, e o ID do item para renderizações de mapa de layout
map_interval	A duração do intervalo de tempo temporal do mapa (como um valor de intervalo)
map_layer_ids	A lista de IDs da camada do mapa visíveis no mapa
map_layers	A lista de camadas do mapa visíveis no mapa
map_rotation	A rotação atual do mapa
map_scale	A escala atual do mapa
map_start_time	O início do intervalo de tempo temporal do mapa (como um valor de data e hora)
map_units	As unidades de medidas do mapa
map_z_range_lower	Lower elevation of the map's elevation range
map_z_range_upper	Upper elevation of the map's elevation range
model_path	Caminho completo (incluindo o nome do arquivo) do modelo atual (ou caminho do projeto se o modelo estiver incorporado em um projeto).
model_folder	Pasta que contém o modelo atual (ou pasta do projeto se o modelo estiver incorporado em um projeto).
model_name	Nome do modelo atual
model_group	Grupo para o modelo atual
notification_message	Conteúdo da mensagem de notificação enviada pelo provedor (disponível apenas para ações acionadas por notificações do provedor).
parent	Refere-se a feição atual na camada pai, fornecendo acesso aos seus atributos e geometria ao filtrar uma função *aggregate*
plot_axis	The associated plot axis, e.g. 'x' or 'y'.
plot_axis_value	The current value for the plot axis.
project_abstract	O resumo do projeto, extraído dos metadados do projeto
project_area_units	A unidade de área do projeto atual, usada no cálculo de áreas de geometrias
project_author	O autor do projeto, extraído dos metadados do projeto
project_basename	O nome base do nome do arquivo do projeto atual (sem caminho e extensão)
project_creation_date	A data de criação do projeto, extraída dos metadados do projeto
project_crs	Identifier for the coordinate reference system of the project (e.g., 'EPSG:4326')
project_crs_arconym	A sigla do Sistema de referência de coordenadas do projeto
project_crs_definition	A definição completa do Sistema de referência de coordenadas do projeto
project_crs_description	A descrição do Sistema de referência de coordenadas do projeto
project_crs_ellipsoid	O elipsóide do Sistema de referência de coordenadas do projeto
project_crs_proj4	A representação do Proj4 do Sistema de referência de coordenadas do projeto
project_crs_wkt	A representação WKT (well known text) do Sistema de referência de coordenadas do projeto
project_distance_units	A unidade de distância para o projeto atual, usada no cálculo de comprimentos de geometrias e distâncias
project_ellipsoid	O nome do elipsóide do projeto atual, usado no cálculo de áreas geodésicas ou comprimentos de geometrias
project_filename	O nome do arquivo do projeto atual
project_folder	A pasta do projeto atual
project_home	O caminho inicial do projeto atual
project_identifier	O identificador do projeto, obtido dos metadados do projeto
project_keywords	As palavras-chave do projeto, extraídas dos metadados do projeto

continua na próxima página

Tabela 9.1 – continuação da página anterior

Variável	Descrição
project_last_saved	Data/hora em que o projeto foi salvo pela última vez.
project_path	O caminho completo (incluindo o nome do arquivo) do projeto atual
project_title	O título do projeto atual
project_units	As unidades do SRC do projeto
project_vertical_crs	Identifier for the vertical Coordinate reference system of the project (e.g., 'EPSG:5703')
project_vertical_crs_definition	The full definition of the vertical coordinate reference system of the project
project_vertical_crs_description	The description of the vertical coordinate reference system of the project
project_vertical_crs_wkt	The WKT (well known text) representation of the vertical coordinate reference system of the project
qgis_locale	O idioma atual do QGIS
qgis_os_name	O nome atual do sistema operacional, por exemplo, 'windows', 'linux' ou 'osx'
qgis_platform	A plataforma QGIS, por exemplo, 'desktop' ou 'servidor'
qgis_release_name	O nome da versão atual do QGIS
qgis_short_version	A string curta da versão atual do QGIS
qgis_version	A string de versão atual do QGIS
qgis_version_no	O número da versão atual do QGIS
row_number	Armazena o número da linha atual
snapping_results	Dá acesso aos resultados de encaixe ao digitalizar uma feição (disponível apenas no adicionar feição)
scale_value	O valor atual da distância da barra de escala
selected_file_path	Caminho de arquivo selecionado do seletor de widget de arquivo ao carregar um arquivo com um sistema de armazenamento externo
symbol_angle	O ângulo do símbolo usado para renderizar a feição (válido apenas para símbolos de marcador)
symbol_color	A cor do símbolo usado para renderizar a feição
symbol_count	O número de feições representadas pelo símbolo (na legenda do layout)
symbol_frame	The frame number (for animated symbols only)
symbol_id	O ID interno do símbolo (na legenda do layout)
symbol_label	O rótulo do símbolo (um rótulo definido pelo usuário ou o rótulo gerado automaticamente por padrão - na legenda do layout)
symbol_layer_count	Número total de camadas de símbolo no símbolo
symbol_layer_index	Current symbol layer index
symbol_marker_column	Número da coluna para marcador (válido apenas para preenchimentos de padrão de ponto).
symbol_marker_row	Número da linha do marcador (válido apenas para preenchimentos de padrão de ponto).
user_account_name	O nome da conta do sistema operacional do usuário atual
user_full_name	O nome do usuário do sistema operacional do usuário atual
value	O valor atual
vector_tile_zoom	Nível exato de zoom do bloco vetorial do mapa que está sendo renderizado (derivado da escala do mapa atual). Normalmente no intervalo [0, 20]. Ao contrário de @zoom_nível, esta variável é um valor de ponto flutuante que pode ser usado para interpolar valores entre dois níveis de zoom inteiros.
with_variable	Permite definir uma variável para uso em uma expressão e evitar recalcular o mesmo valor repetidamente
zoom_level	Nivel de zoom do tile do vetor no mapa que está sendo renderizado (derivado da escala do mapa atual). Normalmente um intervalo [0,20].

Alguns exemplos:

- Retorna a coordenada X de um centro de item de mapa no layout:

```
x( map_get( item_variables( 'map1'), 'map_extent_center' ) )
```

- Retorne, para cada feição na camada atual, o número de feições de aeroporto sobrepostas:

```
aggregate( layer:='airport', aggregate:='count', expression:="code",
              filter:=intersects( $geometry, geometry( @parent ) ) )
```

- Obtém o object_id do primeiro ponto ajustado de uma linha:

```
with_variable(
  'first_snapped_point',
  array_first( @snapping_results ),
  attribute(
    get_feature_by_id(
      map_get( @first_snapped_point, 'layer' ),
      map_get( @first_snapped_point, 'feature_id' )
    ),
    'object_id'
  )
)
```

9.2.29 Funções Recentes

Este grupo contém funções usadas recentemente. Dependendo do contexto de seu uso (seleção de feição, calculadora de campo, genérico), as expressões aplicadas recentemente são adicionadas à lista correspondente (até dez expressões), ordenadas de mais para menos recente. Isso facilita a rápida recuperação e reaplicação de expressões usadas anteriormente.

CAPÍTULO 10

Biblioteca de estilo

10.1 Gerenciador de Estilo

10.1.1 The Style Manager dialog

The *Style Manager* is the place where you can manage and create generic style items. These are symbols, color ramps, text formats or label settings that can be used to symbolize features, layers or print layouts. They are stored in the `symbology-style.db` database under the active *user profile* and shared with all the project files opened with that profile. Style items can also be shared with others thanks to the export/import capabilities of the *Style Manager* dialog.

You can open that modeless dialog either:

- from the *Settings* ► *Style Manager…* menu
- with the Style Manager button from the Project toolbar
- or with the Style Manager button from a vector *Layer Properties* ► menu (while *configuring a symbol* or *formatting a text*).

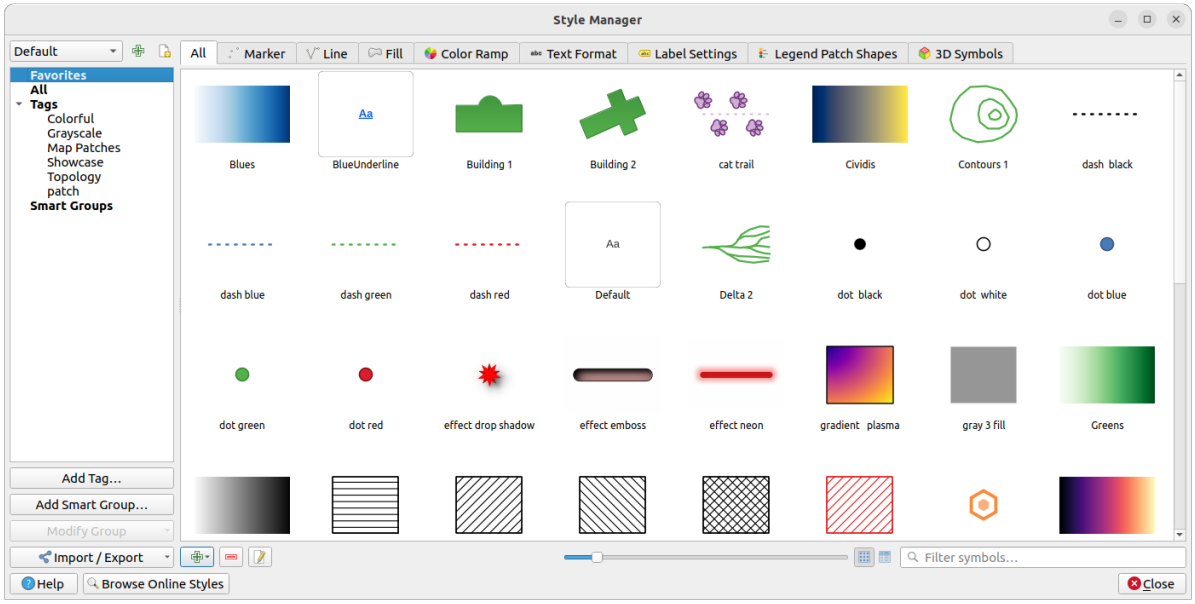


Figura10.1: Gerenciador de Estilo

Organizando itens de estilo

In the upper left corner of the *Style Manager* dialog, within the drop-down menu, you can choose the style database you want to connect to. If you choose *Default* you will be connected to the default style database where you can find all available default style items and the ones that you saved in this database. Choosing *Project Styles* will connect you to the Project Style database where you can find only symbols that you saved in this database. If you created more style databases, they will be listed in the drop-down menu. There are also options to Add existing style database to project and Create new style database (see more *Style database*).

For each style database, you can organize the elements into different categories, listed in the panel on the left:

- **Favoritos**: exibido por padrão ao configurar um item, ele mostra um conjunto extensível de itens;
- **Todos**: lista todos os itens disponíveis para o tipo ativo;
- **Tags**: shows a list of labels you can use to identify the items. An item can be tagged more than once. Select a tag in the list and the tabs are updated to show only their items that belong to it. To create a new tag you could later attach to a set of items, use the *Add Tag...* button or select the *Add Tag...* from any tag contextual menu;
- **Smart Group**: a smart group dynamically fetches its symbols according to conditions set (see eg, Fig.10.2). Click the *Add Smart Group...* button to create smart groups. The dialog box allows you to enter an expression to filter the items to select (has a particular tag, have a string in its name, etc.). Any symbol, color ramp, text format or label setting that satisfies the entered condition(s) is automatically added to the smart group.

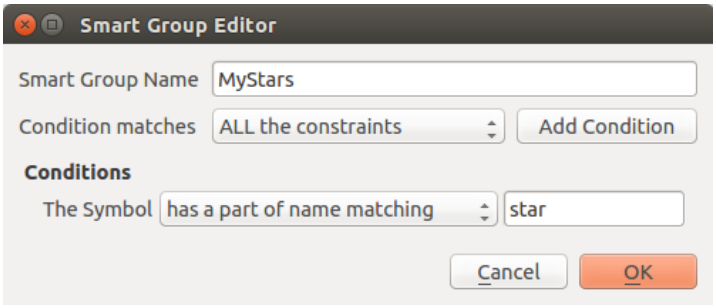


Figura10.2: Criando um Grupo Inteligente

Tags and smart groups are not mutually exclusive: they are simply two different ways to organize your style elements. Unlike the smart groups that automatically fetch their belonged items based on the input constraints, tags are filled by the user. To edit any of those categories, you can either:

- select the items, right-click and choose *Add to Tag* ► and then select the tag name or create a new tag;
- select the tag and press *Modify group…* ► *Attach Selected Tag to Symbols*. A checkbox appears next to each item to help you select or deselect it. When selection is finished, press *Modify group…* ► *Finish Tagging*.
- select the smart group, press *Modify group…* ► *Edit smart group…* and configure a new set of constraints in the *Smart Group Editor* dialog. This option is also available in the contextual menu of the smart group.

To remove a tag or a smart group, right-click on it and select the *Remove* button. Note that this does not delete the items grouped in the category.

The *Style Manager* dialog displays in its center a frame with previewed items organized into tabs:

- *All* for a complete collection of point, linear and surface symbols and label settings as well as predefined color ramps and text formats;
- *Marker* for point symbols only;
- *Linha* apenas para símbolos lineares;
- *Preenchimento* somente para símbolos de superfície;
- *Color ramp*;
- *Text format* to manage *text formats*, which store the font, color, buffers, shadows, and backgrounds of texts (i.e. all the formatting parts of the label settings, which for instance can be used in layouts);
- *Label settings* to manage *label settings*, which include the text formats and some layer-type specific settings such as label placement, priority, callouts, rendering…
- *Legend Patch Shapes* to manage custom legend patch shapes, which include *Marker*, *Line* and *Fill* geometries.
- *3D Symbols* to configure symbols with *3D properties* (extrusion, shading, altitude, …) for the features to render in a *3D Map view*

You can arrange the Styles in *Icon View* or in *List View* on the bottom right side. In both views the tooltip shows a larger instance of the style. The thumbnail size slider at the left of the icons helps you adjust the actual thumbnail sizes in the dialog, for a better preview of the symbols!

Adicionando, editando ou removendo um item

Como visto anteriormente, os elementos de estilo são listados em diferentes abas cujo conteúdo depende da categoria ativa (etiqueta, grupo inteligente, favoritos…). Quando uma aba está habilitada, você pode:

- Add new items: press the Add item button and configure the item following *symbols*, *color ramps* or *text format and label* builder description.
- Modify an existing item: select an item and press Edit item button and configure as mentioned above.
- Delete existing items: to delete an element you no longer need, select it and click Remove item (also available through right-click). The item will be deleted from the local database.

Note that the *All* tab provides access to these options for every type of item.

Right-clicking over a selection of items also allows you to:

- *Adicionar aos favoritos*;
- guilabel:*Remover dos favoritos*;
- *Add to Tag* ► and select the appropriate tag or create a new one to use; the currently assigned tags are checked;
- *Clear Tags*: detaching the symbols from any tag;
- *Remover Item(ns)*;
- guilabel:*Editar Item*: aplica-se ao item sobre o qual você clica com o botão direito do mouse;
- *Copiar Item*;
- *Paste Item* …: pasting to one of the categories of the style manager or elsewhere in QGIS (symbol or color buttons)
- guilabel:*Exportar Símbolo(s) Selecionado(s) como PNG…* (somente disponível com símbolos);
- guilabel:*Exportar Símbolo(s) Selecionado(s) como SVG…* (somente disponível com símbolos);

Compartilhando estilo de itens

The *Import/Export* tool, at the left bottom of the Style Manager dialog, offers options to easily share symbols, color ramps, text formats and label settings with others. These options are also available through right-click over the items.

Exportando itens

Você pode exportar um conjunto de itens para um arquivo `.XML`:

1. Expand the *Import/Export* drop-down menu and select *Export Item(s)…*
2. Escolha os itens que você gostaria de integrar. A seleção pode ser feita com o mouse ou usando uma tag ou um grupo previamente definido.
3. Press *Export* when ready. You'll be prompted to indicate the destination of the saved file. The XML format generates a single file containing all the selected items. This file can then be imported in another user's style library.

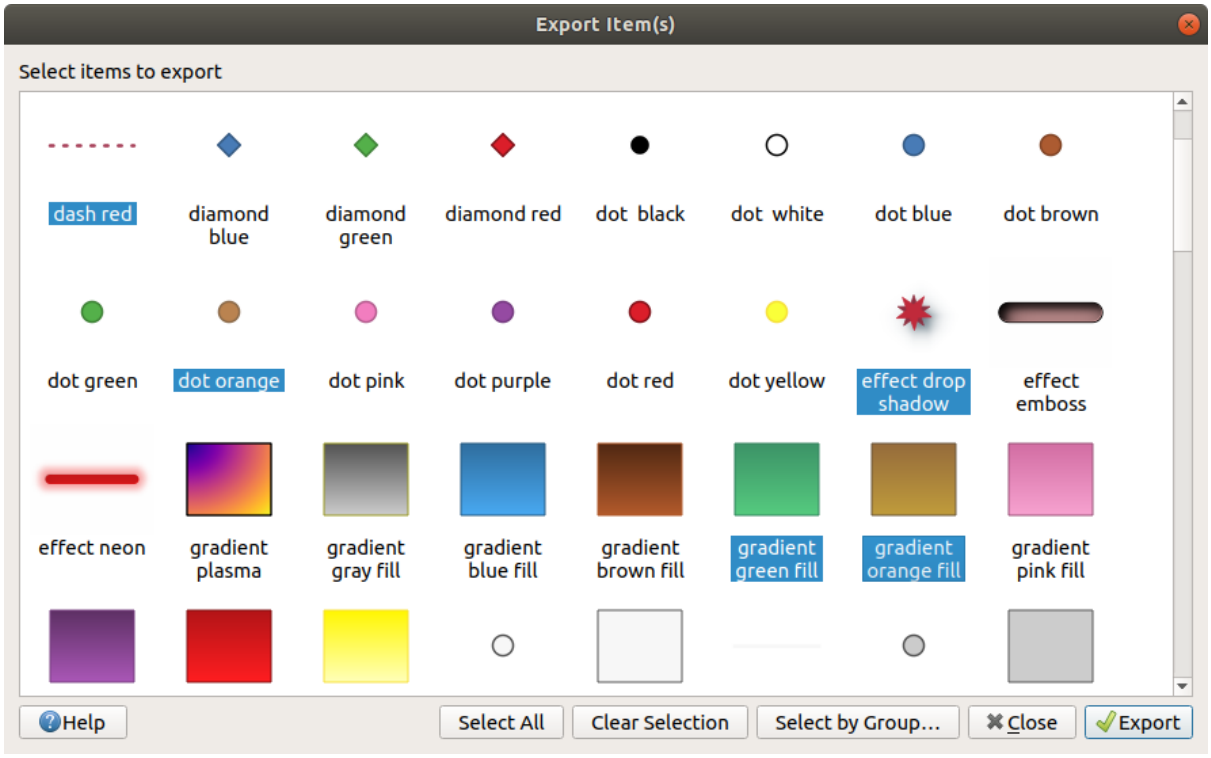


Figura10.3: Exporting style items

When symbols are selected, you can also export them to `.PNG` or `.SVG`. Exporting to `.PNG` or `.SVG` (both not available for other style item types) creates a file for each selected symbol in a given folder. The SVG folder can be added to the *SVG paths* in *Settings* ► *Options* ► *System* menu of another user, allowing him direct access to all these symbols.

Importando itens

Você pode ampliar sua biblioteca de estilos, importando novos itens:

1. Expand the *Import/Export* drop-down menu and select *Import Item(s)* at the left bottom of the dialog.
2. In the new dialog, indicate the source of the style items (it can be an `.xml` file on the disk or a url).
3. Set whether to *Add to favorites* the items to import.
4. Check *Do not import embedded tags* to avoid the import of tags associated to the items being imported.
5. Give the name of any *Additional tag(s)* to apply to the new items.
6. Selecione na pré-visualização os símbolos que deseja adicionar à sua biblioteca.
7. E pressione :guilabel:`importar'.

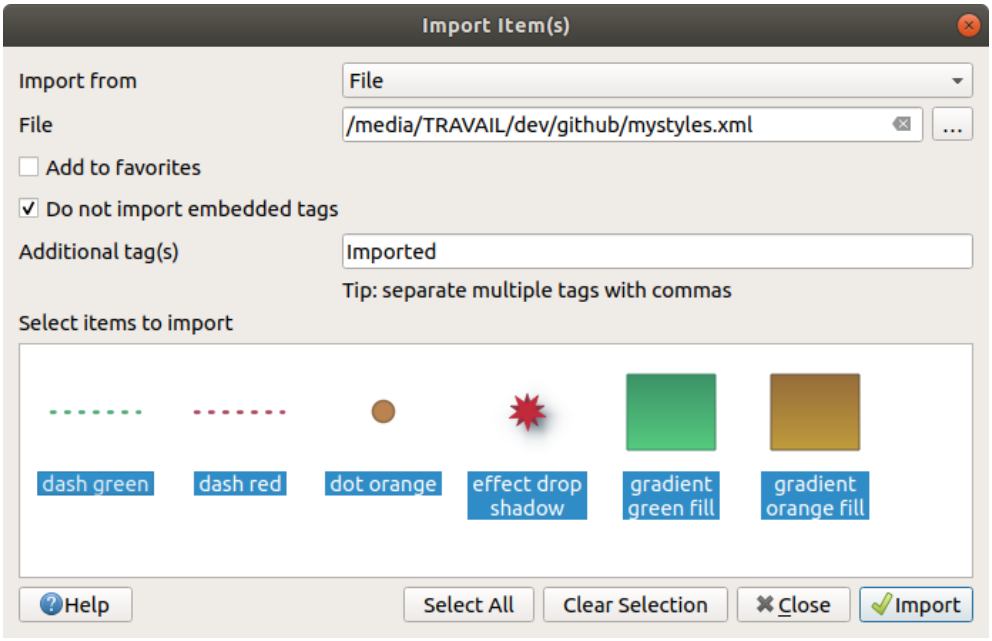


Figura10.4: Importing style items

Using the Browser panel

It's also possible to import style items into the active user profile style database directly from the *Browser* panel:

1. Selecione o estilo :file: arquivo *.xml* no navegador
2. Arraste e solte sobre a tela do mapa ou clique com o botão direito do mouse e selecione *Importar Estilo…*
3. Fill the *Import Items* dialog following *Importando itens*
4. Press *Import* and the selected style items are added to the style database

Double-clicking the style file in the browser opens the *Style Manager* dialog showing the items in the file. You can select them and press *Copy to Default Style…* to import them into the active style database. Tags can be assigned to items. Also available through right-click, *Open Style…* command.

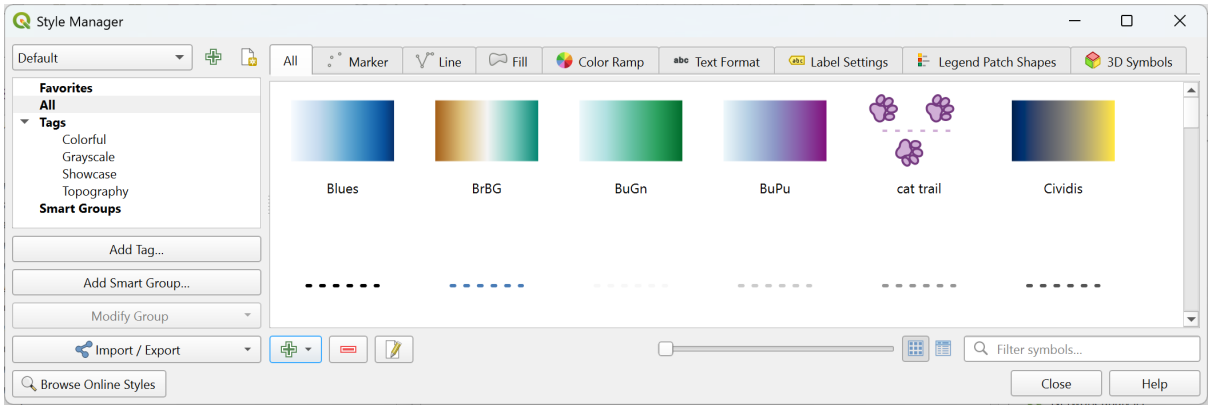


Figura10.5: Opening a style items file

A caixa de diálogo também permite exportar símbolos únicos como arquivos :file: *.PNG* ou `.SVG`.

Usando o repositório online

The QGIS project maintains a repository with a collection of styles shared by QGIS users. This is available at https://hub.qgis.org/styles and can be accessed from the *Style Manager* dialog, pressing the *Browse Online Styles* button at the bottom.

A partir desse repositório, você pode:

1. Browse and search for any style items, based on their type or name
2. Download the style file and unzip it
3. Load the `.xml` based file into your style database in QGIS, using any of the aforementioned import methods.

10.1.2 Setting a Color Ramp

The Color ramp tab in the *Style Manager* dialog helps you preview different color ramps based on the category selected in the left panel.

To create a custom color ramp, activate the Color ramp tab and click the Add item button. The button reveals a drop-down list to choose the ramp type:

- *Gradient*: given a start and end colors, generates a color ramp which can be **continuous** or **discrete**. With double-clicking the ramp preview, you can add as many intermediate color stops as you want. Click on the color stop indicator and under *Gradient stop* you can:
  - adjust its *Relative position* from the color ramp start. Also possible dragging the indicator with the mouse, or pressing the arrow keys (combine with `Shift` key for a larger move)
  - specify the color model to use when interpolating between colors: it can be *RGB*, *HSL* or *HSV*. In some circumstances, this option can help avoid desaturated mid tones, resulting in more visually pleasing gradients.
  - set the direction which the interpolation should follow for the **Hue** component of a *HSL* or *HSV* color specification. It can be *Clockwise* or *Counterclockwise*.
  - set the *color properties*
  - remove the color stop pressing *Delete stop* or `DEL`

  The *Plots* group provides another graphical way to design the color ramp, changing the position or the opacity and HSL components of the color stops.

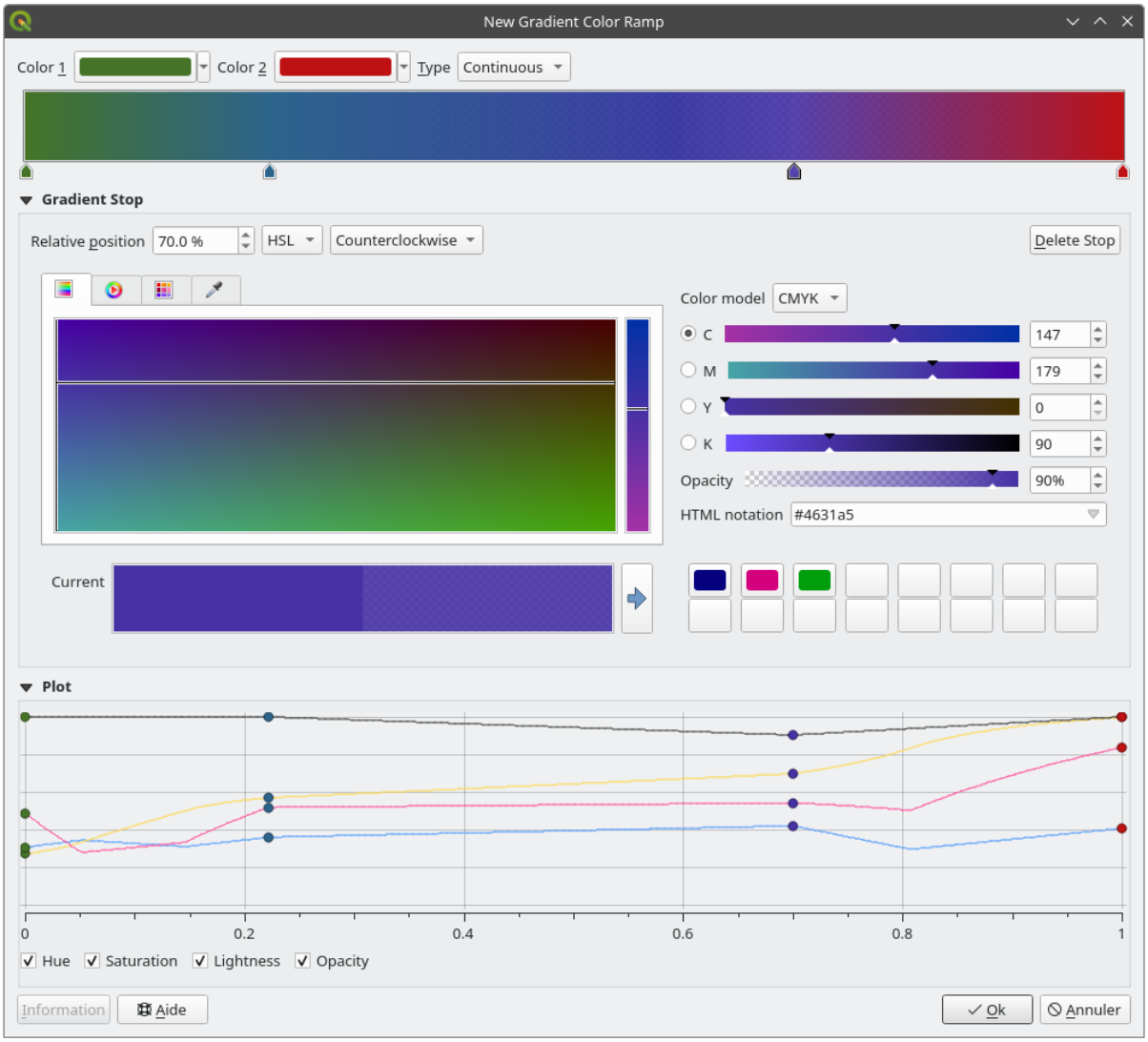


Figura10.6: Example of custom gradient color ramp with multiple stops

> **Dica**
>
> Drag-and-drop a color from a color spot onto the gradient ramp preview adds a new color stop.

- *Color presets*: allows to create a color ramp consisting of a list of colors selected by the user;
- *Random*: creates a random set of colors based on range of values for *Hue*, *Saturation*, *Value* and *Opacity* and a number of colors (*Classes*);
- *Catalog: ColorBrewer*: a set of predefined discrete color gradients you can customize the number of colors in the ramp;
- or *Catalog: cpt-city*: an access to a whole catalog of color gradients to locally *save as standard gradient*. The cpt-city option opens a new dialog with hundreds of themes included 'out of the box'.

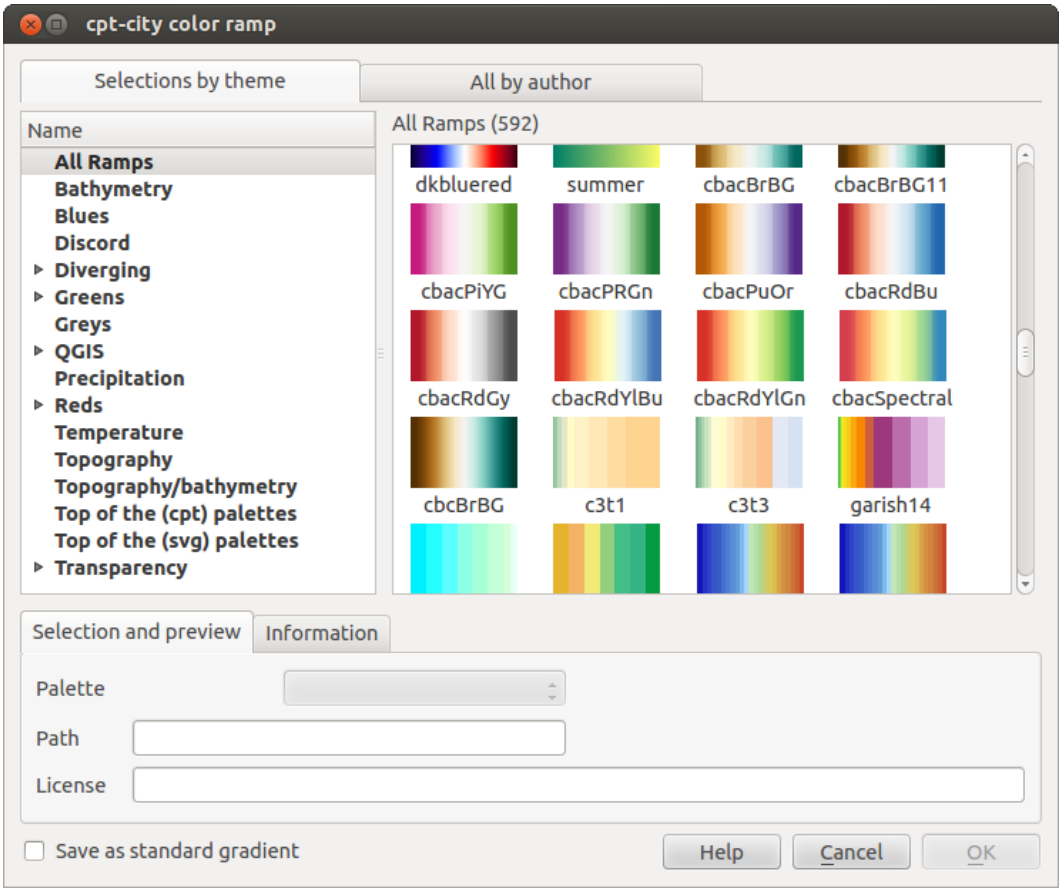


Figura10.7: cpt-city dialog with hundreds of color ramps

10.1.3 Creating a Legend Patch Shape

To create a new Legend Patch Shape, activate the *Legend Patch Shapes* tab and click the Add item button. The button reveals a drop-down list to choose the geometry type:

- *Marker Legend Patch Shape…*: to use with point geometries.
- *Line Legend Patch Shape…*: to use with line geometries.
- *Fill Legend Patch Shape…*: to use with polygon geometries.

Todas as três opções mostrarão a mesma caixa de diálogo.

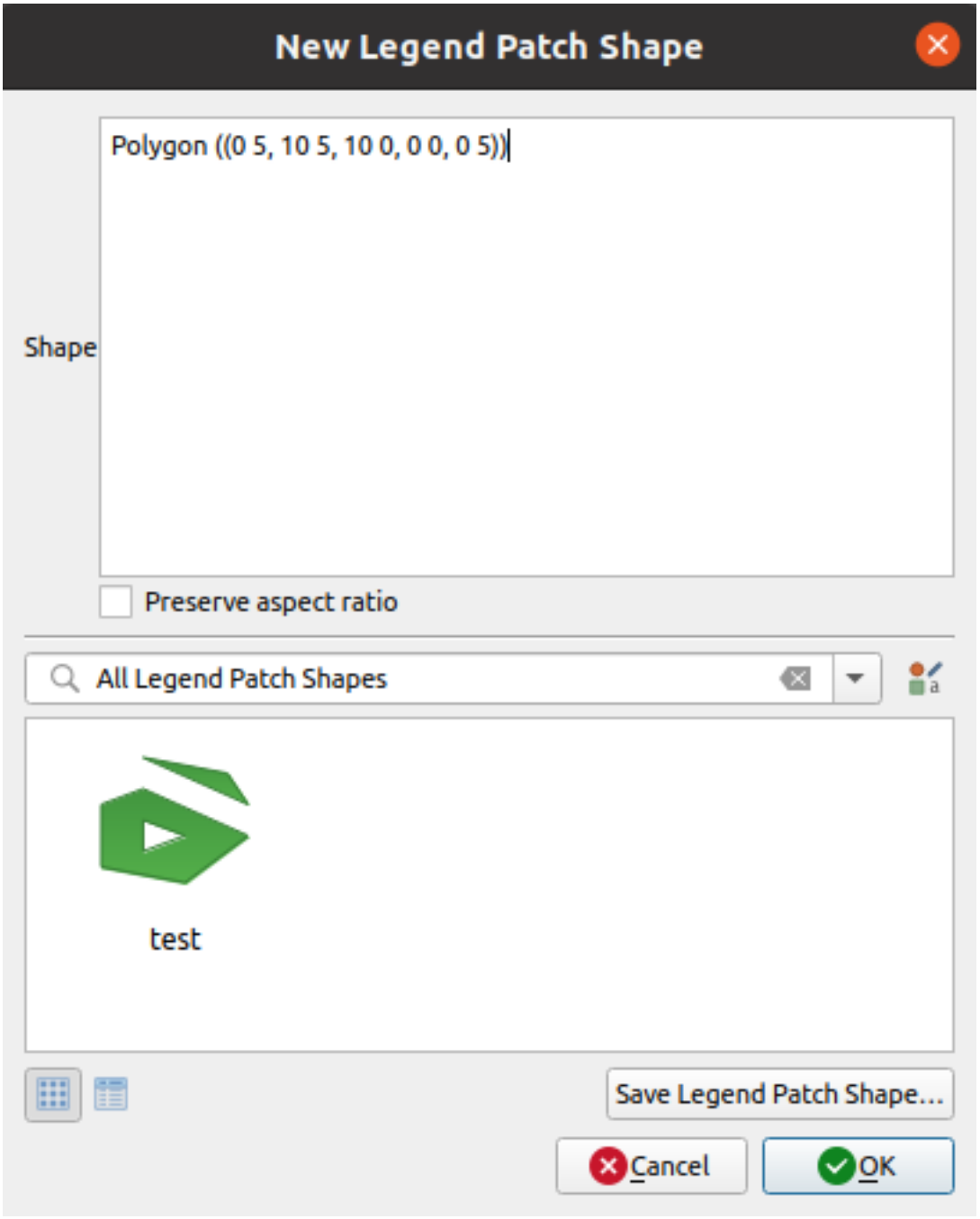


Figura10.8: Create a new Legend Patch Shape

Only the shape type and displayed legend patch shapes will differ regarding to the chosen geometry type. The following options will be available:

- *Shape*: define the shape of the legend patch shape as a WKT string. Single and multipart geometries may be used, but no GeometryCollection.
- |caixa de seleção| *Manter a proporção*
- *Icon View* or *List View* of available legend patch shapes, filtered by tags.

When the new Shape is defined you can *Save Legend Patch Shape…* or press *OK*, which will both lead to the same dialog.

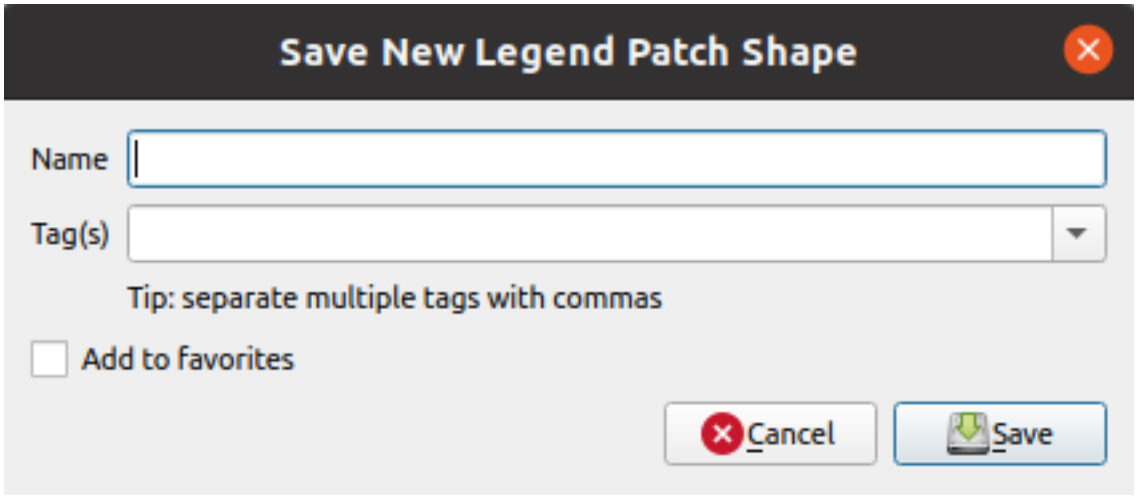


Figura10.9: Save a new Legend Patch Shape

Aqui você tem que escolher um nome, etiquetas para descrever a forma e se ela deve ser adicionada aos favoritos.

If you press *Save…*, the shape is added to the list and you are directed back to the *New Legend Patch Shape* dialog to keep creating new shapes.

10.2 O Seletor de Símbolos

The Symbol selector is the main dialog to design a symbol. You can create or edit Marker, Line or Fill Symbols.

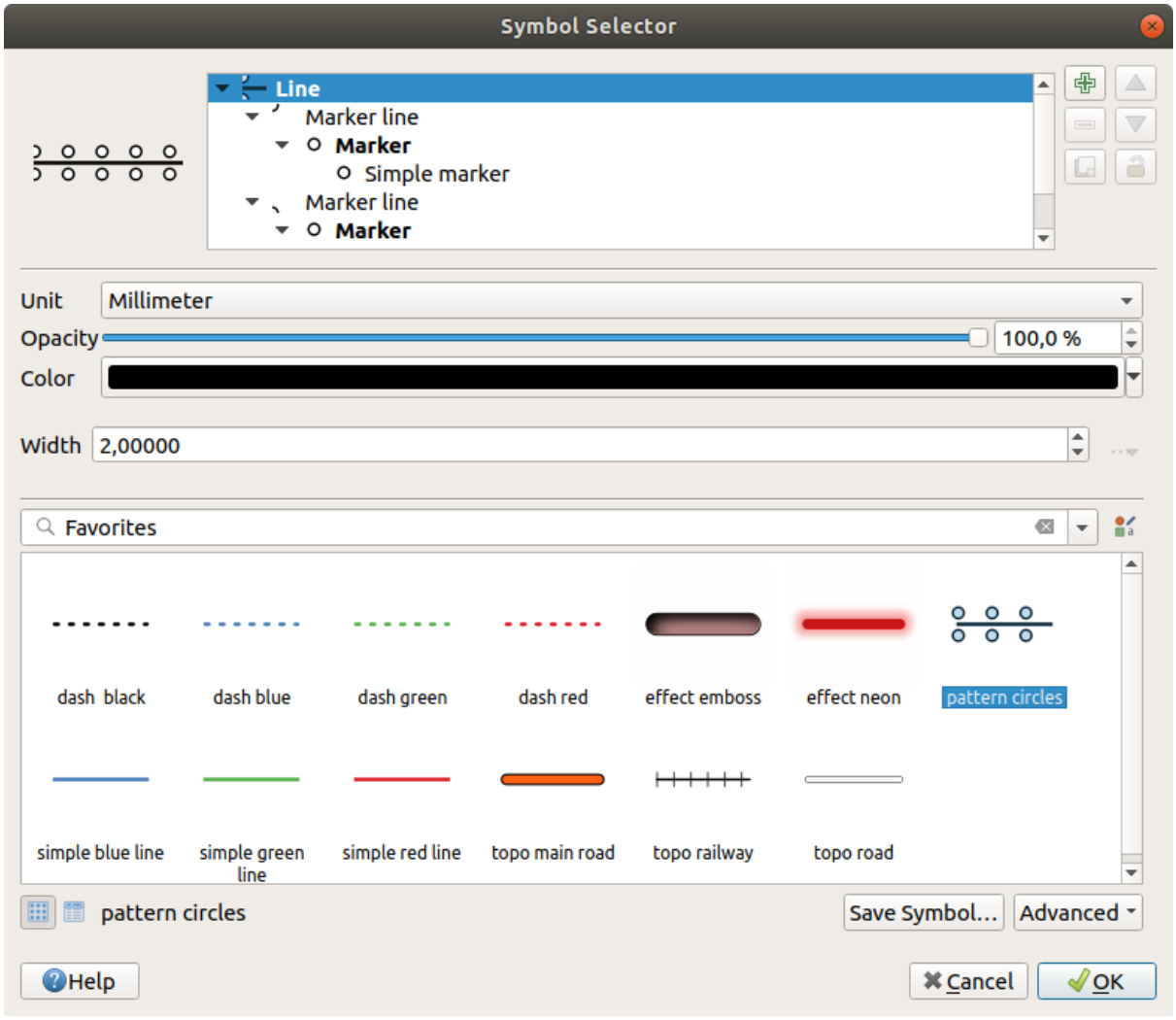


Figura10.10: Designing a Line symbol

Two main components structure the symbol selector dialog:

- the symbol tree, showing symbol layers that are combined afterwards to shape a new global symbol
- and settings to configure the selected symbol layer in the tree.

10.2.1 The symbol layer tree

A symbol can consist of several *Symbol layers*. The symbol tree shows the overlay of these symbol layers that are combined afterwards to shape a new global symbol. Besides, a dynamic symbol representation is updated as soon as symbol properties change.

Depending on the level selected in the symbol tree items, various tools are made available to help you manage the tree:

- add new symbol layer: you can stack as many symbols as you want
- remove the selected symbol layer
- lock colors of symbol layer: a locked color stays unchanged when user changes the color at the global (or upper) symbol level
- duplicate a (group of) symbol layer(s)

- move up or down the symbol layer

10.2.2 Configurando um símbolo

In QGIS, configuring a symbol is done in two steps: the symbol and then the symbol layer.

O símbolo

At the top level of the tree, it depends on the layer geometry and can be of **Marker**, **Line** or **Fill** type. Each symbol can embed one or more symbols (including, of any other type) or symbol layers.

You can setup some parameters that apply to the global symbol:

- *Unit*: it can be **Millimeters**, **Points**, **Pixels**, **Meters at Scale**, **Map units** or **Inches** (see *Seletor de Unidade* for more details)
- *Opacidade*
- *Color*: when this parameter is changed by the user, its value is echoed to all unlocked sub-symbols color
- *Size* and *Rotation* for marker symbols
- *Largura* para símbolos de linha

> **Dica**
>
> Use the *Size* (for marker symbols) or the *Width* (for line symbols) properties at the symbol level to proportionally resize all of its embedded *symbol layers* dimensions.

> **Nota**
>
> The *Data-defined override* button next to the width, size or rotation parameters is inactive when setting the symbol from the Style manager dialog. When the symbol is connected to a map layer, this button helps you create *proportional or multivariate analysis* rendering.

- A preview of the *symbols library*: Symbols of the same type are shown and, through the editable drop-down list just above, can be filtered by free-form text or by *categories*. You can also update the list of symbols using the Style Manager button and open the eponym dialog. There, you can use any capabilities as exposed in *Gerenciador de Estilo* section.

  Os símbolos são exibidos:

  - in an icon list (with thumbnail, name and associated tags) using the List View button below the frame;
  - or as icon preview using the Icon View button.

- Press the *Save Symbol* button to open the *Save New Symbol* dialog. Here, you can choose the *Destination* where you want to add the symbol being edited, give it a *Name* and add *Tag(s)*. You also have the option to Add to favorites your new symbol.

- With the *Advanced* option, you can:

  - for line and fill symbols, *Clip features to canvas extent*.
  - for fill symbols, *Force right-hand rule orientation*: allows forcing rendered fill symbols to follow the standard “right hand rule” for ring orientation (i.e, polygons where the exterior ring is clockwise, and the interior rings are all counter-clockwise).

The orientation fix is applied while rendering only, and the original feature geometry is unchanged. This allows for creation of fill symbols with consistent appearance, regardless of the dataset being rendered and the ring orientation of individual features.

- for marker symbols, the *Buffer settings…* enables addition of a halo effect around the marker symbol in order to make it more readable against different backgrounds. The buffer is calculated and drawn using the shape of ALL the symbol layers in the marker. You can adjust the *Size*, *Join style* and *Symbol* of display.
- for all symbol types, enable and configure a distance for the map canvas *Extent buffer*, or the *Animation settings*.
- Depending on the *symbology* of the layer the symbol is being applied to, additional settings are available in the *Advanced* menu:
  - *Symbol levels…* to define the order of symbols rendering
  - *Data-defined Size Legend*
  - *Match to Saved Symbols…* and *Match to Symbols from File…* to automatically *assign symbols to classes*

A camada de símbolo

At a lower level of the tree, you can customize the symbol layers. The available symbol layer types depend on the upper symbol type. You can apply on the symbol layer *paint effects* to enhance its rendering.

Because describing all the options of all the symbol layer types would not be possible, only particular and significant ones are mentioned below.

Parâmetros comuns

Some common options and widgets are available to build a symbol layer, regardless it's of marker, line or fill sub-type:

- the *color selector* widget to ease color manipulation
- *Units*: it can be **Millimeters**, **Points**, **Pixels**, **Meters at Scale**, **Map units** or **Inches** (see *Seletor de Unidade* for more details)
- the Data-defined override widget near almost all options, extending capabilities of customizing each symbol (see *Data defined override setup* for more information)
- the *Enable symbol layer* option controls the symbol layer's visibility. Disabled symbol layers are not drawn when rendering the symbol but are saved in the symbol. Being able to hide symbol layers is convenient when looking for the best design of your symbol as you don't need to remove any for the testing. The data-defined override then makes it possible to hide or display different symbol layers based on expressions (using, for instance, feature attributes).
- the *Draw effects* button for *effects rendering*.

Nota

While the description below assumes that the symbol layer type is bound to the feature geometry, keep in mind that you can embed symbol layers in each others. In that case, the lower level symbol layer parameter (placement, offset…) might be bound to the upper-level symbol, and not to the feature geometry itself.

Marker Symbols

Appropriate for point geometry features, marker symbols have several *Symbol layer types*:

Tabela10.1: List of marker symbol layer types

Simple marker	*Animated marker*	*Ellipse marker*
Filled marker	*Font marker*	*Geometry generator*
Mask	*Raster image marker*	*Vector Field marker*
SVG marker		

- **Simple marker** (default)

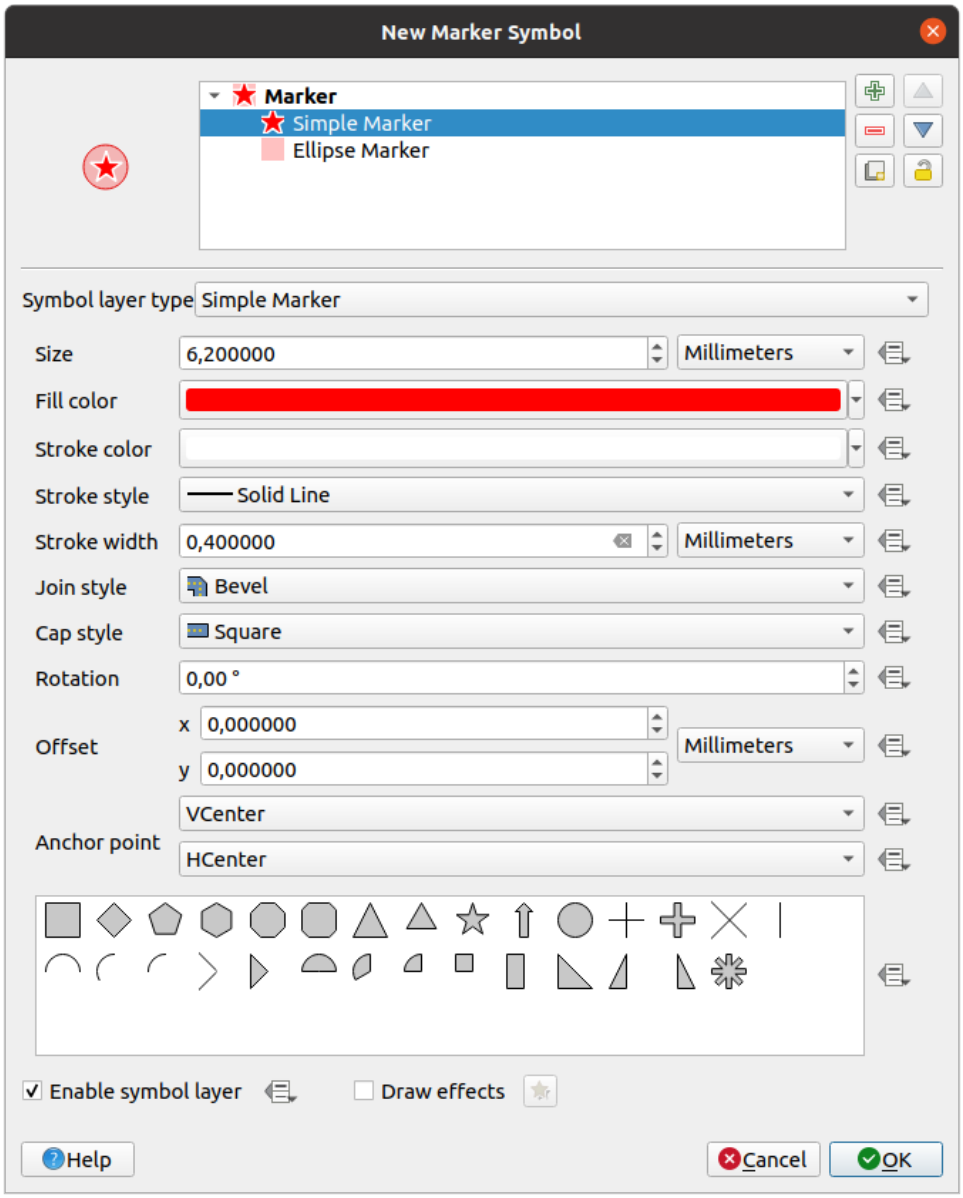


Figura10.11: Designing a Simple Marker Symbol

The simple marker symbol layer type has the following properties:

- *Tamanho* em várias unidades suportadas

  - *Fill color*
  - *Stroke color*, *Stroke style* from a predefined list and *Stroke size*
  - *Join style*: it can be **Bevel**, **Miter** or **Round**
  - *Cap style*: it can be **Square**, **Flat** or **Round**
  - *Rotação*
  - *Offset* in *X* and *Y* directions from the feature
  - *Anchor point*: defining the quadrant point on the symbol to settle as placement origin. This is the point the *Offset* is applied on.
- **Animated marker** (see *Animated marker*)
- **Ellipse marker**: a simple marker symbol layer, with customizable width and height
- **Filled marker**: similar to the simple marker symbol layer, except that it uses a *fill sub symbol* to render the marker. This allows use of all the existing QGIS fill (and stroke) styles for rendering markers, e.g. gradient or shapeburst fills.
- **Font marker**: similar to the simple marker symbol layer, except that it uses installed fonts to render the marker. Its additional properties are:
  - *Font family*
  - *Estilo de fonte*
  - *Character(s)*, representing the text to display as symbol. They can be typed in or selected from the font characters collection widget and you can live *Preview* them with the selected settings.
- **Geometry generator** (see *O Gerador de Geometria*)
- **Mask**: its sub-symbol defines a mask shape whose color property will be ignored and only the opacity will be used. This is convenient when the marker symbol overlaps with labels or other symbols whose colors are close, making it hard to decipher. More details at *Propriedades das Máscaras*.
- **Raster image marker**: use an image (`PNG`, `JPG`, `BMP` ...) as marker symbol. The image can be a file on the disk, a remote URL, embedded in the style database (*more details*) or it can be encoded as a base64 string. Width and height of the image can be set independently or using the Lock aspect ratio. The size can be set using any of the *common units* or as a percentage of the image's original size (scaled by the width).
- **Vector Field marker** (see *The Vector Field Marker*)
- **SVG marker**: provides you with images from your SVG paths (set in *Settings* ► *Options...* ► *System* menu) to render as marker symbol. Width and height of the symbol can be set independently or using the Lock aspect ratio. Each SVG file colors and stroke can also be adapted. The image can be a file on the disk, a remote URL, embedded in the style database (*more details*) or it can be encoded as a base64 string.

  The symbol can also be set with *Dynamic SVG parameters*. See *SVG parametrizável* section to parametrize an SVG symbol.

> **Nota**
>
> SVG version requirements
>
> QGIS renders SVG files that follow the SVG Tiny 1.2 profile, intended for implementation on a range of devices, from cellphones and PDAs to laptop and desktop computers, and thus includes a subset of the features included in SVG 1.1 Full, along with new features to extend the capabilities of SVG.
>
> Some features not included in these specifications might not be rendered correctly in QGIS.

Símbolos de Linha

Appropriate for line geometry features, line symbols have the following symbol layer types:

Tabela10.2: List of line symbol layer types

Simple line	*Arrow*	*Filled line*
Geometry generator	*Hashed line*	*Interpolated line*
Linear referencing	*Lineburst*	*Marker line*
Raster line		

- **Linha simples** (padrão)

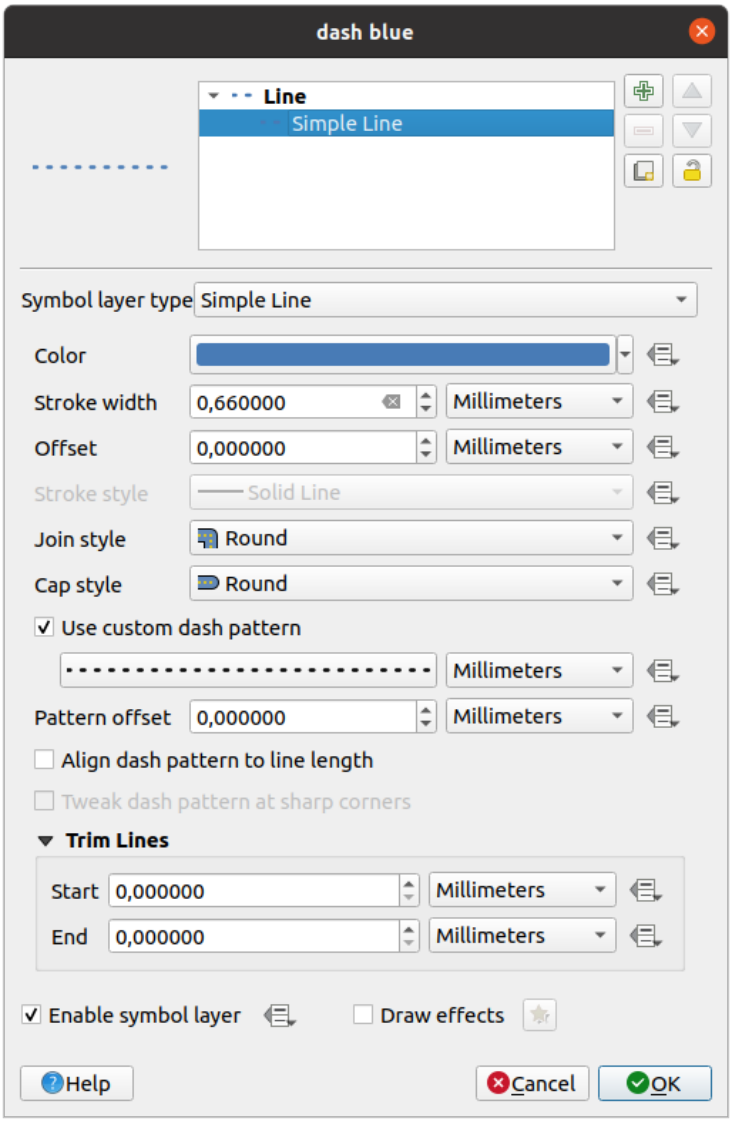


Figura10.12: Projetando um símbolo de linha simples

The simple line symbol layer type has many of the same properties as the *simple marker symbol*, and in addition:

- *Use custom dash pattern*: overrides the *Stroke style* setting with a custom dash. You would need to define length of consecutive dashes and spaces shaping the model, in the chosen unit. The total length of the pattern is displayed at the bottom of the dialog.

  - *Pattern offset*: the positioning of the dashes/spaces in the line can be tweaked, so that they can be placed at nicer positions to account for corners in the line (also can be used potentially to "align" adjacent dash pattern borders)
  - *Align dash pattern to line length*: the dash pattern length will be adjusted so that the line will end with a complete dash element, instead of a gap.
  - *Tweak dash pattern at sharp corners*: dynamically adjusts the dash pattern placement so that sharp corners are represented by a full dash element coming into and out of the sharp corner. Dependent on *Align dash pattern to line length*.
  - *Trim lines* from *Start* and/or *End*: allows for the line rendering to trim off the first x mm and last y mm from the actual line string when drawing the line. It supports a range of *units*, including percentage of the overall line length, and can be data defined for extra control. The start/end trim distance can be used e.g. when creating complex symbols where a line layer should not overlap marker symbol layers placed at the start and end of the line.

- **Arrow**: draws lines as curved (or not) arrows with a single or a double head with configurable (and data-defined):
  - *Head type*
  - *Arrow type*
  - *Arrow width*
  - *Arrow width at start*
  - *Head length*
  - *Head thickness*
  - *Offset*

  It is possible to create *Curved arrows* (the line feature must have at least three vertices) and *Repeat arrow on each segment*. It also uses a *fill symbol* such as gradients or shapeburst to render the arrow body. Combined with the geometry generator, this type of layer symbol helps you representing flow maps.

- **Geometry generator** (see *O Gerador de Geometria*)

- **Interpolated line**: allows to render a line whose *Stroke width* and/or *Color* may be constant (given a *Fixed width* and *Single color* parameters) or vary along the geometry. When varying, necessary inputs are:
  - guilabel:*valor inicial* e *valor final*: Valores que serão utilizados para interpolação nas extremidades da geometria das características. Eles podem ser valores fixos, atributos da característica ou baseados em uma expressão.
  - guilabel:*Valor mínimo* e *Valor máximo*: Valores entre os quais a interpolação é feita. Pressione o botão **|recarregar|** Carregar para preenchê-los automaticamente com base nos valores mínimos e máximos de início/fim aplicados à camada.
  - Only available for the stroke option:
    * *Min. width* and *Max. width*: define the range of the varying width. *Min. width* is assigned to the *Min. value* and *Max. width* to the *Max. value*. A *unit* can be associated.
    * *Use absolute value*: only consider absolute value for interpolation (negative values are used as positive).
    * *Ignore out of range*: by default, when the `[start value - end value]` range of a feature is not included in the `[min. value - max. value]` range, the out-of-bounds parts of the feature's geometry are rendered with the min or max width. Check this option to not render them at all.
  - For varying color, you can use any of the interpolation methods of *color ramp classification*

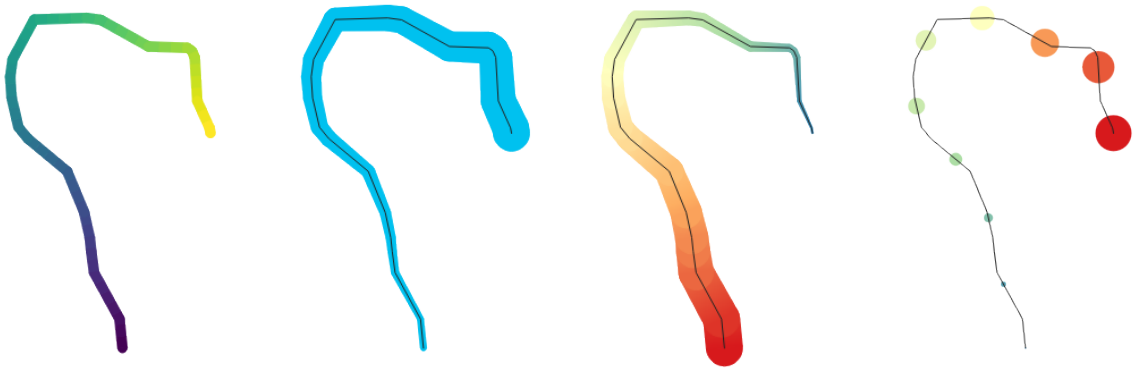


Figura10.13: Exemplos de linhas interpoladas

- **Marker line**: repeats a *marker symbol* over the length of a line.
  - The *Marker placement* can be set using a regular interval setting (starting from the first vertex) and/or the line geometry property (on first or last vertex, inner vertices, the central point of the line or of each segment, or on every curve point).
  - When first or last vertex placement is enabled, the *Place on every part extremity* option will make the markers render also at the first or last vertex for every part of multipart geometries.
  - *Offset along line*: the markers placement can also be given an offset along the line, in the ref:*unit <unit_selector>* of your choice (millimeters, points, map unit, meters at scale, percentage, …):
    - A positive value offsets the markers symbols in the line direction (with *On first vertex* and *With interval* placements) and backwards (with *On last vertex* placement).
    - A negative value on a not closed line will result in no offset (for *On first vertex* and *On last vertex* placements) or backwards offset of the symbols (from the last vertex).
    - With a closed ring, QGIS treats the offset as continuing to loop around the ring (forward or backward). E.g. setting the offset to 150% (resp. -10% or -110%) results in the offset being treated as 50% (resp. 90%) of the length of the closed ring.
  - The *Rotate marker to follow line direction* option sets whether each marker symbol should be oriented relative to the line direction or not.

    Because a line is often a succession of segments of different directions, the rotation of the marker is calculated by averaging over a specified distance along the line. For example, setting the *Average angle over* property to `4mm` means that the two points along the line that are `2mm` before and after the symbol placement are used to calculate the line angle for that marker symbol. This has the effect of smoothing (or removing) any tiny local deviations from the overall line direction, resulting in much nicer visual orientations of the marker line symbols.
  - *Line offset*: the marker symbols can also be offset from the line feature.
- **Hashed line**: repeats a line segment (a hash) over the length of a line symbol, with a line sub-symbol used to render each individual segment. In other words, a hashed line is like a marker line in which marker symbols are replaced with segments. As such, the hashed lines have the *same properties* as marker line symbols, along with:
  - *Hash length*
  - *Hash rotation*

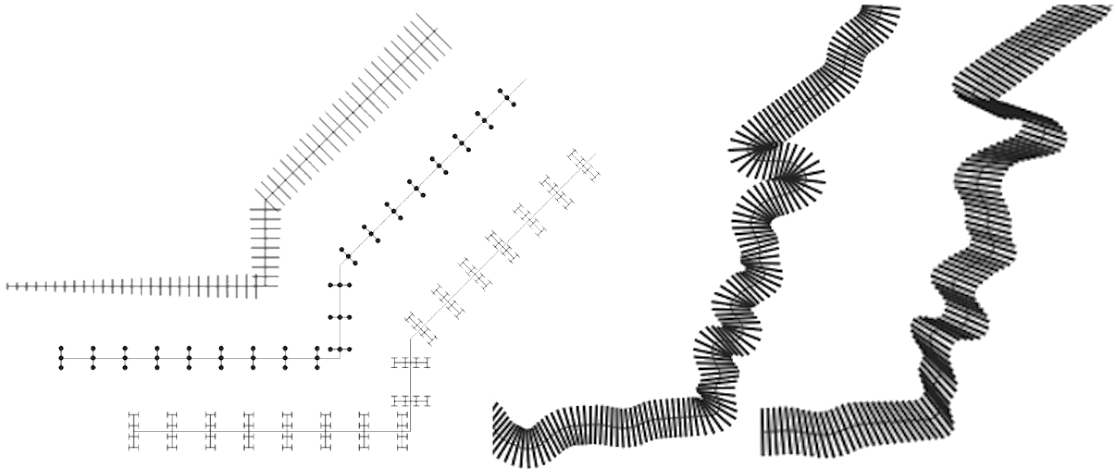


Figura10.14: Examples of hashed lines

- **Raster line**: renders and repeats a raster image following the length of a line feature shape. The *Stroke width*, *Offset*, *Join style*, *Cap style* and *Opacity* can be adjusted.

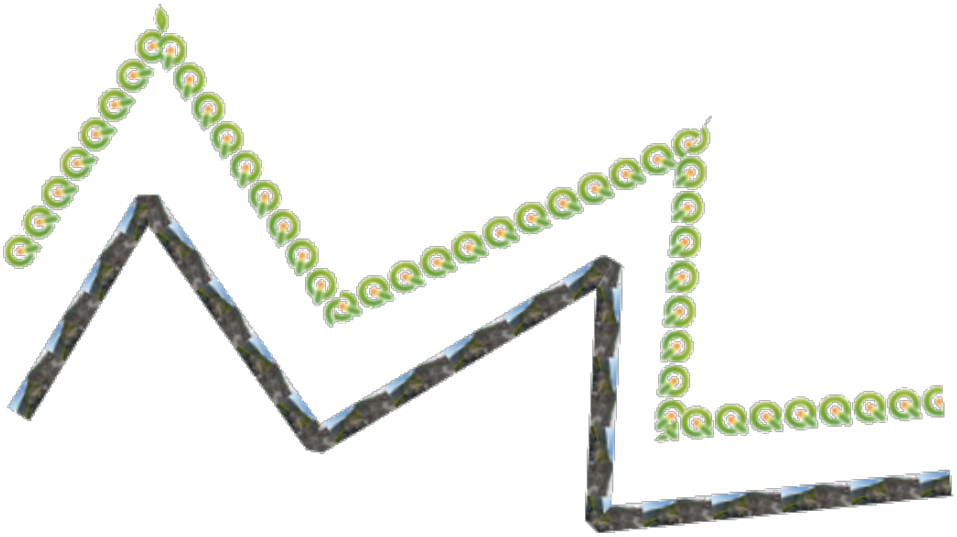


Figura10.15: Examples of raster lines

- **Linear referencing**: allows placing text labels at regular intervals along a line, or at positions corresponding to existing vertices. Positions can be calculated using Cartesian distances, or interpolated from z/m values. Parameters include:
  - *Measure placement*: Labels can be placed at regular cartesian 2d distances, at regular linearly interpolated spacing calculated using the Z or M values in geometries, or at existing vertices. A fixed or data-defined *Interval* is given.
  - *Quantity*: Label displayed at each position represent the running distance along the line, or the linearly interpolated Z or M value. Labels are rendered using the full range of *text* and *number* formatting functionalities available in QGIS.

> **Nota**
>
> When using the distance-based placement or labels, the distances are calculated using 2D only. Cartesian calculations are based on the original layer CRS.

- *Skip multiples of*: If set, labels which are a multiple of this value will be skipped over. This allows construction of complex referencing labels, e.g. where a symbol has two linear referencing symbol layers, one set to label every 100m in a small font, skipping multiples of 1000, and a second set to label every 1000m in a big bold font.
- *Average angle over*: Labels are rendered using an angle calculated by averaging the linestring, so sharp tiny jaggies don't result in unslightly label rotation.
- *Show marker symbols*, at referenced points in the line feature, using a full QGIS marker symbol. This allows e.g. showing a cross-hatch at the labeled point, for a "ruler" style line.

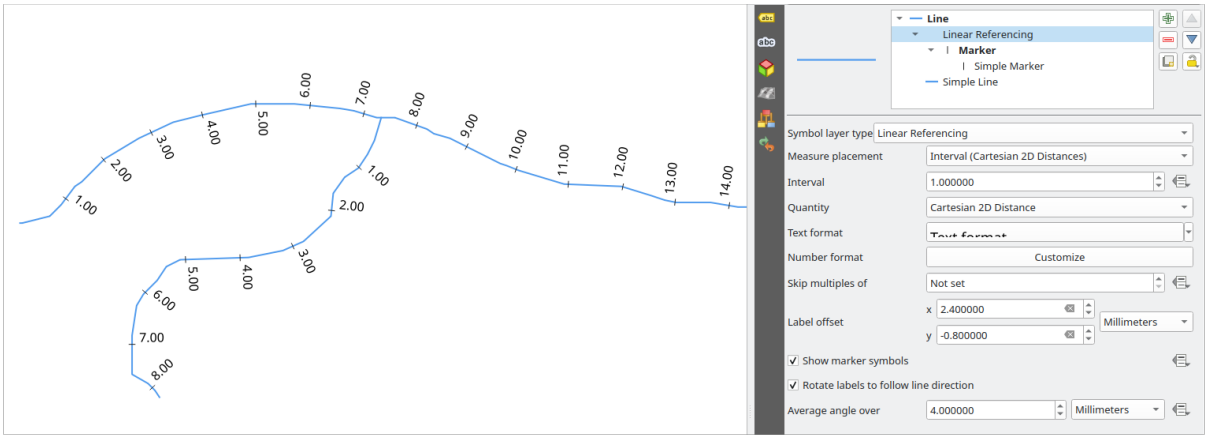


Figura10.16: Drawing linear labels along line feature

- **Lineburst**: renders a gradient along the width of a line. You can choose between *Two color* or *Color ramp* and the *Stroke width*, *Offset*, *Join style*, *Cap style* can be adjusted.

Figura10.17: Examples of lineburst lines

- **Filled line**: renders the interior of the lines using a fill symbol allowing for lines filled with gradients, line hatches, etc. The *Stroke width*, *Offset*, *Join style*, *Cap style* can be adjusted.

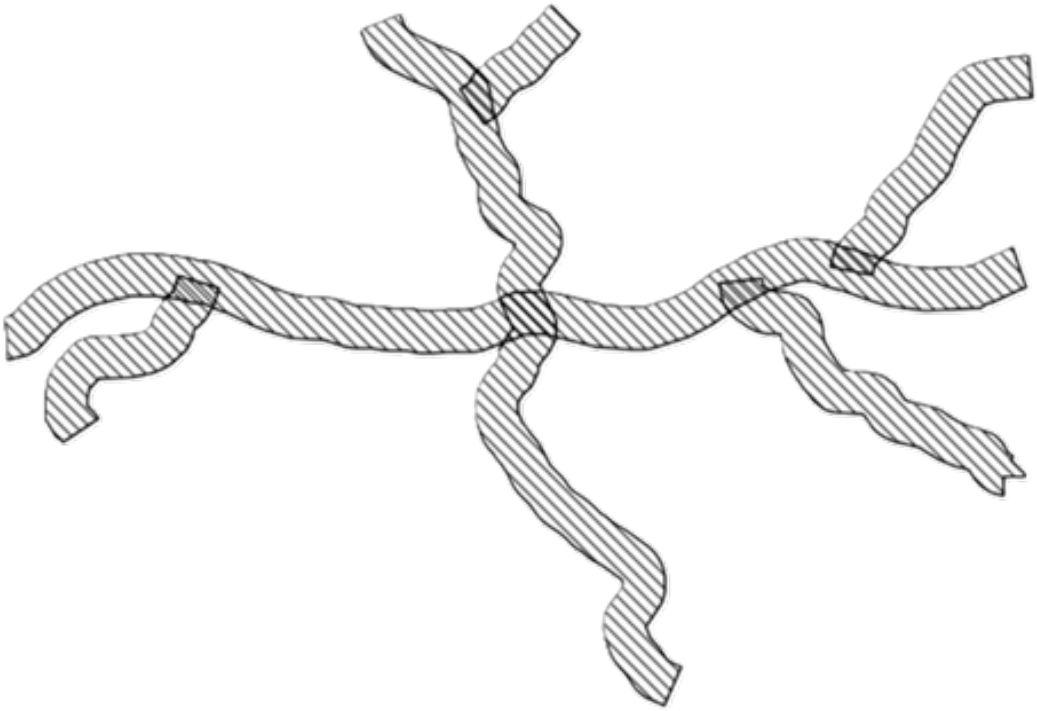


Figura10.18: Examples of filled lines

Símbolos de Preenchimento

Appropriate for polygon geometry features, fill symbols have also several symbol layer types:

Tabela10.3: List of fill symbol layer types

For polygon interior	For polygon ring
Simple fill	*Outline: Arrow*
Centroid fill	*Outline: Filled line*
Geometry generator	*Outline: Hashed line*
Gradient fill	*Outline: Interpolated line*
Line pattern fill	*Outline: Linear referencing*
Point pattern fill line	*Outline: Lineburst*
Random marker fill	*Outline: Marker line*
Raster image	*Outline: Raster line*
SVG fill	*Outline: Simple line*
Shapeburst fill	

- **Simple fill** (default): fills a polygon with a uniform color

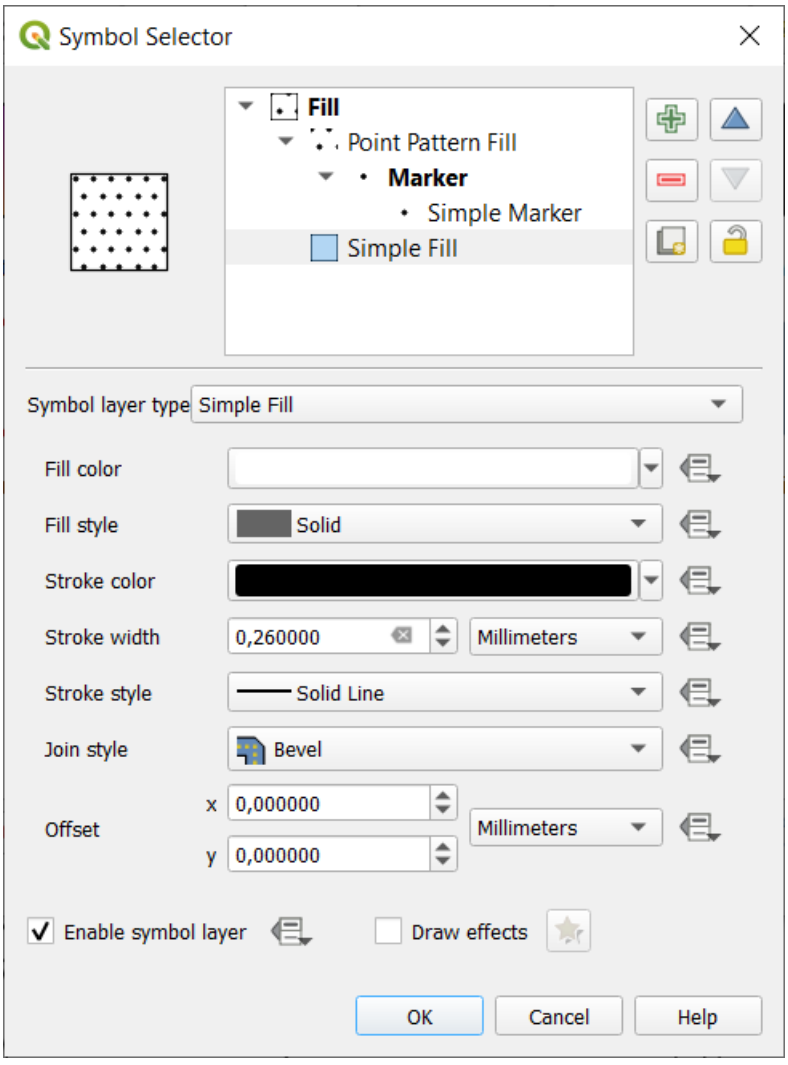


Figura10.19: Designing a Simple Fill Symbol

- **Centroid fill**: places a *marker symbol* at the centroid of the visible feature. The position of the marker may

not be the real centroid of the feature, because calculation takes into account the polygon(s) clipped to area visible in map canvas for rendering and ignores holes. Use the *geometry generator symbol* if you want the exact centroid.

Você pode:

- guilabel: 'Força o posicionamento de marcadores dentro de polígonos'.
- *Draw markers on every part of multi-part features* or place the point only on its biggest part
- display the marker symbol(s) in whole or in part, keeping parts overlapping the current feature geometry (*Clip markers to polygon boundary*) or the geometry part the symbol belongs to (*Clip markers to current part boundary only*)

- **Geometry generator** (see *O Gerador de Geometria*)
- **Gradient fill**: uses a radial, linear or conical gradient, based on either simple two color gradients or a predefined *gradient color ramp* to fill polygons. The gradient can be rotated and applied on a single feature basis or across the whole map extent. Also start and end points can be set via coordinates or using the centroid (of feature or map). A data-defined offset can be defined.
- ** Preenchimento padrão de linha**: preenche o polígono com um padrão de preenchimento camada de símbolo de linha. Você pode definir:
  - *Alignment*: defines how the pattern is positioned relative to the feature(s):
    - *Align pattern to feature*: lines are rendered within each feature
    - *Align pattern to map extent*: a pattern is rendered over the whole map extent, allowing lines to align nicely across features
  - *Rotação* das linhas, no sentido anti-horário
  - *Espaçamento*: distância entre linhas consecutivas
  - *Offset* distance of the lines from the feature boundary
  - *Clipping*: allows to control how lines in the fill should be clipped to the polygon shape. Options are:
    - *Clip During Render Only*: lines are created covering the whole bounding box of the feature and then clipped while drawing. Line extremities (beginning and end) will not be visible.
    - *Clip Lines Before Render*: lines are clipped to the exact shape of the polygon prior to rendering. Line extremities (including cap styles, start/end marker line objects, …) will be visible, and may sometimes extend outside of the polygon (depending on the line symbol settings).
    - *No Clipping*: no clipping at all is done - lines will cover the whole bounding box of the feature

> **Nota**
>
> When exporting maps to vector formats such as PDF or SVG, or when the line symbol in the pattern has data-defined properties, QGIS renders the line pattern fill using a line-by-line vector-based approach. This results in smaller output file sizes, higher quality exports, and improved editability of the pattern lines in external vector graphics applications.

- **Point pattern fill**: fills the polygon with a grid pattern of *marker symbol*. You can set:
  - *Alignment*: defines how the pattern is positioned relative to the feature(s):
    - *Align pattern to feature*: marker lines are rendered within each feature
    - *Align pattern to map extent*: a pattern is rendered over the whole map extent, allowing markers to align nicely across features

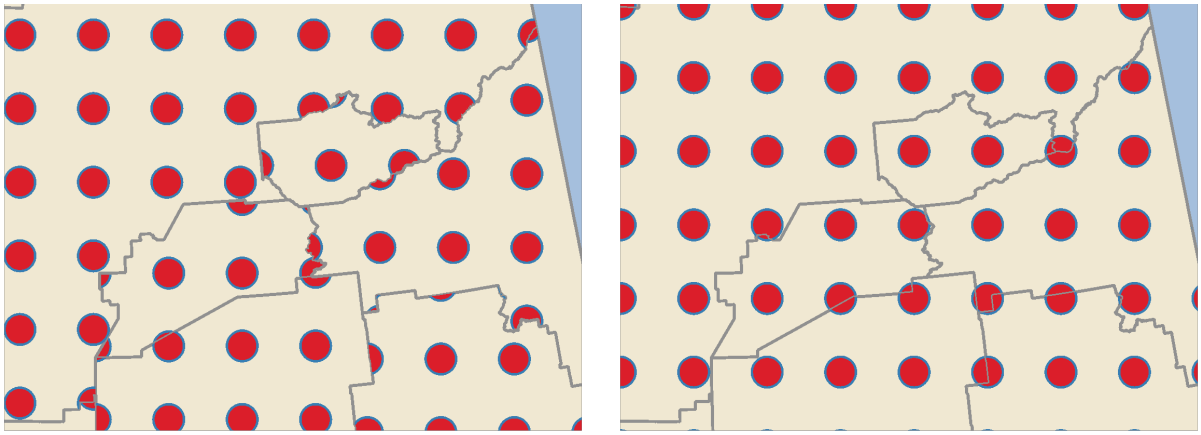


Figura10.20: Aligning point pattern to feature (left) and to map extent (right)

- *Distância*: *Horizontal* e *Vertical* distâncias entre marcadores consecutivos
- *Displacement*: a *Horizontal* (resp. *Vertical*) offset of alignment between consecutive markers in a column (resp. in a row)
- *Offset*: *Horizontal* and *Vertical* distances from the feature boundary
- *Clipping*: allows to control how markers in the fill should be clipped to the polygon shape. Options are:
  - *Clip to shape*: markers are clipped so that only the portions inside the polygon are visible
  - *Marker centroid within shape*: only markers where the center of the marker falls inside the polygon are drawn, but these markers won't be clipped to the outside of the polygon
  - *Marker completely within shape*: only markers which fall completely within the polygon are shown
  - *No clipping*: any marker which intersects at all with the polygon will be completely rendered (strictly speaking its the "intersects with the bounding box of the marker")

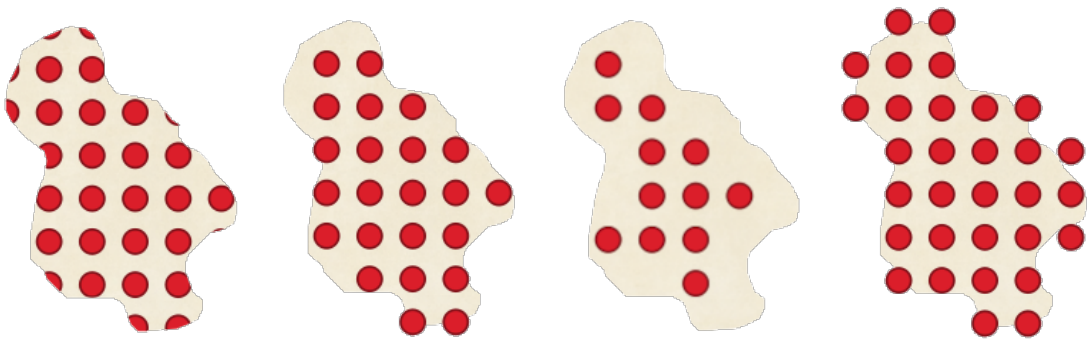


Figura10.21: Clipping markers in fill - From left to right: Clip to shape, Marker centroid within shape, Marker completely within shape, No clipping

- *Rotation* of the whole pattern, clockwise
- The *Randomize pattern* group setting allows each point in a point pattern fill to be randomly shifted up to the specified maximum distance *Horizontally* or *Vertically*. You can specify the maximum offset in any supported units, such as millimeters, points, map units, or even "percentage" (where percentage is relative to the pattern width or height).

  You can set an optional random number seed to avoid the symbol patterns "jumping" around between map refreshes. Data defined overrides are also supported.

> **Nota**
>
> The main difference between the *Randomize pattern* and the *random marker fill* symbol type is that the random offset with a point pattern allows for quasi-"regular" placement of markers – because the points in the pattern are effectively constrained to a grid, this allows creation of semi-random fills which don't have empty areas or overlapping markers. (As opposed to the random marker fill, which will always place points completely randomly… sometimes resulting in visual clusters of points or unwanted empty areas).

- **Random marker fill**: fills the polygon with a *marker symbol* placed at random locations within the polygon boundary. You can set:
  - *Count method*: whether the number of marker symbols to render is considered as an absolute count or density-based
  - *Point count*: the number of marker symbols to render,
  - an optional random number *seed*, to give consistent placement
  - *Área de densidade*: no caso do método de contagem baseado em densidade, garante que a densidade de preenchimento dos marcadores permaneça a mesma em diferentes níveis de escala/zoom dos marcadores sempre que os mapas são atualizados (também permite a colocação aleatória para jogar bem com servidor QGIS e renderização baseada em mosaico)
  - *Clip markers to polygon boundary*: whether markers rendered near the edges of polygons should be clipped to the polygon boundary or not
- **Raster image fill**: fills the polygon with tiles from a raster image (`PNG` `JPG`, `BMP` …). The image can be a file on the disk, a remote URL or an embedded file encoded as a string (*more details*). Options include (data defined) opacity, size, coordinate mode (object or viewport), rotation and offset. Under the option Size you can independently adjust the width and height of the fill pattern, enabling stretched raster fills in either the horizontal or vertical directions. The image width and height can be set using any of the *common units* or as a percentage of the original size.
- **SVG fill**: fills the polygon using *SVG markers* of a given size (*Texture width*).
- **Shapeburst fill**: buffers a gradient fill, where a gradient is drawn from the boundary of a polygon towards the polygon's centre. Configurable parameters include distance from the boundary to shade, use of color ramps or simple two color gradients, optional blurring of the fill and offsets.
- **Outline: Arrow**: uses a line *arrow symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The settings for the outline arrow are the same as for arrow line symbols.
- **Outline: Filled line**: uses a *filled line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The settings are the same as for filled line symbols.
- **Outline: Hashed line**: uses a *hash line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for hashed line symbols.
- **Outline: Interpolated line**: uses an *interpolated line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for interpolated line symbols.
- **Outline: Linear referencing**: uses a *linear referencing symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for linear referencing symbols.
- **Outline: Lineburst**: uses a *lineburst symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings are the same as for lineburst symbols.

- **Outline: Marker line**: uses a *marker line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings for the outline marker line are the same as for marker line symbols.
- **Outline: Raster line**: uses a *raster line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The other settings for the outline hashed line are the same as for hashed line symbols.
- **Outline: Simple line**: uses a *simple line symbol* layer to represent the polygon boundary (*Rings*) which can be the interior rings only, the exterior ring only or all the rings. The *Draw line only inside polygon* option displays the polygon borders inside the polygon and can be useful to clearly represent adjacent polygon boundaries. The other settings for the outline simple line are the same as for simple line symbols.

Nota

When geometry type is polygon, you can choose to disable the automatic clipping of lines/polygons to the canvas extent. In some cases this clipping results in unfavourable symbology (e.g. centroid fills where the centroid must always be the actual feature's centroid).

SVG parametrizável

You have the possibility to change the colors of a *SVG marker*. You have to add the placeholders `param(fill)` for fill color, `param(fill-opacity)` for fill opacity, `param(outline)` and `param(outline-opacity)` for stroke color and opacity respectively, and `param(outline-width)` for stroke width. These placeholders can optionally be followed by a default value, e.g.:

```
<svg width="100%" height="100%">
<rect fill="param(fill) #ff0000" fill-opacity="param(fill-opacity) 1" stroke=
↪"param(outline) #00ff00" stroke-opacity="param(outline-opacity) 1" stroke-width=
↪"param(outline-width) 10" width="100" height="100">
</rect>
</svg>
```

More generally, SVG can be freely parametrized using `param(param_name)`. This param can either be used as an attribute value or a node text:

```
<g stroke-width=".265" text-anchor="middle" alignment-baseline="param(align)">
  <text x="98" y="147.5" font-size="6px">param(text1)</text>
  <text x="98" y="156.3" font-size="4.5px">param(text2)</text>
</g>
```

The parameters can then be defined as expressions in the *Dynamic SVG parameters* table.

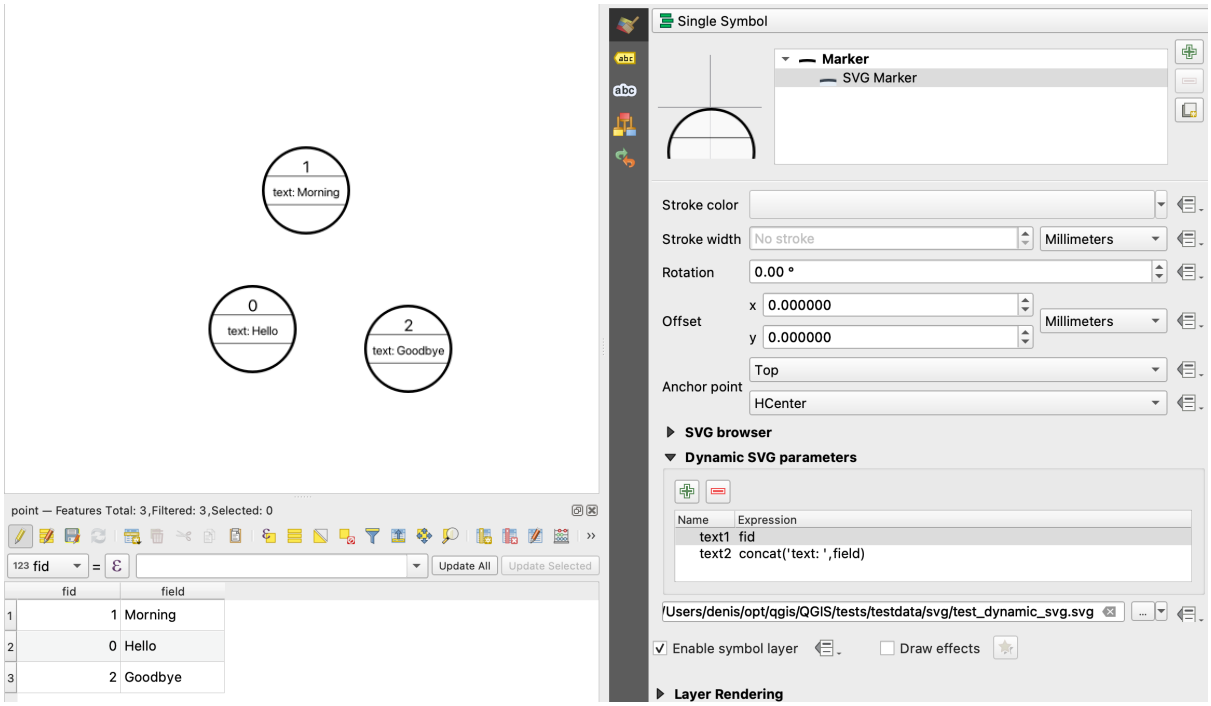


Figura10.22: Tabela de parâmetros dinâmicos SVG

Nota

QGIS is looking for a complete SVG node. So if your parameter is within a more complex node, you need to inject the complete node with the expression. For instance, `transform="rotate(param(angle)"` will not work. Instead, you need to do `transform="param(rotation)"` and `rotation` parameter will be defined with the expression `'rotate(' || coalesce(my_field, 0) || ')'`.

O Gerador de Geometria

Available with all types of symbols, the *geometry generator* symbol layer allows to use *expression syntax* to generate a geometry on the fly during the rendering process. The resulting geometry does not have to match with the original *Geometry type* and you can add several differently modified symbol layers on top of each other.

A *Units* property can be set: when the geometry generator symbol is not applied to a layer (e.g., it is used on a layout item), this allows more control over the generated output.

Alguns exemplos:

- Render symbol as the centroid of a feature

```
centroid( $geometry )
```

- Visually overlap features within a 100 map units distance from a point feature, i.e generate a 100m buffer around the point

```
buffer( $geometry, 100 )
```

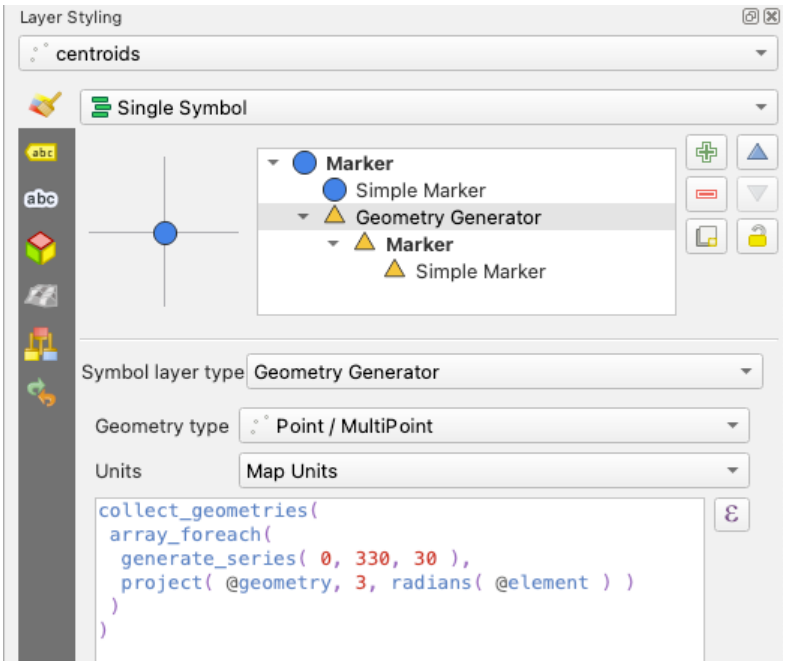
- Create a radial effect of points surrounding the central feature point when used as a MultiPoint geometry generator

```
collect_geometries(
  array_foreach(
```

(continua na próxima página)

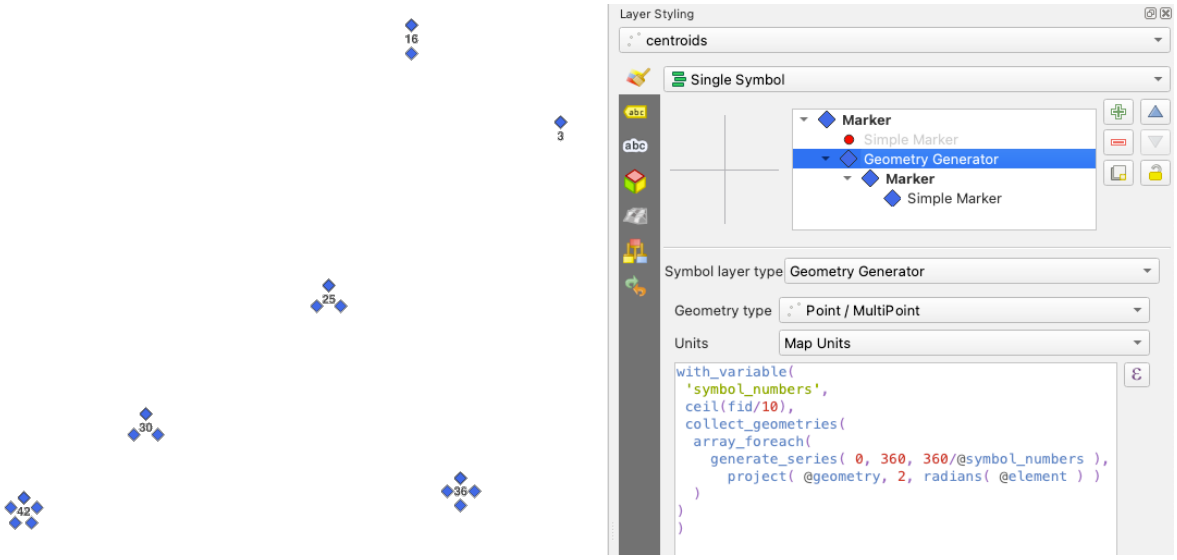
(continuação da página anterior)

```
    generate_series( 0, 330, 30 ),
    project( $geometry, 3, radians( @element ) )
  )
)
```



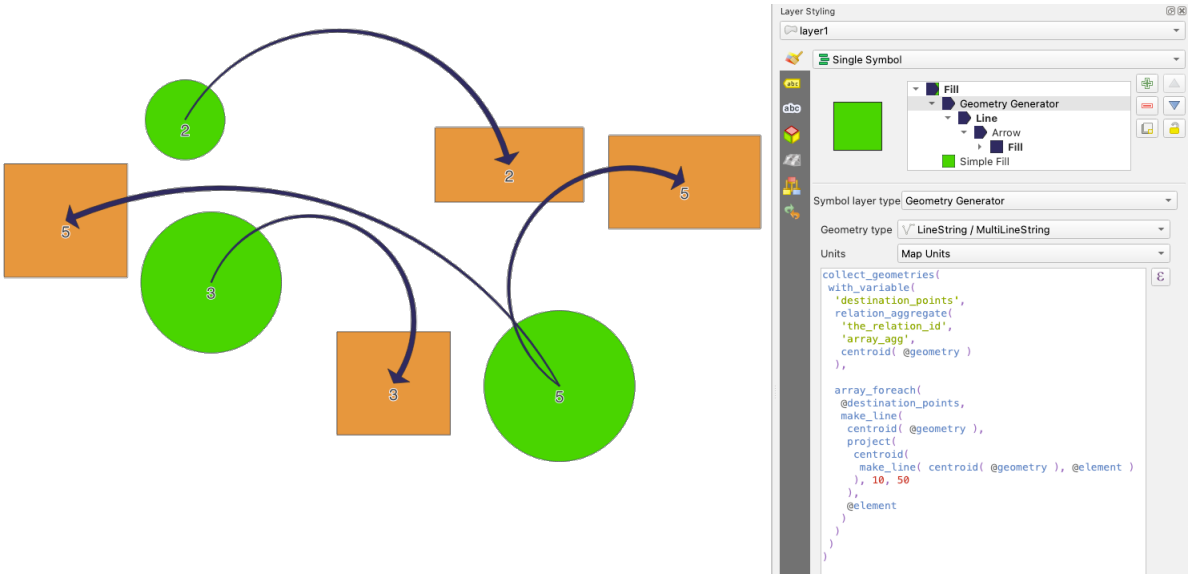
- Create a radial effect of points surrounding the central feature point. The number of points varies based on a field.

```
with_variable(
  'symbol_numbers',
  ceil(fid/10),
  collect_geometries(
    array_foreach(
      generate_series( 0, 360, 360/@symbol_numbers ),
      project( $geometry, 2, radians( @element ) )
    )
  )
)
```



- Create a curved arrow line connecting features of two layers based on their *relation*

```
collect_geometries(
  with_variable(
    'destination_points',
    relation_aggregate( 'the_relation_id', 'array_agg', centroid( $geometry ) ),
    array_foreach(
      @destination_points,
      make_line(
        centroid( @geometry ),
        project(
          centroid( make_line( centroid( @geometry ), @element ) ),
          10,
          50
        ),
        @element
      )
    )
  )
)
```



The Vector Field Marker

The vector field marker is used to display vector field data such as earth deformation, tidal flows, and the like. It displays the vectors as lines (preferably arrows) that are scaled and oriented according to selected attributes of data points. It can only be used to render point data; line and polygon layers are not drawn by this symbology.

O campo vetorial é definido por atributos nos dados, que podem representar o campo ou por:

- componentes **cartesianos** (componentes `x` e `y` do campo)
- ou coordenadas **polares**: neste caso, os atributos definem `Comprimento` e `Ângulo`. O ângulo pode ser medido no sentido horário a partir do norte ou no sentido anti-horário a partir do leste, e pode ser em graus ou radianos.
- or as **height only** data, which displays a vertical arrow scaled using an attribute of the data. This is appropriate for displaying the vertical component of deformation, for example.

A magnitude do campo pode ser aumentada ou diminuída para um tamanho apropriado para visualização do campo.

Animated marker

Animated marker symbol type allows you to use a `.GIF`, `.WebP`, `.MNG`, etc. animation file to represent points on your map. You can specify:

- *File* path,
- *Frame rate*: number of steps that are shown per second, indicating how fast the animation is played,
- *Size* in any *supported unit*,
- *Opacity*,
- *Rotation*,
- *Offset* in *x* and *y* directions from the marker position,
- *Anchor point*

There are two ways to handle animated symbols:

- **When your map is not configured as an animation** (i.e. it's a standard QGIS project without animations), the frame for the animated markers will be determined solely by the current timestamp.
- **When your map is** *configured as an animation*, the animated markers will sync with the animation's timeline. This means that animated markers will pause when the animation is paused, progress with the animation, and so forth. The map will also be redrawn according to the frame rate established for temporal animation. This mode is also applied when exporting an animation using the temporal controller.

10.3 Configurando um rótulo

Labels are textual information you can display on maps. They add details you could not necessarily represent using symbols, and may refer to vector features, raster cells, mesh elements, or simple annotations on the map… Two types of text-related items are available in QGIS:

- *Text Format*: defines the appearance of the text, including *font, size, colors*, *shadow*, *background*, *buffer*, …

  They can be used to render texts over the map (layout/map title, decorations, scale bar, …), usually through the *font* widget.

  Para criar um item *Formato de texto*:

  1. Open the *Style Manager* dialog

2. Ativar a aba *Formato de texto*

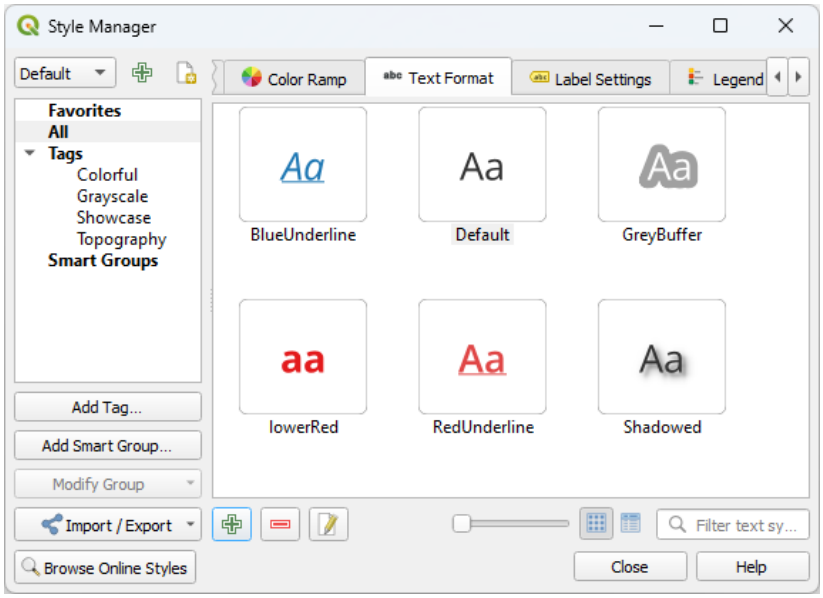


Figura10.23: Text formats in Style Manager dialog

3. Pressione o botão Adicionar item. O diálogo *Formato do texto* abre para *configuração*. Como de costume, estas propriedades são *dados definíveis*.

- *Label Settings*: extend the text format settings with properties related to the location or the interaction with other texts or features (*callouts*, *placement*, *overlay, scale visibility*, *mask* …).

  They are used to configure smart labelling for vector and mesh layers through the *Labels* tab of the vector or mesh *Layer Properties* dialog or *Layer Styling* panel or using the Layer Labeling Options button of the *Label toolbar*.

  Para criar um item :guilabel:`Configurações de rótulos’:

  1. Open the *Style Manager* dialog
  2. Ativar a aba :guilabel:`Configurações de rótulos’:

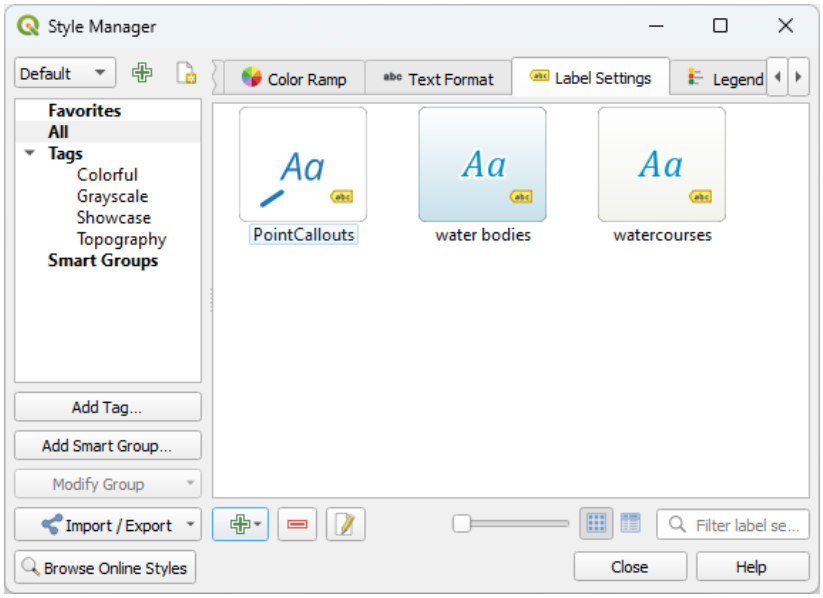


Figura10.24: Label Settings in Style Manager dialog

3. Pressione o menu Adicionar item e selecione a entrada correspondente ao tipo de geometria das características que você deseja rotular.

The *Label Settings* dialog opens with the following properties. As usual, these properties are *data-definable*.

10.3.1 Formatando o texto do rótulo

Whether you are configuring a *Text Format* or *Label Settings* item, and depending on the type of layer you are configuring, you will be given the following options:

Tabela10.4: Label properties by data type

Aba de propriedades	Formato de texto	Configurações de rótulos	Vector or mesh layer	raster layer
Texto	☑	☑	☑	☑
:guilabel:`Formatação'	☑	☑	☑	☑
Buffer	☑	☑	☑	☑
Máscara		☑	☑	
Plano de Fundo	☑	☑	☑	☑
Sombra	☑	☑	☑	☑
Chamada		☑	☑	
Posicionamento		☑	☑	☑
Renderização		☑	☑	☑

Atenção

While for legibility, “feature” is the name used below to indicate the item being labeled, depending on the underlying layer type, it can be replaced by “pixel”, “face” or “vertex”.

Aba de texto

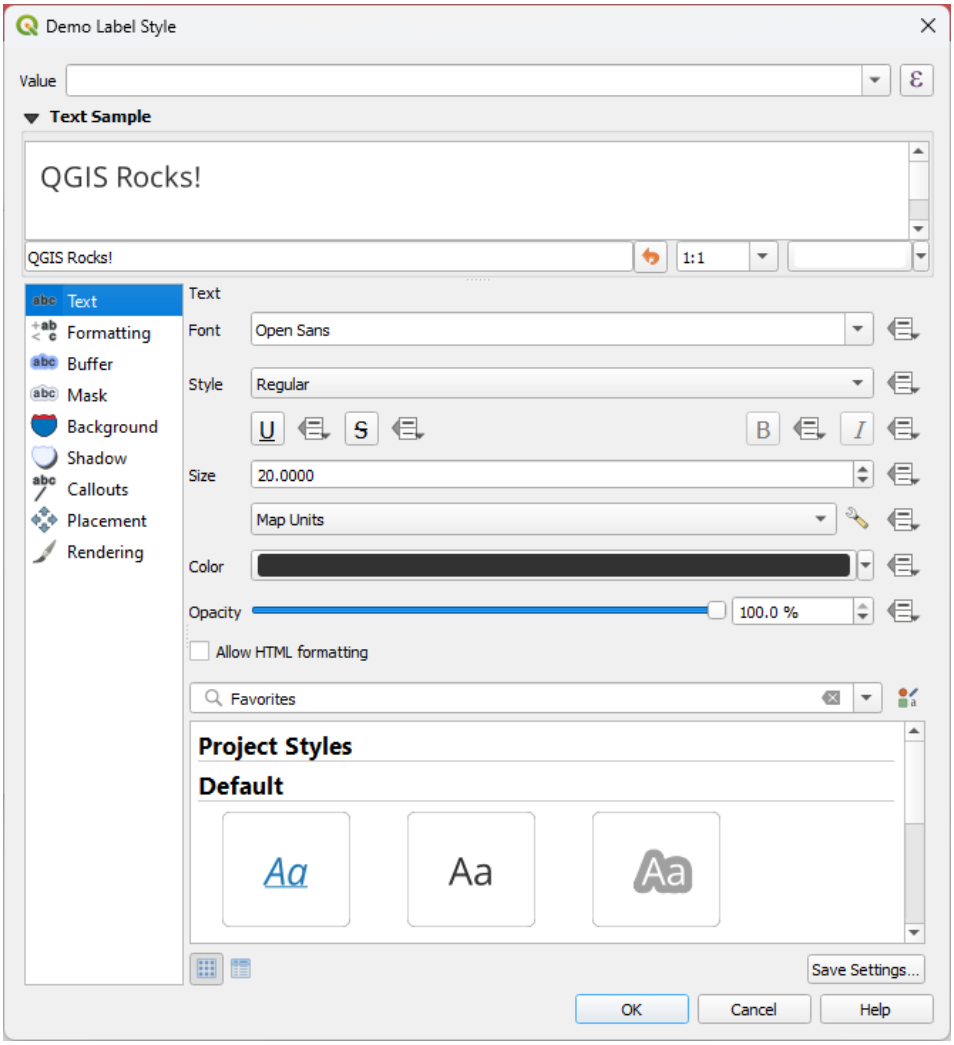


Figura10.25: Configurações de rótulos - Aba de texto

Na aba abc *Texto*, você pode definir:

- a *Fonte*, a partir das disponíveis em sua máquina
- o *Estilo*: juntamente com os estilos comuns da fonte, você pode definir se o texto deve ser sublinhado ou riscado através
- o *tamanho* em qualquer :ref:`unidade suportada `
- A *Cor*
- a *Opacidade*
- and *Allow HTML Formatting* enables the use of a subset of HTML tags and CSS rules to customize the label.

At the bottom of the tab, a widget shows a filterable list of compatible items stored in your *style manager database*. This allows you to easily configure the current text format or label setting based on an existing one, and also save a new item to the style database: Press the *Save format…* or *Save settings…* button and provide a name and tag(s).

Nota

Ao configurar um item *Configurações de rótulos*, os itens de formato de texto também estão disponíveis neste elemento. Selecione um para sobrescrever rapidamente as :ref:`propriedades textuais <text_format>' atuais da etiqueta. Da mesma forma, você pode criar/sobreescrever um formato de texto a partir daí.

Allow HTML Formatting

With *Allow HTML Formatting* enabled, you need to provide the HTML code in the *Value* field. Use whitespaces instead of tabs for any kind of indentation. The expression is parsed and any supported HTML tag overrides its corresponding setting in the labels properties. Because it is impossible to list and detail every HTML tag and CSS property that QGIS currently supports, we invite you to explore and test in your labels the ones supported by the underlying Qt library.

Examples of supported HTML tags:

- Text formatting, such as italic or bold, e.g.:

```
<i>QGIS</i> <b>rocks!</b>
```

- Superscript and subscript, where the text will be vertically $^{\text{super}}$ or $_{\text{sub}}$ aligned and automatically sized to 2/3 of the parent font size. You can also set a fixed font size for the superscript/subscript by including css rules, e.g.:

```
<sup style="font-size:33pt">my superscript text</sup>
```

- Text horizontal alignment, using either HTML the `align="xxx"` attribute or `center` tag. For HTML5 compatibility, prefer using the CSS property `text-align`.

> **Atenção**
>
> Horizontal alignment can not be used for curved labels.

- Structuring a block of text using header tags (such as `h1`, `h2`, ...), or paragraphs (with `p`, `div`, `br`):

```
<div class="myDiv">
  <h2>QGIS always rocks!!</h2>
  <p align="center">Let's dive into details of its nice features.</p>
</div>
```

- Image insertion: any image format readable by QGIS can be used in HTML label content. It can be served from local file paths, HTTP links, or base64 embedded content, using the `src="xxx"` attribute. Image sizes can be specified via the `width="##"` and `height="##"` attributes, in `points` unit. If width or height is not specified it will automatically be calculated from the original image size. Images are placed inline only, not as floating images, and not on curved text labels.

```
<img src="qgis.png" width=40 height=60>
```

Examples of supported CSS properties:

- Font properties (`color`, `font-family`, `font-size`, `font-weight`, `font-style`, `word-spacing`). Note that `word-spacing` will always use unit points.
- Text decorations such as underline, overline and line-through (`text-decoration`)
- Text alignment (`vertical-align`, `text-align`). Horizontal alignment can not be used for curved labels.
- Line height, in `points` or `percent` unit, e.g. “line-height: 40pt” or “line-height: 40%”
- Background properties such as `background-color` and `background-image`. They are supported for block type items (e.g. `div`) or inline items (e.g. `span`). For images, the CSS should be formatted as `background-image: url(xx)` and supports local file paths, HTTP links, or base64 embedded content.

> **Atenção**
>
> Backgrounds are not supported for curved text and are always rendered above any background shape for the label, and below drop shadows/buffers properties.

- Margin properties, available for block type items only, such as `div`, `p`, `h1`,... They can be specified in either the longhand or shorthand way, in `points` unit only. Negative values can be set for the bottom margin.

```
<div class="myDiv">
  <h2 style="margin-left: 5pt; margin-right: 10pt">QGIS still rocks...</h2>
  <p style="margin: 5pt 0pt -10pt 0pt">Thanks to you!!</p>
</div>
```

CSS properties can be set on HTML tags with the `style` attribute. The HTML tag `span` does not apply any formatting to text by itself and is ideal if you just want to apply CSS styling. A CSS property name and its value are separated by a colon (`:`). Multiple CSS properties are separated by semicolon (`;`), e.g.:

```
<span style="text-decoration:underline;text-align:center;color:blue;word-spacing:20
↪">I will be displayed as blue underlined and centered text with increased space␣
↪between words</span>
```

Abaixo um exemplo de uma expressão e renderização baseadas em HTML (aplica cores e sublinhados diferentes ao mesmo rótulo):

```
format(
  '<span style="color:blue">%1</span> ( <span style="color:red"><u>%2 ft</u></span>
↪ )',
  title( lower( "Name" ) ),
  round($length)
)
```

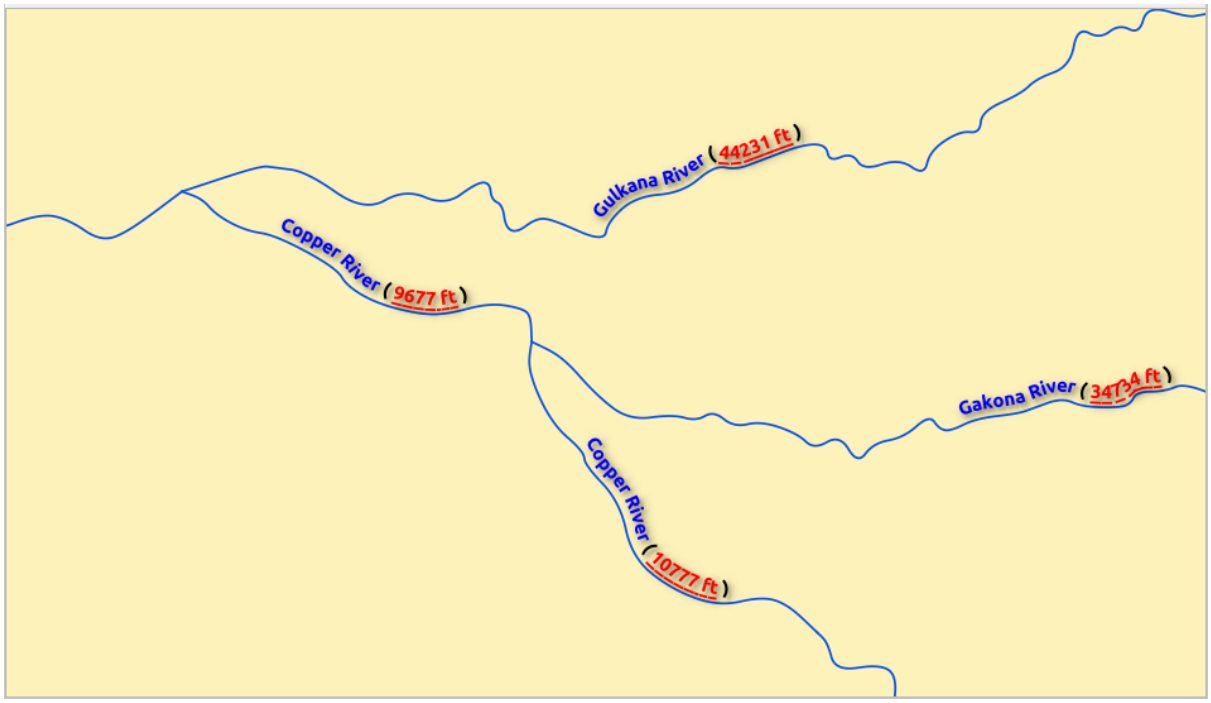


Figura10.26: Rotulagem com formatação HTML habilitada

Aba de formatação

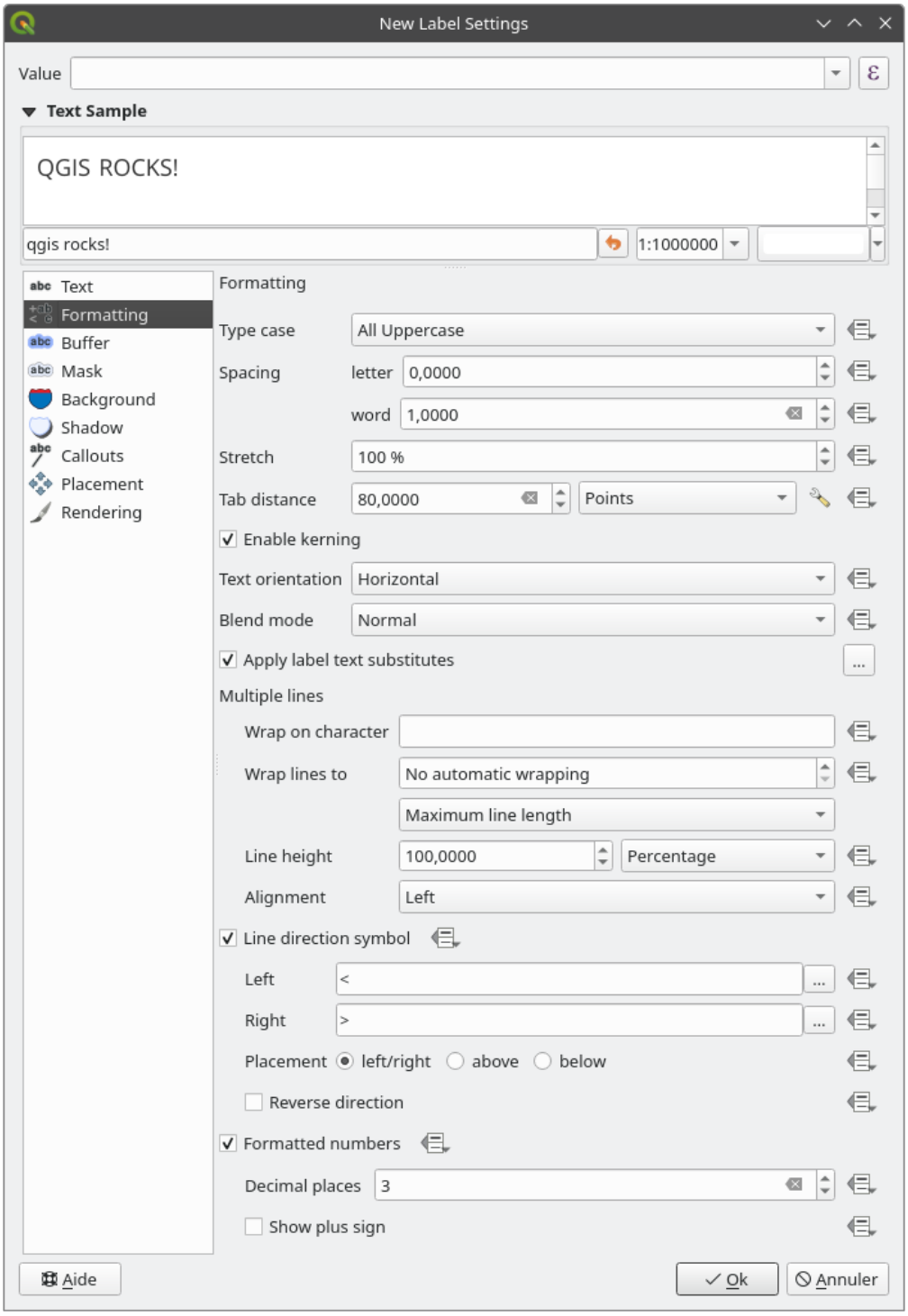


Figura10.27: Configurações de rótulos - Aba de formatação

Na aba :guilabelting:`Formatação', você pode:

- Utilize a opção *Tipo de formatação* para mudar o estilo de capitalização do texto. Você tem a possibilidade de renderizar o texto como:
  - *Sem mudança*

  - *Todas maiúsculas*
  - *Todas minúsculas*
  - *Small Caps*: renders lowercase characters as small caps.
  - *All Small Caps*: renders all characters as small caps regardless of their original case.
  - *Caso de letra maiúscula*: modifica a primeira letra de cada palavra em maiúscula, e transforma as outras letras em minúsculas se o texto original estiver utilizando um único tipo de caixa. No caso de casos de tipo misto no texto, as outras letras são deixadas intocadas.
  - guilabel:*Forçar a primeira letra maiúscula*: modifica a primeira letra de cada palavra em maiúsculas e deixa as outras letras no texto intocadas.
- Sob *Espaçamento*, mude o espaço entre as palavras e entre as letras individuais.
- With *Tab distance* you can adjust the replacement spacing for `Tab` characters in the label, in any supported unit. This allows e.g. to properly separate or vertically align elements of a layout legend text.

  Pressing the Configure tab stops button on the right, you can add a list of tab positions, instead of a single distance. This allows e.g. the creation of table-like multiline labels, where texts are split over different "columns" and lined up nicely, based on the `Tab` characters.
- *Stretch* ratio: allows text to be horizontally stretched or condensed by a factor. Handy for tweaking the widths of fonts to fit a bit of extra text into labels.
- *Habilitar o kerning* da fonte de texto
- Definir a *orientação do texto* que pode ser *Horizontal* ou *Vertical*. Também pode ser *Baseado na rotação* ao definir um rótulo (por exemplo, para características de linha de rótulos adequadas em :ref:`paralelo ` ao modo de posicionamento).
- Utilize a opção *Modo de mistura* para determinar como seus rótulos se misturarão com as características do mapa abaixo delas (mais detalhes em *Modos de Mistura*).
- The *Apply label text substitutes* option allows you to specify a list of texts to substitute to texts in feature labels (e.g., abbreviating street types). Replacement texts are used when displaying labels on the map. Users can also export and import lists of substitutes to make reuse and sharing easier.
- Configure *Múltiplas linhas*:
  - Set a character that will force a line break in the text with the *Wrap on character* option
  - Defina um tamanho de linha ideal para a quebra automática utilizando a opção :guilabel:`Quebrar linhas para'. O tamanho pode representar tanto o :guilabel:`Comprimento máximo da linha' ou o :guilabel:`Comprimento mínimo da linha'.
  - Decide the *Line Height*: values can be set to be in *Millimeters*, *Points*, *Pixels*, *Percentage*, or *Inches*. When line height is set to percentage it is the percentage of the default text line spacing of that font family. Typically 1.2 to 1.5 times the text size.
  - Formatar o *Alinhamento*: os valores típicos disponíveis são *Esquerda*, *Direita*, :guilabel:`Justificado' e :guilabel:`Centro'.

    When setting point labels properties, the text alignment can also be *Follow label placement*. In that case, the alignment will depend on the final placement of the label relative to the point. E.g., if the label is placed to the left of the point, then the label will be right aligned, while if it is placed to the right, it will be left aligned.

> **Nota**
>
> A formatação *Linhas múltiplas* ainda não é suportada por curvas baseadas em *posicionamento de etiquetas*. As opções serão então desativadas.

- For line labels you can include *Line direction symbol* to help determine the line directions, with symbols to use to indicate the *Left* or *Right*. They work particularly well when used with the *curved* or *Parallel* placement options from the *Placement* tab. There are options to set the symbols position, and to *Reverse direction*.
- Use the *Formatted numbers* option to format numeric texts. You can set the number of *Decimal places*. By default, 3 decimal places will be used. Use the *Show plus sign* if you want to show the plus sign for positive numbers.

Aba de buffer

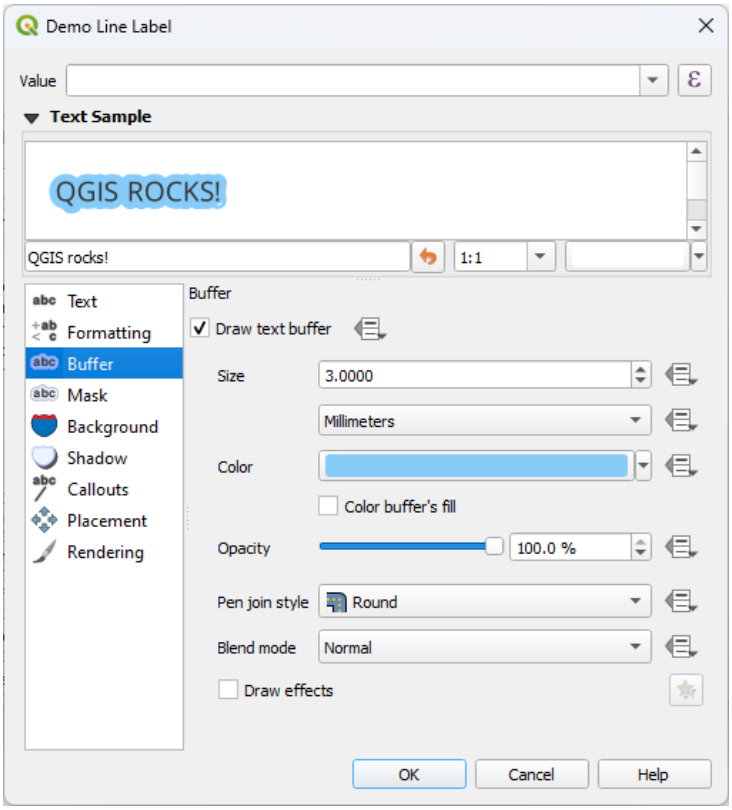


Figura10.28: Configurações de rótulos - Aba buffer

To create a buffer around the label, activate the *Draw text buffer* checkbox in the *Buffer* tab. Then you can:

- Set the buffer's *Size* in any *supported unit*
- Select the buffer's *Color*
- *Color buffer's fill*: The buffer expands from the label's outline, so, if the option is activated, the label's interior is filled. This may be relevant when using partially transparent labels or with non-normal blending modes, which will allow seeing behind the label's text. Unchecking the option (while using totally transparent labels) will allow you to create outlined text labels.
- Defina o deslocamento *Opacidade*
- Apply a *Pen join style*: it can be *Round*, *Miter* or *Bevel*
- Use the *Blend mode* option to determine how your label's buffer will mix with the map components below them (more details at *Modos de Mistura*).
- Check *Draw effects* to add advanced *paint effects* for improving text readability, eg through outer glows and blurs.

Aba de Plano de Fundo

The *Background* tab allows you to configure a shape that stays below each label. To add a background, activate the *Draw Background* checkbox and select the *Shape* type. It can be:

- uma forma regular, como *Retângulo*, *quadrado*, *Círculo* ou *Elípse* utilizando todas as propriedades de um símbolo de preenchimento.
- um símbolo *SVG* de um arquivo, um URL ou embutido no banco de dados do projeto ou estilo (*mais detalhes*)
- ou uma *Símbolo de marcador* você pode criar ou selecionar na *biblioteca de símbolos*.

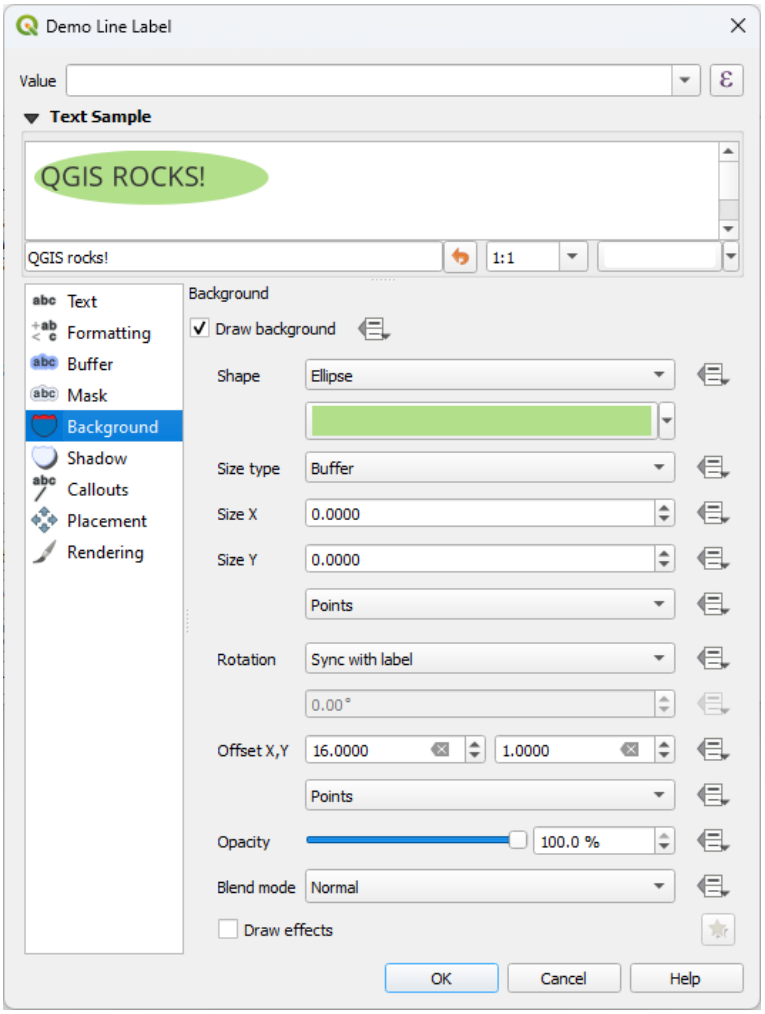


Figura10.29: Configurações de rótulos - Aba de Plano de Fundo

Dependendo da forma selecionada, você precisa configurar algumas das seguintes propriedades:

- O *Tipo de tamanho* da moldura, que pode ser:
  - *Fixado*: utilizando o mesmo tamanho para todas os rótulos, independentemente do tamanho do texto
  - or a *Buffer* over the text's bounding box
- O *tamanho* da moldura nas direções X e Y, utilizando qualquer :ref:`unidades suportadas `
- A *Rotation* of the background, between *Sync with label*, *Offset of label* and *Fixed*. The last two require an angle in degrees.
- Um *Deslocamento X,Y* para mudar o item de fundo nas direções X e/ou Y

- Um *Radius X,Y* para arredondar os cantos da forma do fundo (aplica-se somente a formas retangulares e quadradas)
- Uma: 'guilabel':`Opacidade' do plano de fundo
- A *Blend mode* to mix the background with the other items in the rendering (see *Modos de Mistura*).
- For SVG symbol, you can use its default properties (*Load symbol parameters*) or set a custom *Fill color*, *Stroke color* and *Stroke width*.
- *Draw effects* to add advanced *paint effects* for improving text readability, eg through outer glows and blurs.

Aba sombra

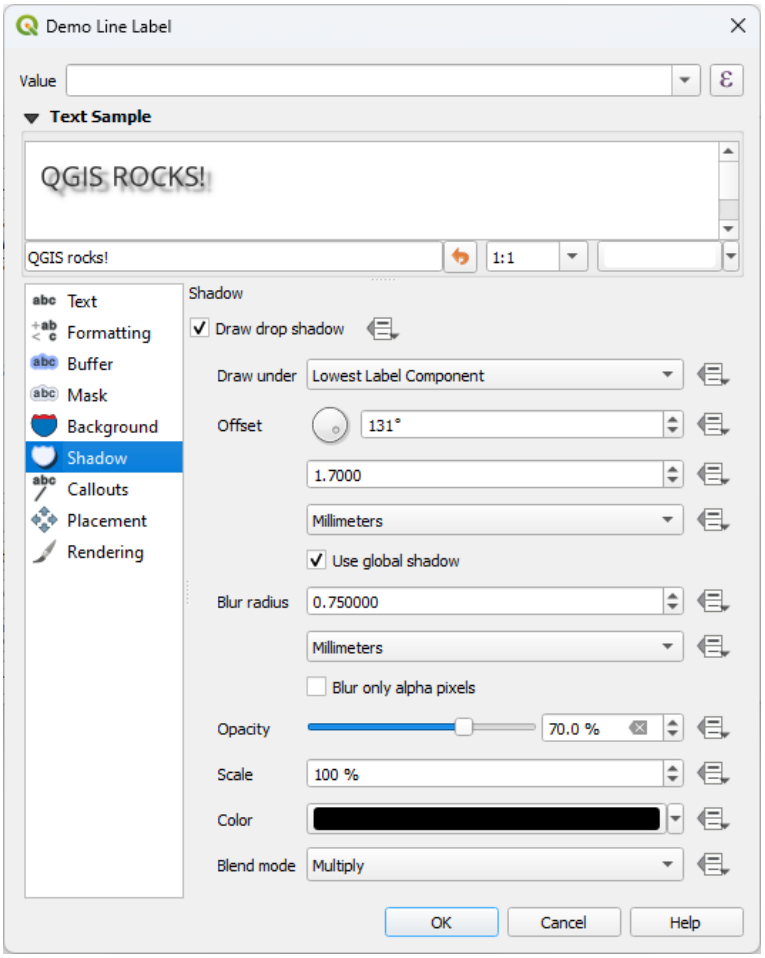


Figura10.30: Configurações de rótulo - guia Sombra

To add a shadow to the text, enable the *Shadow* tab and activate the *Draw drop shadow*. Then you can:

- Indicate the item used to generate the shadow with *Draw under*. It can be the *Lowest label component* or a particular component such as the *Text* itself, the *Buffer* or the *Background*.
- Set the shadow's *Offset* from the item being shadowed, ie:
  - O ângulo: no sentido horário, depende da orientação do item subjacente
  - The distance of offset from the item being shadowed
  - As unidades do deslocamento

Se você marcar a caixa *Usar sombra global* , então o ponto zero do ângulo é sempre orientado para o norte e não depende da orientação do item da etiqueta.

- Influenciar a aparência da sombra com o :guilabel:'Raio de embaçamento'. Quanto maior o número, mais suaves as sombras, nas unidades de sua escolha.
- Defina para a sombra :guilabel:`Opacidade'.
- Redimensionar o tamanho da sombra utilizando o fator *escala*
- Escolha a cor da sombra *Cor*
- Use the *Blend mode* option to determine how your label's shadow will mix with the map components below them (more details at *Modos de Mistura*).

10.3.2 Configurando a interação com rótulos

Além das configurações de formatação de texto expostas acima, você também pode definir como os rótulos interagem umas com as outras ou com as feições.

Aba de máscara

The *Mask* tab allows you to define a mask area around the labels. This feature is very useful when you have overlapping symbols and labels with similar colors, and you want to make the labels visible. A label mask prevents specified features from drawing within the boundary set for the mask. For example, you could set a label mask so that a specified layer does not draw within 2mm of the label, but allow features from another layer to still show. Label masks are similar to label buffers in that they allow control of the legibility of labels that cover other features. The label buffer draws on top of any underlying features, while the label mask selectively stops other layers from drawing.

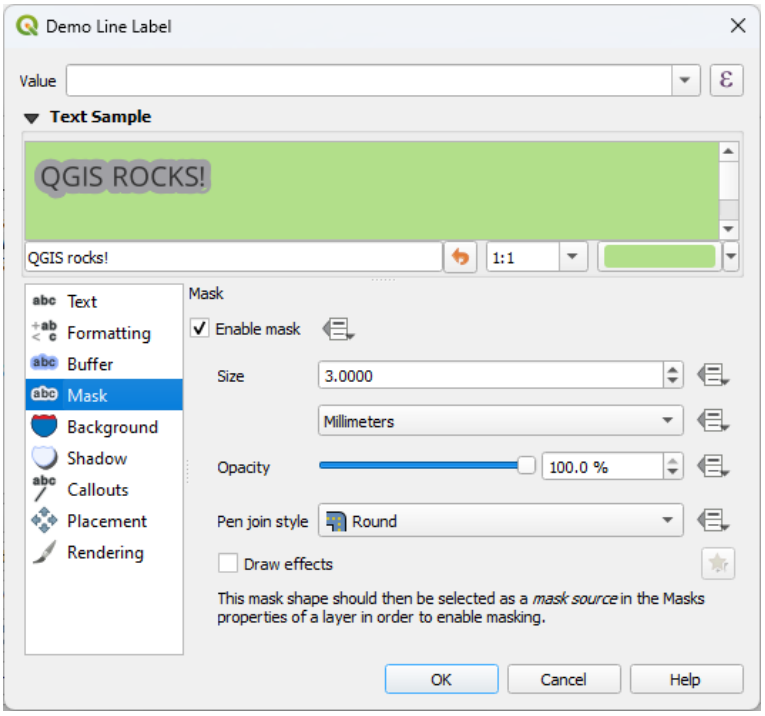


Figura10.31: Labels settings - Mask tab (with the text sample showing a green background representing another layer being excluded)

Para criar efeitos de mascaramento nos rótulos:

1. Activate the *Enable mask* checkbox in the tab.
2. Depois, você pode definir:
   - the mask's *Size* in the *supported units*
   - a: 'guilabel:`Opacidade' da área da máscara ao redor do rótulo
   - a *Pen Join Style*
   - *paint effects* through the *Draw effects* checkbox.
3. Select this mask shape as a mask source in the overlapping layer properties *Mask* tab (see *Propriedades das Máscaras*).

Aba de chamadas

A common practice when placing labels on a crowded map is to use **callouts** - labels which are placed outside (or displaced from) their associated feature are identified with a dynamic line connecting the label and the feature. If one of the two endings (either the label or the feature) is moved, the shape of the connector is recomputed.

Figura10.32: Rótulos com várias configurações de chamadas

To add a callout to a label, enable the *Callouts* tab and activate the *Draw callouts*. Then you can:

1. Selecione o *Estilo* de conector, um de:
   - *Linhas simples*: uma linha reta, o caminho mais curto
   - *Estilo Manhattan*: uma linha quebrada a 90°
   - *Linhas curvadas*: uma linha curvada
   - :guilabel:`Balões': uma bolha de fala que envolve o rótulo e aponta para a feição. Ela pode ter cantos arredondados.
2. Set the properties of the callout. The following table shows the different properties, with description and compatible connector style.

Tabela10.5: Label callout properties

Property	Style of callout	Descrição
Fill style	Balloons	A *fill symbol* with full display capabilities, including layer effects, data-defined settings, … for drawing the balloon shape.
Corner radius		Corner radius of the speech bubble
Wedge width		Sets how large the bubble speech connection with feature's pointer should be
Margins		Margins around the label's text, in the unit of your choice
Line style	All but balloons	A *line symbol* with full display capabilities, including layer effects, data-defined settings, … for drawing the connector line.
Curvature	Curved lines	The percentage of curvature of the connection line
Orientation		Orientation, starting from the label to the feature. It can be *Clockwise*, *Counter-clockwise*, or *Automatic* (determining an optimal orientation for each label).
Minimum length	All but balloons	Minimum length of the connector line
Draw lines to all feature parts		In case of a multi-part feature, indicates whether a connector line should be drawn from the label to each geometry part.
Label anchor point		Controls where the connector line should join to the label text. Available options: • *Ponto mais próximo* • *Centróide* • Fixed position at the edge (*Top left*, *Top center*, *Top right*, *Left middle*, *Right middle*, *Bottom left*, *Bottom center* and *Bottom right*).
Offset from label area		Controls the distance from the label anchor point (where the callout line ends). This avoids drawing lines right up against the text.
Offset from feature	Tudo	Controls the distance from the feature (or its anchor point if a polygon) where callout lines end. E.g., this avoids drawing lines right up against the edges of the features.
Feature anchor point		Where the connector line ends on the (polygon) feature. Available options: • *Pole of inaccessibility* • :guilabel:`Ponto no exterior' • :guilabel:`Ponto na superfície • *Centróide*
End point marker	Balloons	A *marker symbol* with full display capabilities including layer effects, data-defined, … for rendering a marker symbol below the endpoint of the balloon callout.
Blend mode	Tudo	Controls the *blending* of the callout.

Under the *Data defined placement* group, coordinates of the *Origin* (on the label side) and/or *Destination* (on the feature side) points of the callout can be controlled. Callouts can also be controlled manually by using the Move Label, Diagram or Callout tool in the *Labeling Toolbar*. The start and end points of each callout can be moved this way. The nodes should be highlighted when the mouse pointer is nearby. If needed the `Shift` Key can be held during the movement. This will snap the point in a way that the angle between the two callout points increments by 15 degrees.

Aba de posicionamento

Choose the *Placement* tab for configuring label placement and labeling priority. Note that the placement options differ according to the type of vector or mesh layer, namely point, line or polygon, and are affected by the global *PAL setting*.

Posicionamento para camadas de ponto

Os modos de posicionamento de rótulos de camadas de ponto disponíveis são:

- :guilabel:`Cartográfico': os rótulos pontuais são gerados com uma melhor relação visual com a feição pontual, seguindo as regras ideais de posicionamento cartográfico. Os rótulos podem ser colocados:
  - at a set *Distance* in *supported units*, either from the point feature itself or from the bounds of the symbol used to represent the feature (set in *Distance offset from*). The latter option is especially useful when the symbol size isn't fixed, e.g. if it's set by a data defined size or when using different symbols in a *categorized* renderer.
  - within a *Maximum Distance* from the feature, which is an optional setting that allows you to control how far a label can be placed from the feature it's labeling. This works alongside the *Distance* setting to create a range for label placement, adding flexibility to position labels more effectively, especially on busy maps, ensuring they fit neatly around their corresponding features.
  - using the *Prioritize Placement* option, which decides what's more important when placing labels. There are two options:
    - *Prefer closer labels*: By default, labels are kept close to the feature.
    - *Prefer position ordering*: The label will try to stay in a specific position (like top left or top right), even if it's a bit farther away from the feature. The label only moves to other positions if there's no room within the maximum distance at your preferred position.
  - following a *Position priority* which dictates placement candidates for anchoring labels around and (centered) over the point feature, and the order in which the positions are tested. The default order, based on guidelines from Krygier and Wood (2011) and other cartographic textbooks, is as follows:
    1. superior direito
    2. superior esquerdo
    3. inferior direito
    4. inferior esquerdo
    5. direita central
    6. esquerda central
    7. superior, ligeiramente à direita
    8. inferior, ligeiramente à esquerda.

    Using the Data-defined override button, you can provide a comma separated list of placements in order of priority. This also allows only certain placements to be used, for certain features only, so e.g., for coastal features you can prevent labels being placed over the land.
- *Around Point*: labels are placed in a circle around the feature with an equal radius set in *Distance*. Additionally you can set *Maximum Distance* from the feature, to control how far a label can be placed from the feature it's labeling. The placement priority is clockwise from the "top right". The position can be constrained using the data-defined *Quadrant* option.
- *Offset from Point*: labels are placed at an *Offset X,Y* distance from the point feature, in various units, or preferably over the feature. You can use a data-defined *Quadrant* to constrain the placement and can assign a *Rotation* to the label.

Posicionamento para camadas de linha

Os modos de rótulos para camadas de linha incluem:

- *Paralelo*: desenha o rótulo paralelamente a uma linha generalizada que representa a característica, com preferência para colocação em vez de porções mais retas da linha. Você pode definir:
  - *Posições permitidas*: *Acima da linha*, *Em linha*, *Abaixo da linha* e *Posição dependente da orientação da linha* (colocando a etiqueta à esquerda ou à direita do limite do polígono). É possível selecionar várias opções ao mesmo tempo. Nesse caso, o QGIS procurará a posição ideal do rótulo.
  - :guilabel:"distância" entre a etiqueta e a linha
- guilabel:*Curvado*: desenha o rótulo seguindo a curvatura da característica da linha. Além dos parâmetros disponíveis com o modo *Paralelo*, você pode definir a *ângulo máximo entre caracteres curvados*, seja dentro ou fora.
- guilabel:*Horizontal*: desenha rótulos na horizontal ao longo do traço da linha.

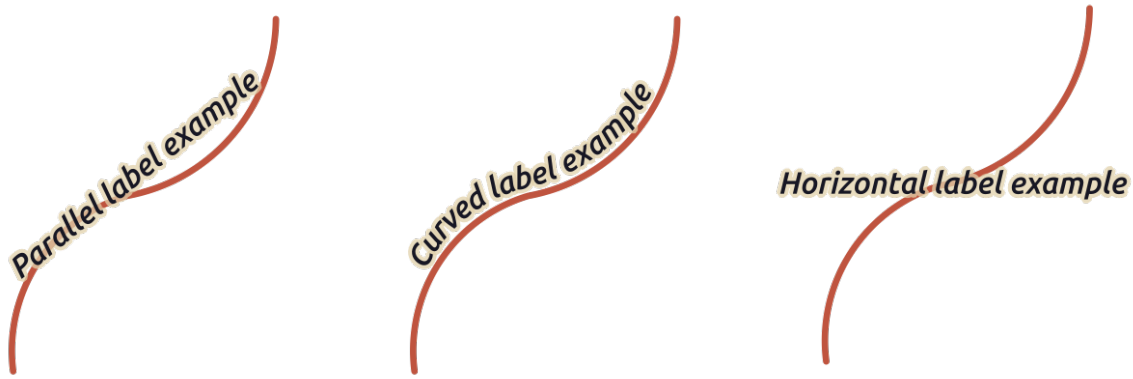


Figura10.33: Exemplos de posicionamento de rótulos para linhas

Ao lado dos modos de posicionamento, você pode definir:

- :guilabel:`Rótulos repetitivos' :guilabel:`Distância' para exibir várias vezes o rótulo ao longo do comprimento do recurso. A distância pode ser em "Milímetros", "Pontos", "Pixels", "Medidores em escala", "Unidades do mapa" e "Polegadas".
- A *Label Overrun Distance* (not available for horizontal mode): specifies the maximal allowable distance a label may run past the end (or start) of line features. Increasing this value can allow for labels to be shown for shorter line features.
- *Label Anchoring*: controls the placement of the labels along the line feature they refer to. Click on *Settings …* to choose:
  - a posição ao longo da linha (como proporção) a que rótulos serão colocados próximos. Pode ser definido pelos dados e os valores possíveis são:
    - *Centro da linha*
    - *Começo da linha*
    - *Final da linha*
    - ou *Customizar*.
  - *Clipping*: Determines how the label placement on a line is calculated. By default only the visible extent of the line is used but the whole extent can be used to have more consistent results.

  - *Anchor text*: controls which part of the text (start, center or end) will line up with the anchor point. Using *Automatic* anchoring means that:
    * For labels anchored near the start of the line (0-25%), the anchor placement will be the **start** of the label text
    * For labels anchored near the end of the line (75-100%), the anchor placement will be the **end** of the label text
    * For labels anchored near the center of the line (25-75%), the anchor placement will be the **center** of the label text
  - *Placement Behavior*: use *Preferred Placement Hint* to treat the label anchor only as a hint for the label placement. By choosing *Strict*, labels are placed exactly on the label anchor.

Posicionamento para camadas de polígono

Você pode escolher um dos seguintes modos para a colocação de rótulos de polígonos:

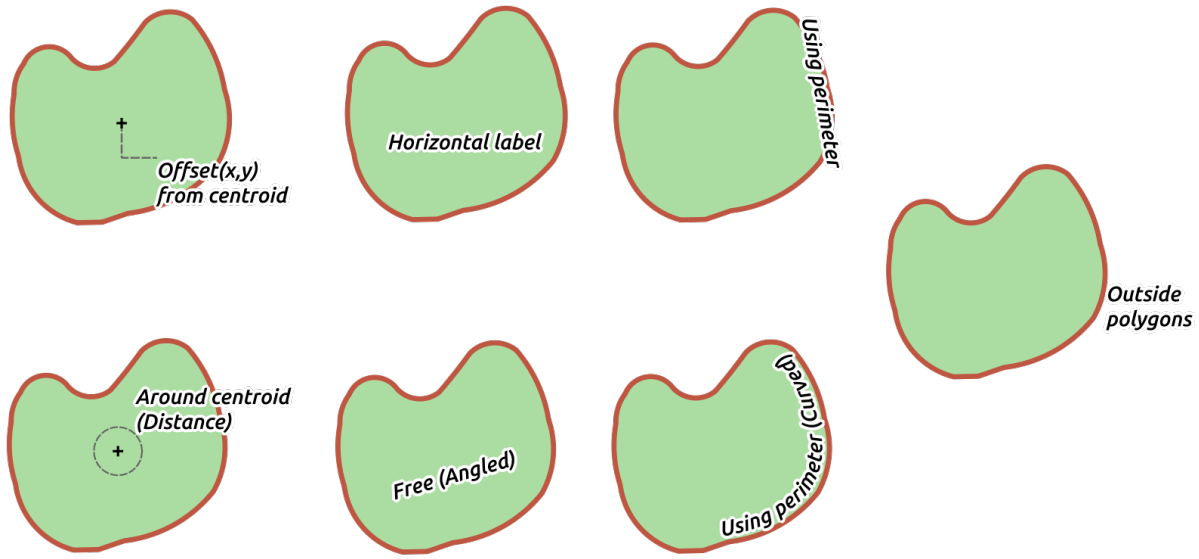


Figura10.34: Exemplos de posicionamento de rótulos para polígonos

- *Offset from Centroid*: labels are placed over the feature centroid or at a fixed *Offset X,Y* distance (in *supported units*) from the centroid. The reference centroid can be determined based on the part of the polygon rendered in the map canvas (*visible polygon*) or the *whole polygon*, no matter if you can see it. You can also:
  - forçar o ponto do centroide a ficar dentro de seu polígono
  - colocar o rótulo dentro de um quadrante específico
  - atribuir uma rotação
  - *Allow placing labels outside of polygons* when it is not possible to place them inside the polygon. Thanks to data-defined properties, this makes possible to either allow outside labels, prevent outside labels, or force outside labels on a feature-by-feature basis.
- *Around Centroid*: places the label within a preset distance around the centroid, with a preference for the placement directly over the centroid. Again, you can define whether the centroid is the one of the *visible polygon* or the *whole polygon*, and whether to force the centroid point inside the polygon.
- guilabel:*Horizontal*: coloca na um rótulo na melhor posição dentro do polígono. A colocação preferida é mais longe das bordas do polígono. É possível *Permitir a colocação de rótulos fora dos polígonos.*

- guilabel:*Livre (Angulado)*: coloca um rótulo na melhor posição girada dentro do polígono. A rotação respeita a orientação do polígono e a colocação preferida está mais longe das bordas do polígono. É possível *Permitir a colocação de rótulos fora dos polígonos*.
- *Usando o perímetro*: desenha o rótulo paralelamente a uma linha generalizada que representa o limite do polígono, com preferência por porções mais retas do perímetro. Você pode definir:
  - :guilabel:``Posições permitidas`: *Acima da linha*, *Em linha*, *Abaixo da linha* e *Posição dependente da orientação da linha* (colocando a etiqueta à esquerda ou à direita do limite do polígono). É possível selecionar várias opções ao mesmo tempo. Nesse caso, o QGIS procurará a posição ideal do rótulo.
  - *Distância* entre o rótulo e o contorno do polígono
  - o :guilabel:`Repetindo Rótulos' :guilabel:`Distância' para exibir várias vezes o rótulo ao longo do comprimento do perímetro.
- *Usando Perímetro (Curvado)*: desenha o rótulo seguindo a curvatura do limite do polígono. Além dos parâmetros disponíveis com o modo :guilabel:`Usando Perímetro', você pode definir o modo :guilabel:`ângulo máximo entre os caracteres curvos do polígono', seja dentro ou fora.
- guilabel:`Polígonos externos': sempre coloca etiquetas fora dos polígonos, em um conjunto :guilabel:`Distância'.

Configurações comuns de posicionamento

Alguns ajustes de colocação de rótulos estão disponíveis para todos os tipos de geometria de camadas:

Label Spacing

The *Label Spacing* option allows you to set the margin around the label, preventing all other labels from being placed within that area. You can dynamically control *Margin around label* using the Data-defined override button. Choose unit from the unit selector, and there you can also *adjust scaling range*.

Atenção

This option is only available on QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).

Duplicate Labels

Use *Avoid duplicate labels* to improve label appearance in situations like road networks, where features such as dual carriageways or service roads can result in many identical labels appearing close together. When active, it prevents placing labels with the exact same text (case-sensitive) closer than the minimum distance defined in the *Settings…* option. The setting applies across layers, so duplicate text from any layer will be considered.

Atenção

This option is only available on QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).

Gerador de Geometria

A seção *Gerador de geometria* permite ao usuário alterar a geometria subjacente utilizada para colocar e renderizar o rótulo, utilizando :ref:`expressões `. Isto pode ser útil para realizar o deslocamento da geometria dinamicamente ou para convertê-la em outra geometria (tipo).

A fim de utilizar o gerador geométrico:

1. Selecione a opção **|caixaDeSeleção|** *Gerador de geometria*
2. Digite a expressão gerando a geometria em que se pode confiar
3. Se relevante, selecione o tipo de geometria da saída da expressão: as configurações baseadas na geometria do rótulo, tais como posicionamento ou renderização, são atualizadas para combinar com as novas capacidades do tipo de geometria.

Alguns casos de utilização incluem:

- Use uma geometria que é salva em outro campo "label_position"
- Utilize a *geometria gerada* a partir da simbologia também para rotulagem
- Use the @map_scale variable to calculate distances / sizes be zoom level independent.
- Combinado com o modo de posicionamento curvo, cria uma etiqueta circular em torno de uma característica de ponto:

```
exterior_ring(make_circle($geometry, 20))
```

- Adicionar um rótulo no início e no final de uma linha:

```
collect_geometries( start_point($geometry), end_point($geometry) )
```

- Confie em uma linha suave de um rio para obter mais espaço para a colocação de rótulos:

```
smooth( $geometry, iterations:=30, offset:=0.25, min_length:=10 )
```

Dados Definidos

The *Data Defined* group provides direct control on labels placement, on a feature-by-feature basis. It relies on their attributes or an expression to set:

- coordenada *X* e *Y*
- o alinhamento do texto sobre a posição personalizada definida acima:
  - *Horizontal*: it can be **Left**, **Center** or **Right**
  - the text *Vertical*: it can be **Bottom**, **Base**, **Half**, **Cap** or **Top**
- the text *Rotation*. Rotation is defined as clockwise angle with 0° pointing in the direction of East for `Horizontal` oriented text and with 0° pointing in North direction for `Vertical` oriented text. Different units can be defined for the labeling rotation (e.g. `degrees`, `minutes of arc`, `turns`). QGIS prioritizes screen display rotation angles by default and always rotates text to be most readable on screen, so "upside-down" rotation angles are overridden while rendering. To change this behavior and force QGIS to render exact data-defined rotation angles there is an option in *Rendering* tab, under *Show upside-down labels*. Check the *Preserve data rotation values* entry if you want to keep the rotation value in the associated field and apply it to the label, whether the label is pinned or not. If unchecked, unpinning the label rotation is reset and its value cleared from the attribute table.

> **Nota**
>
> A rotação definida com características de polígono é atualmente suportada apenas com o modo de posicionamento *Em torno do centróide*.

> **Nota**
>
> Expressions can not be used in combination with the labels map tools (ie the *Rotate label* and *Move label* tools) to *data-define* labels placement. The widget will be reset to the corresponding *auxiliary storage field*.

Prioridade

Na seção *Prioridade* você pode definir o grau de prioridade de colocação de cada rótulo, ou seja, se houver diferentes diagramas ou possíveis rótulos para o mesmo local, o item com a prioridade mais alta será exibido e os outros poderão ser deixados de fora.

A classificação de prioridade também é usada para avaliar se um rótulo pode ser omitido devido a uma feição *obstáculo de maior peso*.

Obstáculos

Em alguns contextos (por exemplo, rótulos de alta densidade, feições sobrepostas…), a colocação dos rótulos pode resultar na colocação de rótulos sobre feições não relacionadas.

Um obstáculo é uma feição sobre a qual o QGIS evita a colocação de etiquetas ou diagramas de outras feições. Isto pode ser controlado pela :guilabel:seção: `Obstáculos'

1. Activate the *Features act as obstacles* option to decide that features of the layer should act as obstacles for any label and diagram (including items from other features in the same layer).

   Instead of the whole layer, you can select a subset of features to use as obstacles, using the Data-defined override control next to the option.

2. Utilize o botão :guilabel:`Ajustes' para ajustar o peso do obstáculo.

   - For every potential obstacle feature you can assign an *Obstacle weight*: any *label* or *diagram* whose placement priority rank is greater than this value can be placed over. Labels or diagrams with lower rank will be omitted if no other placement is possible.

     Esta ponderação também pode ser definida por dados, de modo que, dentro da mesma camada, certas características têm mais probabilidade de serem cobertas do que outras.

   - Para camadas de polígonos, você pode escolher o tipo de obstáculo que a feição é:

     – **over the feature's interior**: avoids placing labels over the interior of the polygon (prefers placing labels totally outside or just slightly inside the polygon)

     – or **over the feature's boundary**: avoids placing labels over the boundary of the polygon (prefers placing labels outside or completely inside the polygon). This can be useful for layers where the features cover the whole area (administrative units, categorical coverages, …). In this case, it is impossible to avoid placing labels within these features, and it looks much better when placing them over the boundaries between features is avoided.

Aba de renderização

Na aba :guilabel:`Renderização', você pode afinar quando os rótulos podem ser renderizados e sua interação com outros rótulos e feições.

Opções de rótulos

Abaixo de :guilabel:`Opções de rótulos':

- You find the *scale-based* and the *Pixel size-based* visibility settings.
- The *Label z-index* determines the order in which labels are rendered, as well in relation with other feature labels in the layer (using data-defined override expression), as with labels from other layers. Labels with a higher z-index are rendered on top of labels (from any layer) with lower z-index.

  Além disso, a lógica foi ajustada de modo que se doiss rótulos tiverem índices z correspondentes, então:

  - se forem da mesma camada, o rótulo menor será desenhada acima do rótulo maior
  - se forem de diferentes camadas, os rótulos serão desenhados na mesma ordem que suas próprias camadas (ou seja, respeitando a ordem definida na legenda do mapa).

  Nota

  Esta configuração não faz com que os rótulos sejam desenhados abaixo das feições de outras camadas, apenas controla a ordem na qual os rótulos são desenhados por cima de todas as feições das camadas.

- *Allow inferior fallback placements*: By default QGIS tries to render labels at their best placement, following your settings. Check this mode to allow features to fallback to worse placement options when there's no other choice, e.g. when a line is too short to fit a curved label text then the label may be placed horizontally just over the feature's center point.
- Com expressões definidas em *Mostrar rótulos* e *Sempre mostrar* você pode ajustar quais rótuloss devem ser renderizados.
- Allow to *Show upside-down labels*: alternatives are **never**, **when rotation defined** or **always**.
  - **never** - default setting, screen readability is prioritized,
  - **when rotation defined** - label rotation should be defined under *Placement* tab, within the *Data Defined* group
  - **always** - upside-down labels are allowed

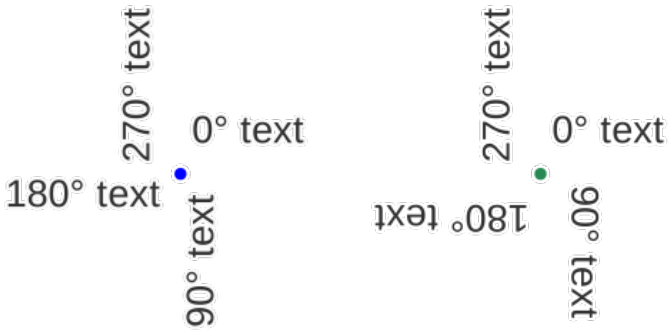


Figura10.35: Data defined label rotation with show upside-down labels option set to: "never" (left), "when rotation defined" (right)

- The *Overlapping labels* group allows you to control whether overlapping labels are permitted for features in the layer and how each of them should be handled:
  - *Never overlap*: never ever place overlapping labels for the layer, even if it means some labels will be missing
  - *Allow overlaps if required*: if the label can't otherwise be placed, draw an overlapping label. This mode will cause the label to be moved to a less ideal placement if possible, e.g. moving the label further from the center of a line or polygon, IF doing so will avoid overlapping labels. But if there's no other positions possible, then draw the label overlapping.
  - *Allow overlaps without penalty*: It doesn't matter at all if the label overlaps other labels or obstacles, that's fine to do and the best placement (e.g most central placement) should always be used even if an alternate further placement is possible which avoids overlaps entirely.

  Allowing both overlapping labels and fallback placements options will guarantee that all features in the layer are labeled… not necessarily at their best rendering!

Opções de feição

Sob *Opções de feição*:

- Você pode optar por *Rotular cada parte de uma feição multi-parte* e :guilabel:`Limitar o número de feições a serem rotutadas'.
- Ambas as camadas de linha e polígono oferecem a opção de definir um tamanho mínimo para as características a serem rotuladas, utilizando :guilabel:`Rotulagem de características menores do que'.
- Para as características poligonais, você também pode filtrar os rótulos para mostrar de acordo com se elas se encaixam completamente dentro de sua característica ou não.
- For line features, you can choose to *Merge connected lines to avoid duplicate labels*, rendering a quite airy map in conjunction with the *Distance* or *Repeat* options in the *Placement* tab.

10.4 Criando símbolos 3D

O *Gerenciador de Estilo* te ajuda a criar e armazenar símbolos 3D para todo tipo de geometria a ser renderizada no :ref:`3D map view

A partir dos demais itens, habilite o *Símbolos 3D* e expanda o menu de botões para criar:

- *3D point symbols*
- *3D line symbols*
- *3D polygon symbols*

10.4.1 Camadas de Ponto

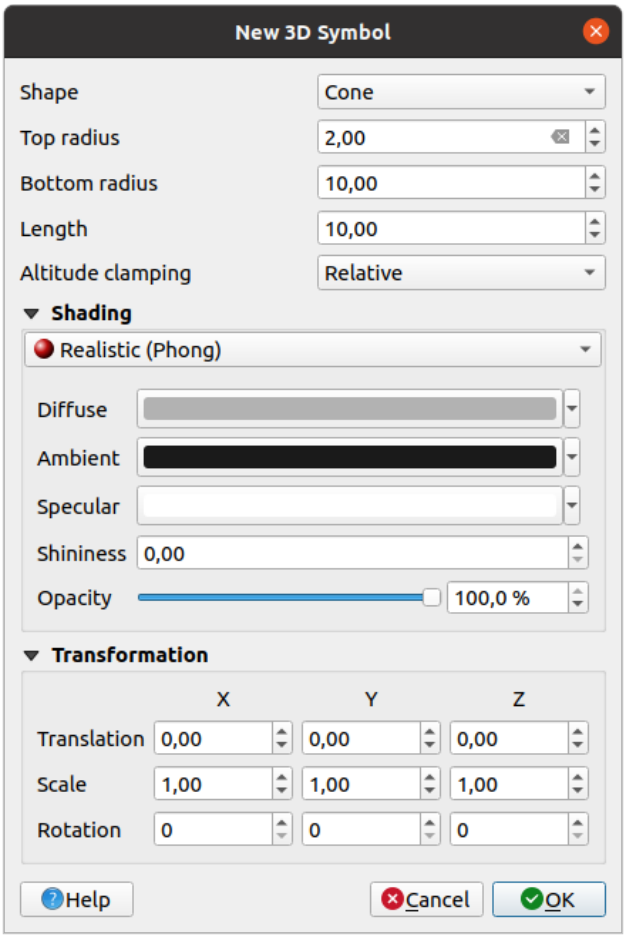


Figura10.36: Propriedades de um símbolo de ponto 3D

- Você pode definir diferentes tipos de *Forma* 3D para usar para símbolos de ponto. Eles são definidos principalmente por suas dimensões, cuja unidade se refere ao SRC do projeto. Os tipos disponíveis são:
  - *Esfera* definida por um *Raio*
  - *Cilindro* definido por um *Raio* e *Comprimento*
  - *Cubo* definido por um *Tamanho*
  - *Cone* definido por um *Raio superior*, um *Raio inferior* e um *Comprimento*
  - *Plano* definido por um *Tamanho*
  - *Toro* definido por um *Raio* e um *Raio menor*
  - *3D Model*, using a 3D model file: supported formats include wavefront `.obj`, `.glTF` and `.fbx`. Models can be a file on disk, a remote URL or *embedded in the project*. Community-created models are shared on the QGIS Hub.
  - *Quadro de avisos*, definido pela *Altura do quadro de avisos* e pelo *Símbolo do quadro de avisos* (geralmente baseado em um :ref:`marker symbol `). O símbolo terá um tamanho estável. Conveniente para visualizar Formas de nuvens de pontos 3D.
- The *Altitude clamping* can be set to *Absolute*, *Relative* or *Terrain*. The *Absolute* setting can be used when height values of the 3d vectors are provided as absolute measures from 0. *Relative* and *Terrain* add given elevation values to the underlying terrain elevation.

- The *shading* properties can be defined.
- Under the *Transformations* frame, you can apply affine transformation to the symbol:
  - *Translation* to move objects in x, y and z axis.
  - *Scale* to resize the 3D shapes
  - *Rotation* around the x-, y- and z-axis.

10.4.2 Camadas de linha

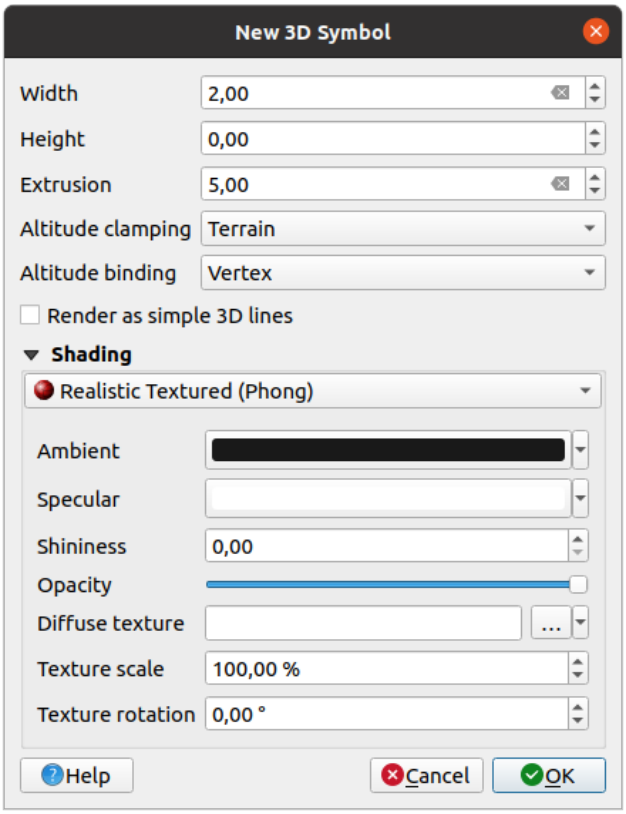


Figura10.37: Propriedades de um símbolo de linha 3D

- Beneath the *Width* and *Height* settings you can define the *Extrusion* of the vector lines. If the lines do not have z-values, you can define the 3d volumes with this setting.
- With the *Altitude clamping* you define the position of the 3D lines relative to the underlying terrain surface, if you have included raster elevation data or other 3D vectors.
- The *Altitude binding* defines how the feature is clamped to the terrain. Either every *Vertex* of the feature will be clamped to the terrain or this will be done by the *Centroid*.
- It is possible to *Render as simple 3D lines*.
- The *shading* properties can be defined.

10.4.3 Camadas de Ppolígonos

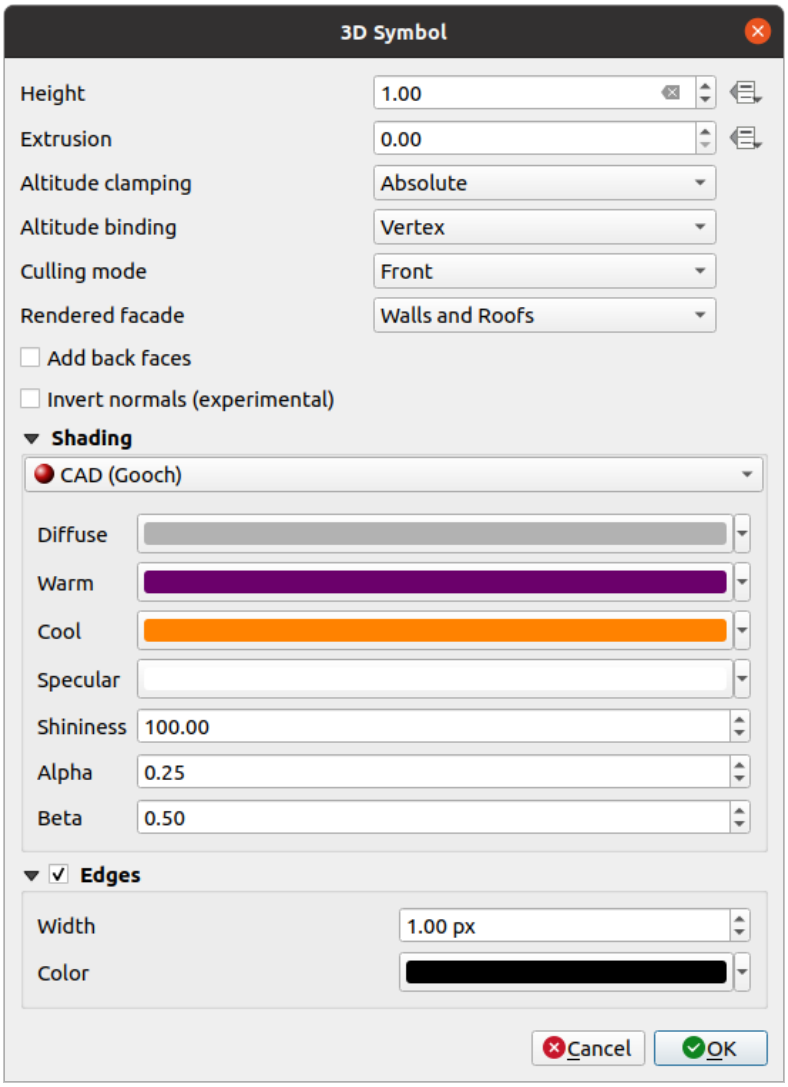


Figura10.38: Propriedades de um símbolo de polígono 3D

- As for the other ones, *Height* can be defined in CRS units. You can also use the button to overwrite the value with a custom expression, a variable or an entry of the attribute table
- Again, *Extrusion* is possible for missing z-values. Also for the extrusion you can use the button in order to use the values of the vector layer and have different results for each polygon:

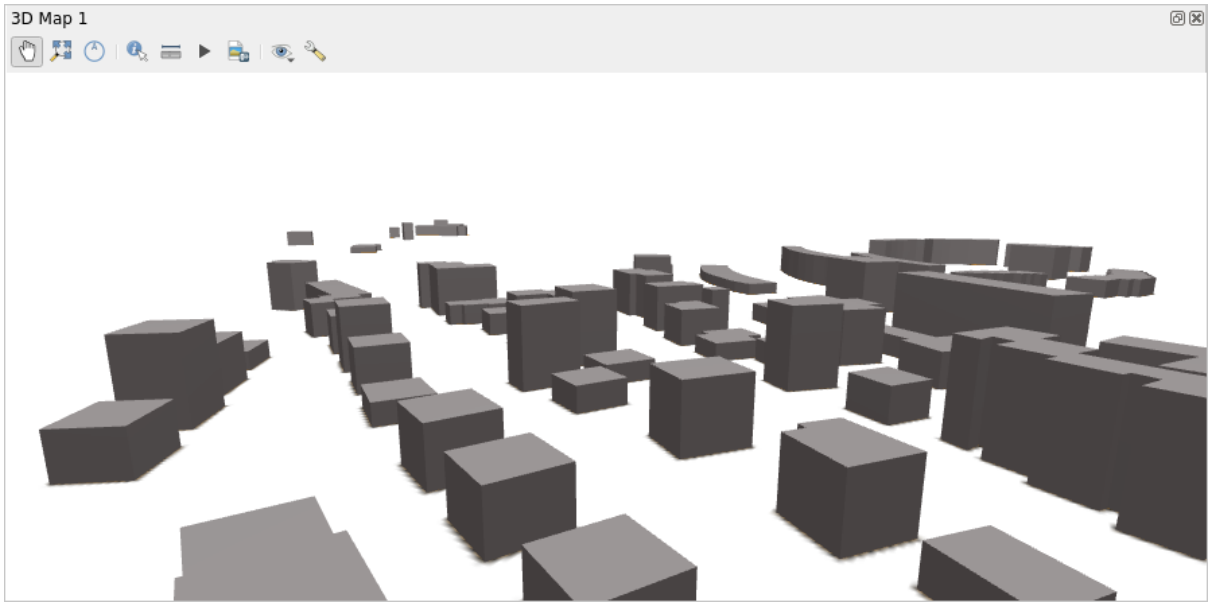


Figura10.39: Data Defined Extrusion

- The *Altitude clamping*, *Altitude binding* can be defined as explained above.
- The *Culling mode* to apply to the symbol; it can be:
  - *No Culling*: this can help to avoid seemingly missing surfaces when polygonZ/multipatch data do not have consistent ordering of vertices (e.g. all clock-wise or counter clock-wise)
  - *Front*
  - or *Back*
- The *Rendered facade* determines the faces to display. Possible values are *No facades*, *Walls*, *Roofs*, or *Walls and roofs*
- *Add back faces*: for each triangle, creates both front and back face with correct normals - at the expense of increased number of vertex data. This option can be used to fix shading issues (e.g., due to data with inconsistent order of vertices).
- *Invert normals (experimental)*: can be useful for fixing clockwise/counter-clockwise face vertex orders
- The *shading* properties can be defined.
- Display of the *Edges* of the symbols can be enabled and assigned a *Width* and *Color*.

Dica

Combination for best rendering of 3D data

Culling mode, *Add back faces* and *Invert normals* are all meant to fix the look of 3D data if it does not look right. Typically when loading some data, it is best to first try `culling mode=back` and `add back faces=disabled` - it is the most efficient. If the rendering does not look correct, try `add back faces=enabled` and keep `culling mode=no culling`. Other combinations are more advanced and useful only in some scenarios based on how mixed up is the input dataset.

10.4.4 Shading the texture

Shading helps you reveal 3d details of objects which may otherwise be hidden due to the scene's lighting. Ultimately, it's an easier material to work with as you don't need to worry about setting up appropriate scene lighting in order to visualise features.

Various techniques of shading are used in QGIS and their availability depends on the geometry type of the symbol:

- *Realistic (Phong)*: describes the way a surface reflects light as a combination of the *Diffuse* reflection of rough surfaces with the *Specular* reflection of shiny surfaces (*Shininess*). It also includes an *Ambient* option to account for the small amount of light that is scattered about the entire scene. Use the *Opacity* slider to render semi-transparent objects in 3D. Read more at Phong reflection description.
- *Realistic Textured (Phong)*: same as the *Realistic (Phong)* except that an image is used as *Diffuse Texture*. The image can be a file on disk, a remote URL or *embedded in the project*. The *Texture scale* and *Texture rotation* are required. Use the *Opacity* slider to render semi-transparent objects in 3D.
- *CAD (Gooch)*: this technique allows shading to occur only in mid-tones so that edge lines and highlights remain visually prominent. Along with the *Diffuse*, *Specular*, *Shininess* options, you need to provide a *Warm* color (for surface facing toward the light) and a *Cool* color (for the ones facing away). Also, the relative contributions to the cool and warm colors by the diffuse color are controlled by *Alpha* and *Beta* properties respectively. See also Gooch shading.
- *Metal Roughness*: a physically based rendering material that provides an accurate representation of how light interacts with surfaces. Options are available for setting the material *Base color*, *Metalness* and *Roughness*.
- *Embedded Textures* with 3D models shape

10.4.5 Application example

To go through the settings explained above you can have a look at https://app.merginmaps.com/projects/saber/luxembourg/tree.

CAPÍTULO 11

Gerenciando fonte de dados

11.1 Abrir Dados

Como parte de um ecossistema de software de código aberto, o QGIS é baseado em diferentes bibliotecas que, combinadas com seus próprios provedores, oferecem capacidades para ler e muitas vezes escrevem vários formatos:

- Vector data formats include GeoPackage, GML, GeoJSON, GPX, KML, Comma Separated Values, ESRI formats (Shapefile, Geodatabase…), MapInfo and MicroStation file formats, AutoCAD DWG/DXF, GRASS and many more… Read the complete list of supported vector formats.
- Raster data formats include GeoTIFF, JPEG, ASCII Gridded XYZ, MBTiles, R or Idrisi rasters, GDAL Virtual, SRTM, Sentinel Data, ERDAS IMAGINE, ArcInfo Binary Grid, ArcInfo ASCII Grid, and many more… Read the complete list of supported raster formats.
- Database formats include PostgreSQL, SQLite/SpatiaLite, Oracle, MS SQL Server, SAP HANA, MySQL…
- Web map and data services (WM(T)S, WFS, WCS, CSW, XYZ tiles, ArcGIS services, …) are also handled by QGIS providers. See *Working with OGC / ISO protocols* for more information about some of these.
- You can read supported files from archived folders and use QGIS native formats such as QML files (*QML - O Formato de Arquivo de Estilo QGIS*) and virtual and memory layers.

More than 80 vector and 140 raster formats are supported by GDAL and QGIS native providers.

> **Nota**
>
> Not all of the listed formats may work in QGIS for various reasons. For example, some require external proprietary libraries, or the GDAL/OGR installation of your OS may not have been built to support the format you want to use. To see the list of available formats, run the command line `ogrinfo --formats` (for vector) and `gdalinfo --formats` (for raster), or check the *Settings* ► *Options* ► *GDAL* menu in QGIS.

In QGIS, depending on the data format, there are different tools to open a dataset, mainly available in the *Layer* ► *Add Layer* ► menu or from the *Manage Layers* toolbar (enabled through *View* ► *Toolbars* menu). However, all these tools point to a unique dialog, the *Data Source Manager* dialog, that you can open with the Open Data Source Manager

button, available on the *Data Source Manager Toolbar*, or by pressing `Ctrl+L`. The *Data Source Manager* dialog (Fig.11.1) offers a unified interface to open file-based data as well as databases or web services supported by QGIS.

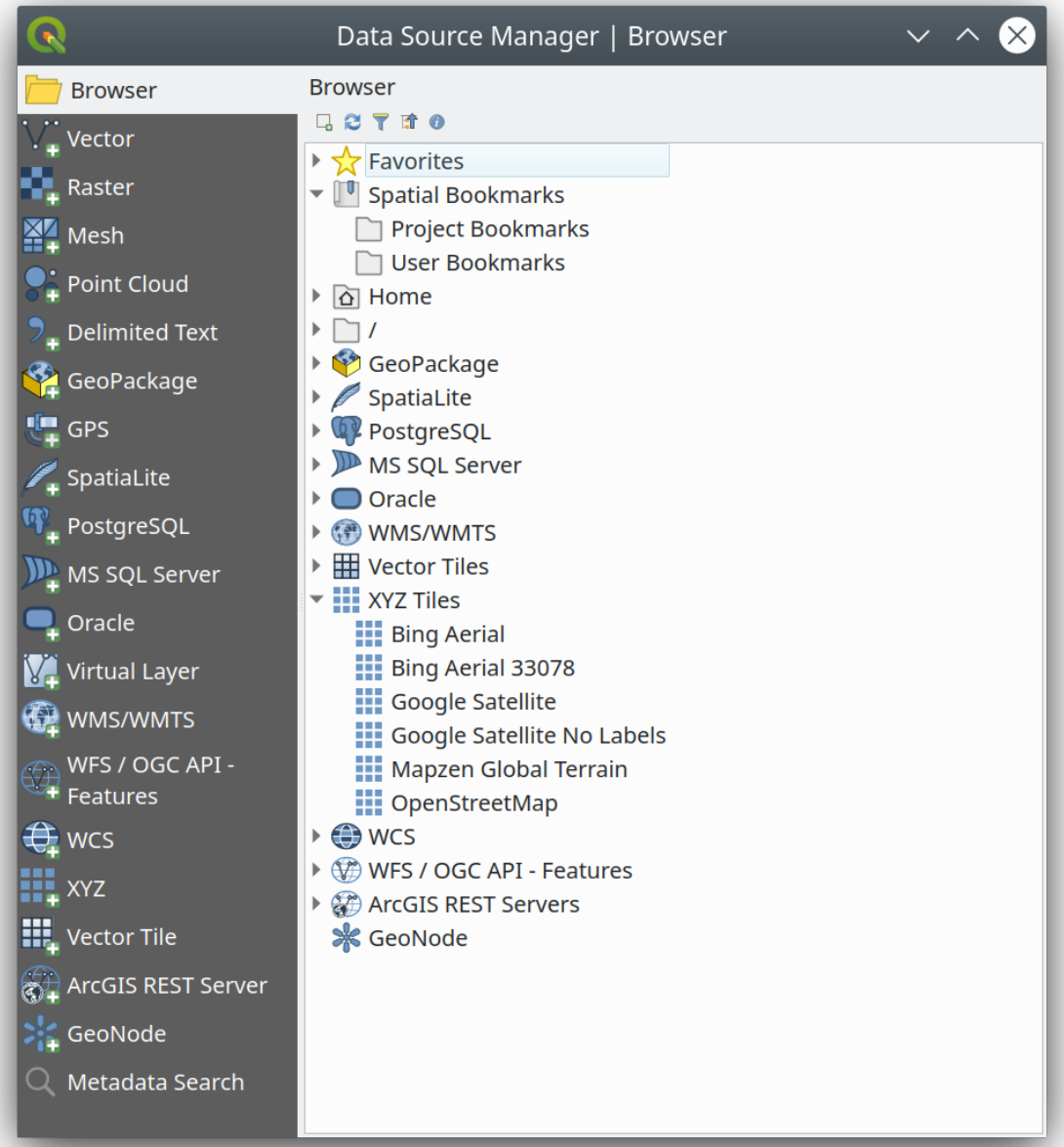


Figura11.1: QGIS Data Source Manager dialog

Beside this main entry point, you also have the *DB Manager* plugin that offers advanced capabilities to analyze and manipulate connected databases. More information on DB Manager capabilities can be found in *Plugin do gerenciador de banco de dados*.

Existem muitas outras ferramentas, complementos nativos ou de terceiros, que ajudam a abrir vários formatos de dados.

Este capítulo descreve apenas as ferramentas fornecidas por padrão no QGIS para carregar dados. Ele se concentra principalmente na caixa de diálogo *Gerenciador de Fonte de Dados*, mas, além de descrever cada guia, também explora as ferramentas baseadas no provedor de dados ou nas especificidades do formato.

11.1.1 The Browser Panel

The *Browser* is one of the main ways to quickly and easily add your data to projects. It's available as:

- a *Data Source Manager* tab, enabled pressing the Open Data Source Manager button (`Ctrl+L`);
- as a QGIS panel you can open from the menu *View* ► *Panels* (or *Settings* ► *Panels*) or by pressing `Ctrl+2`.

In both cases, the *Browser* helps you navigate in your file system and manage geodata, regardless the type of layer (raster, vector, table), or the datasource format (plain or compressed files, databases, web services).

Explorando a Interface

Na parte superior do painel Navegador, você encontra alguns botões que ajudam a:

- Add Selected Layers: you can also add data to the map canvas by selecting **Add selected layer(s)** from the layer's context menu;
- Refresh the browser tree;
- Filter Browser to search for specific data. Enter a search word or wildcard and the browser will filter the tree to only show paths to matching DB tables, filenames or folders – other data or folders won't be displayed. See the Browser Panel(2) example in Fig.11.2. The comparison can be case-sensitive or not. It can also be set to:
  - *Normal*: mostra itens que contêm o texto de pesquisa
  - *Wildcard(s)*: fine tune the search using the `?` and/or `*` characters to specify the position of the search text
  - *Expressão regular*
- Collapse All the whole tree;
- Enable/disable properties widget: when toggled on, a new widget is added at the bottom of the panel showing, if applicable, metadata for the selected item.

The entries in the *Browser* panel are organised hierarchically, and there are several top level entries:

1. *Favorites* where you can place shortcuts to often used locations
2. *Spatial Bookmarks* where you can store often used map extents (see *Bookmarking extents on the map*)
3. *Project Home*: for a quick access to the folder in which (most of) the data related to your project are stored. The default value is the directory where your project file resides.
4. *Home* directory in the file system and the filesystem root directory.
5. Connected local or network drives
6. Em seguida, vêm vários tipos de contêiner/banco de dados e protocolos de serviço, dependendo da sua plataforma e bibliotecas subjacentes:
   - *GeoPackage*
   - *SpatiaLite*
   - *PostgreSQL*
   - *SAP HANA*
   - *MS SQL Server*
   - *Oracle*

- *WMS/WMTS*
- *Vector Tiles*
- *XYZ Tiles*
- *WCS*
- *WFS/OGC API-Features*
- *ArcGIS REST Server*

Interacting with the Browser items

The browser supports drag and drop within the browser, from the browser to the canvas and *Layers* panel, and from the *Layers* panel to layer containers (e.g. GeoPackage) in the browser.

Project file items inside the browser can be expanded, showing the full layer tree (including groups) contained within that project. Project items are treated the same way as any other item in the browser, so they can be dragged and dropped within the browser (for example to copy a layer item to a geopackage file) or added to the current project through drag and drop or double click.

The context menu for an element in the *Browser* panel is opened by right-clicking on it.

Para entradas de diretório do sistema de arquivos, o menu de contexto oferece o seguinte:

- *New* ► to create in the selected entry a:
  - *Diretório…*
  - *GeoPackage…*
  - *ShapeFile…*
- *Add as a Favorite*: favorite folders can be renamed (*Rename favorite…*) or removed (*Remove favorite*) any time.
- *Hide from Browser*: hidden folders can be toggled to visible from the *Settings* ► *Options* ► *Data Sources* ► *Hidden browser paths* setting
- *Fast Scan this Directory*
- *Abrir Diretório*
- *Abrir no Terminal*
- *Propriedades…*
- *Propriedades do diretório…*

For leaf entries that can act as layers in the project, the context menu will have supporting entries. For example, for non-database, non-service-based vector, raster and mesh data sources:

- **:menuSeleção:`Exportar camada --> Para arquivo...`**
- *Adicionar camada ao projeto*
- *Propriedades da Camada*
- *Open with Data Source Manager…*
- *Gerenciar* ► *Renomear "<name of file>"…* ou *Deletar "<name of file>"…*
- *Mostrar em arquivos*
- *Propriedades do arquivo*

In the *Layer properties* entry, you will find (similar to what you will find in the *vector* and *raster* layer properties once the layers have been added to the project):

- *Metadados* for the layer. para a camada. Grupos de Metadados: *Informações do provedor* (se possível,será um hiperlink para a fonte *Path*), *Identificação*, *Extensão*, *Acesso*, *Campos* (para camadas vetoriais), *Bandas* (para camadas raster), *Contatos*, *Links* (para camadas vetoriais), *Referências* (para camadas raster), *Histórico*.
- A *Preview* panel
- The attribute table for vector sources (in the *Attributes* panel).

Use *Open with Data Source Manager…* to directly open and configure the data source in the *Data Source Manager* using the URI of your data source. This simplifies the process of adding a layer from the *Browser* by allowing you to set specific opening options for the data source. It is currently available for vector (including the dedicated GeoPackage entry), raster, and SpatiaLite data sources.

Para adicionar uma camada ao projeto usando a *Navegador*:

1. Enable the *Browser* as described above. A browser tree with your file system, databases and web services is displayed. You may need to connect databases and web services before they appear (see dedicated sections).
2. Encontre a camada na lista.
3. Use the context menu, double-click its name, or drag-and-drop it into the *map canvas*. Your layer is now added to the *Layers panel* and can be viewed on the map canvas.

> **Dica**
>
> **Abra um projeto QGIS diretamente do navegador**
>
> You can also open a QGIS project directly from the Browser panel by double-clicking its name or by drag-and-drop into the map canvas.

Depois que um arquivo é carregado, você pode ampliar usando as ferramentas de navegação do mapa. Para alterar o estilo de uma camada, abra a caixa de diálogo *Propriedades da Camada* clicando duas vezes no nome da camada ou clicando com o botão direito do mouse no nome da legenda e escolhendo :menuselection:` Propriedades` no menu de contexto. Veja a seção *Propriedades da simbologia* para mais informações sobre a configuração de simbologia para camadas vetoriais.

Right-clicking an item in the browser tree helps you to:

- for a file or a table, display its metadata or open it in your project. Tables can even be renamed, deleted or truncated.
- para uma pasta, adicione-a aos favoritos ou oculte-a na árvore do navegador. As pastas ocultas podem ser gerenciadas a partir da guia *Configurações* ► *Opções* ► *Fontes de Dados*.
- manage your *spatial bookmarks*: bookmarks can be created, exported and imported as `XML` files.
- crie uma conexão com um banco de dados ou um serviço da web.
- atualizar, renomear ou excluir um esquema.

Você também pode importar arquivos para bancos de dados ou copiar tabelas de um esquema/banco de dados para outro com um simples arrastar e soltar. Há um segundo painel do navegador disponível para evitar rolagem longa enquanto arrasta. Basta selecionar o arquivo e arrastar e soltar de um painel para o outro.

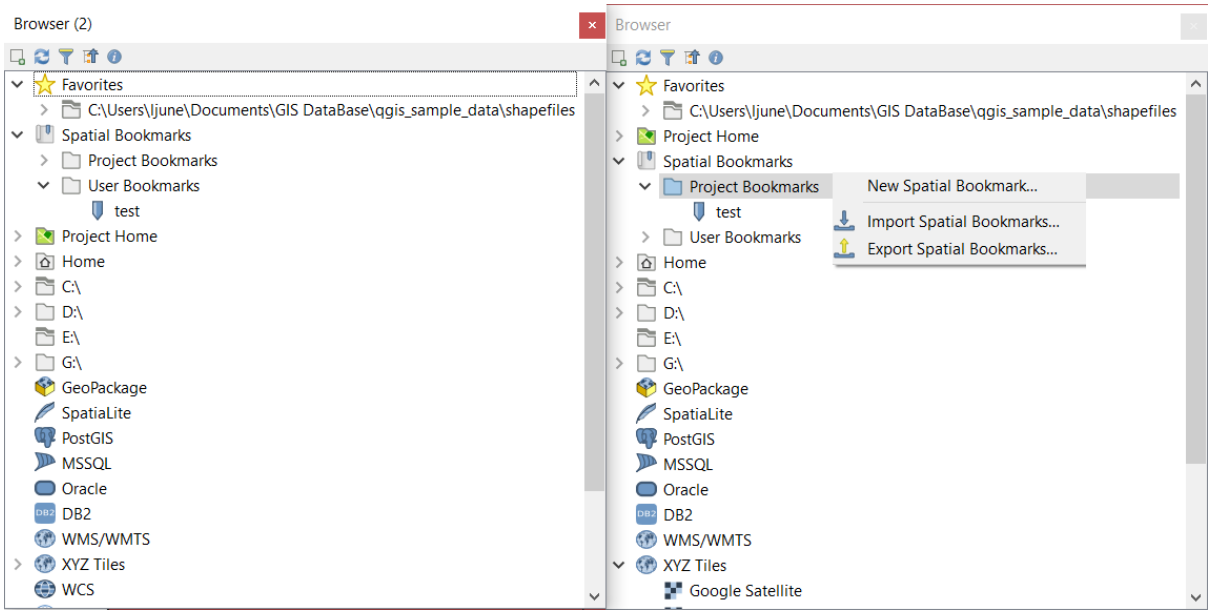


Figura11.2: QGIS Browser panels side-by-side

Dica

Add layers to QGIS by simple drag-and-drop from your OS file browser

You can also add file(s) to the project by drag-and-dropping them from your operating system file browser to the *Layers Panel* or the map canvas.

11.1.2 The DB Manager

The *DB Manager* Plugin is another tool for integrating and managing spatial database formats supported by QGIS (PostgreSQL, SpatiaLite, GeoPackage, Oracle Spatial, MS SQL Server, Virtual layers). It can be activated from the *Plugins* ► *Manage and Install Plugins...* menu.

The DB Manager Plugin provides several features:

- connect to databases and display their structure and contents
- preview tables of databases
- add layers to the map canvas, either by double-clicking or drag-and-drop.
- add layers to a database from the QGIS Browser or from another database
- create SQL queries and add their output to the map canvas
- criar *virtual layers*

More information on DB Manager capabilities is found in *Plugin do gerenciador de banco de dados*.

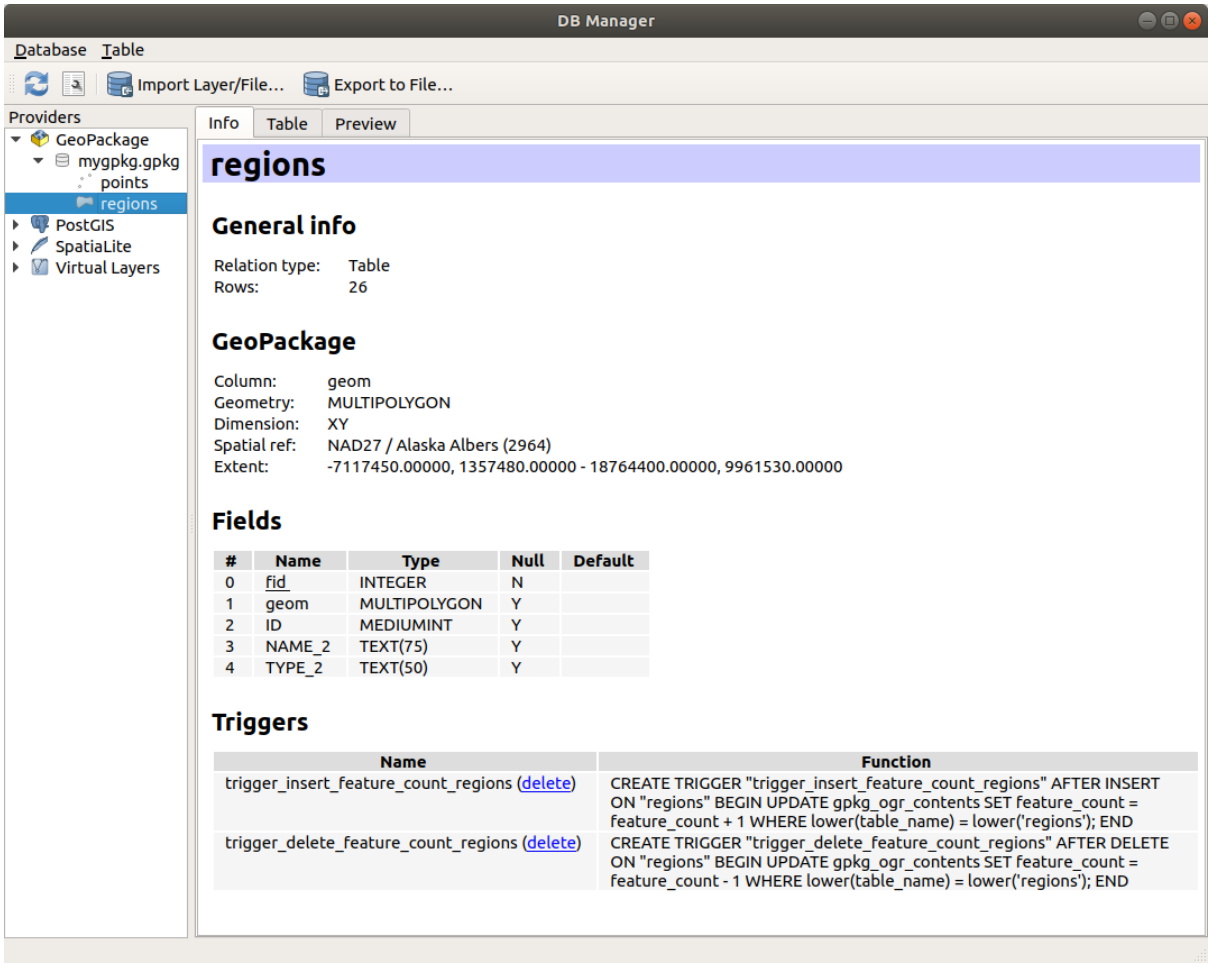


Figura11.3: Caixa de diálogo Gerenciador DB

11.1.3 Provider-based loading tools

Beside the Browser Panel and the DB Manager, the main tools provided by QGIS to add layers, you'll also find tools that are specific to data providers.

> **Nota**
>
> Alguns: 'plugins externos <plugins>' também fornecem ferramentas para abrir arquivos de formato específico no QGIS.

Carregando uma camada de um arquivo

Para carregar uma camada de um arquivo:

1. Abra a guia do tipo de camada na caixa de diálogo *Gerenciador de Fonte de Dados*, ou seja, clique no botão Abrir Gerenciador de Fonte de Dados (ou pressione: kbd:*Ctrl+L*) e ative a guia de destino ou:
   - for vector data (like GML, ESRI Shapefile, Mapinfo and DXF layers): press `Ctrl+Shift+V`, select the *Layer* ► *Add Layer* ► *Add Vector Layer* menu option or click on the Add Vector Layer toolbar button.

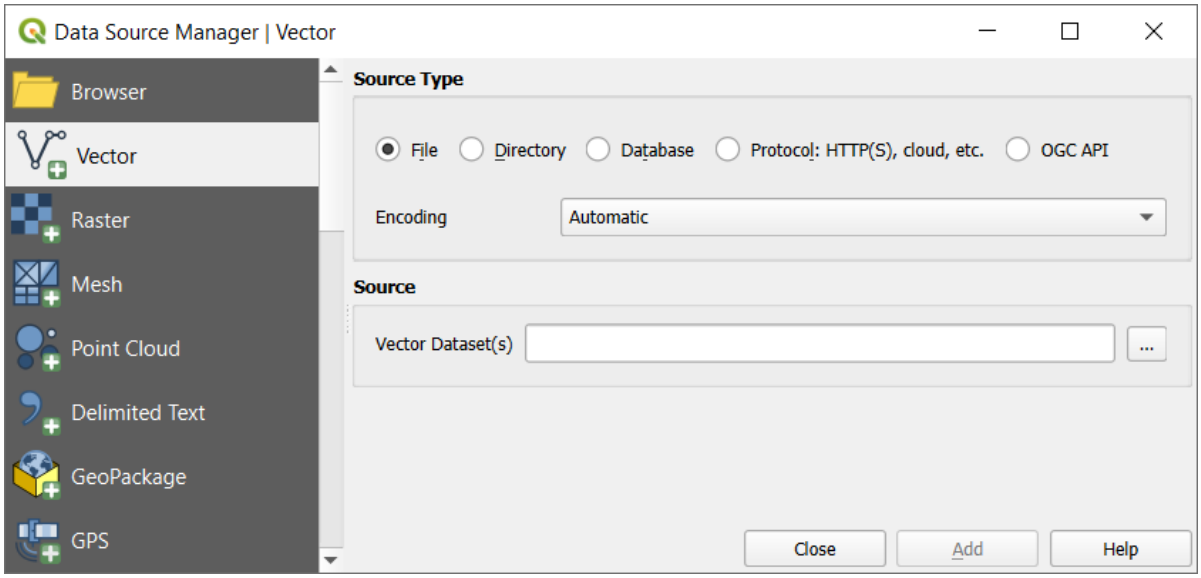


Figura11.4: Add Vector Layer Dialog

- para dados raster (como camadas GeoTiff, MBTiles, GRIdded Binary e DWG): pressione `Ctrl+Shift+R`, selecione a opção *Camada* ► *Adicionar Camada* ► *Adicionar Camada Raster* do menu ou clique no botão Adicionar Camada Raster da barra de ferramentas.

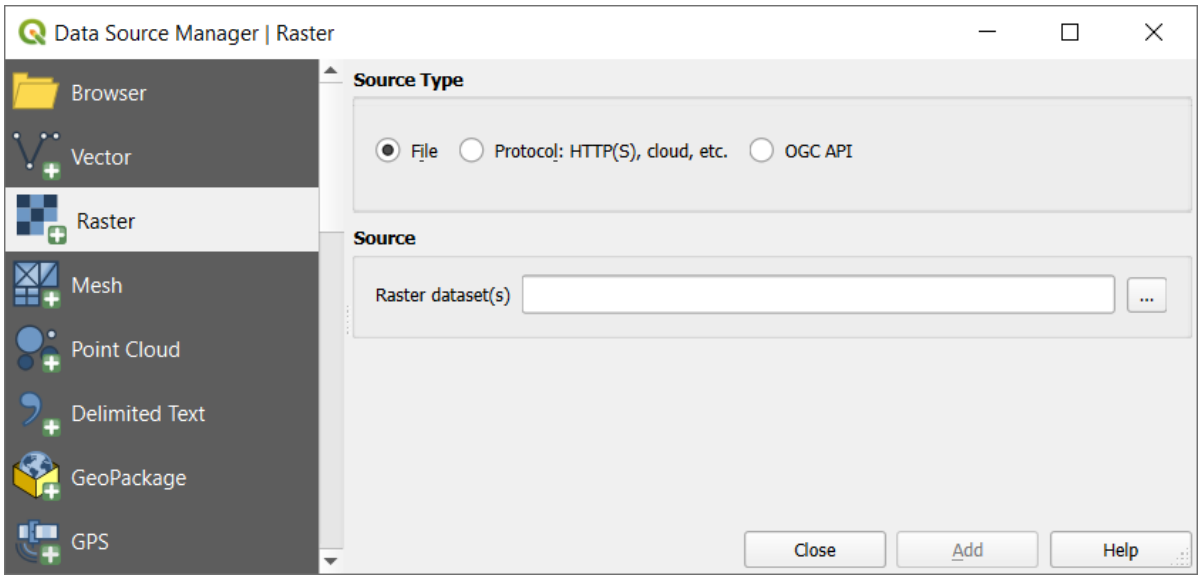


Figura11.5: Add Raster Layer Dialog

2. Check *File* source type
3. Click on the ... Browse button
4. Navegue no sistema de arquivos e carregue uma fonte de dados suportada. É possível carregar mais de uma camada ao mesmo tempo mantendo pressionada a tecla `Ctrl` e clicando em vários itens da caixa de diálogo ou pressionando a tecla `Shift` para selecionar um intervalo de itens clicando em o primeiro e o último item do intervalo. Somente os formatos que foram bem testados aparecem no filtro de formatos. Outros formatos podem ser carregados selecionando `Todos os arquivos` (o item superior no menu pull-down).
5. Press *Open* to load the selected file into *Data Source Manager* dialog.

Depending on the selected layer type, additional *Options* (encoding, geometry type, table filtering, file locking, data formatting …) are available for configuring. These options are described in detail in the specific GDAL vector or raster driver documentation. At the top of the options, a text with hyperlink will directly lead to the documentation of the appopriate driver for the selected file format.

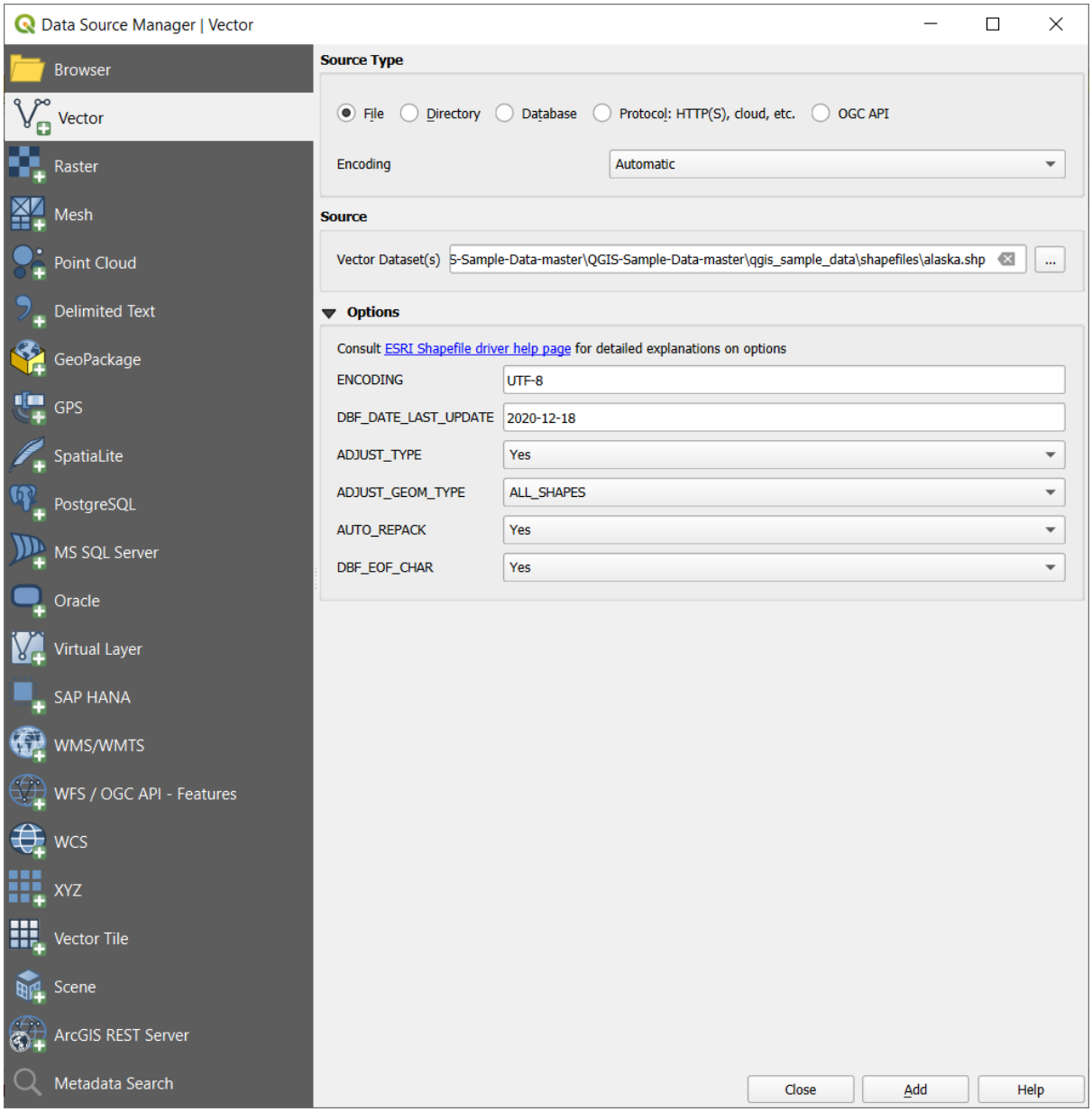


Figura11.6: Carregando um Shapefile com opções em aberto

6. Press *Add* to load the file in QGIS and display them in the map view. When adding vector datasets containing multiple layers, the *Select Items to Add* dialog will appear. In this dialog, you can choose the specific layers from your dataset that you want to add. Also, under *Options* you can choose to:

   - *Add layers to a group*
   - *Show system and internal tables*
   - *Show empty vector layers.*

   Fig.11.7 shows QGIS after loading the `alaska.shp` file.

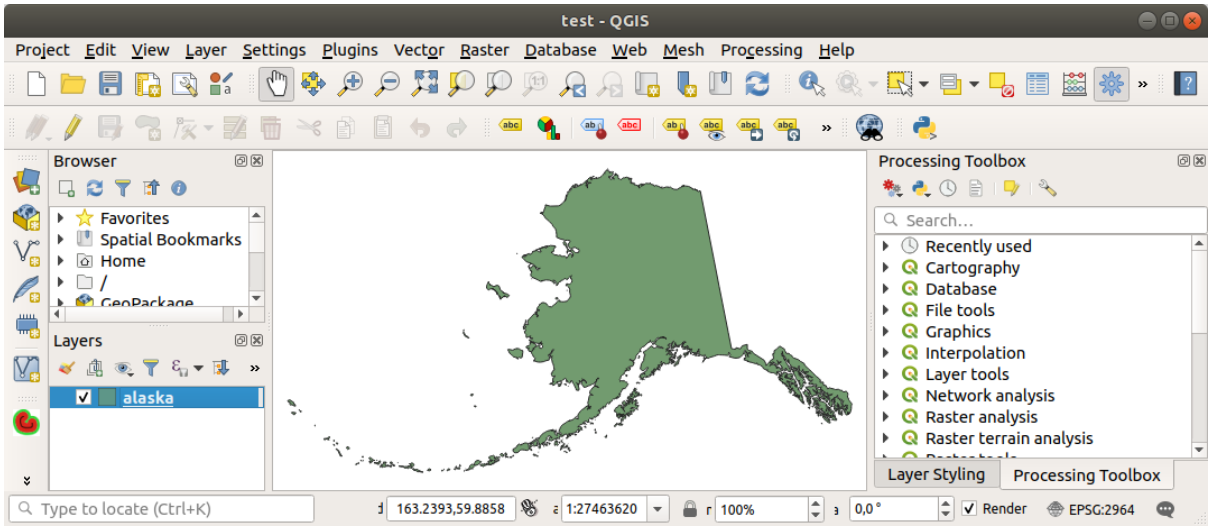


Figura11.7: QGIS com Shapefile do Alasca carregado

Nota

Como alguns formatos como MapInfo (por exemplo, `.tab`) ou Autocad (`.dxf`) permitem misturar diferentes tipos de geometria em um único arquivo, carregar tais conjuntos de dados abre um diálogo para selecionar as geometrias a serem utilizadas a fim de ter uma geometria por camada.

The Add Vector Layer and Add Raster Layer tabs allow loading of layers from source types other than *File*:

- You can load specific vector formats like `ArcInfo Binary Coverage`, `UK. National Transfer Format`, as well as the raw TIGER format of the `US Census Bureau` or `OpenfileGDB`. To do that, you select *Directory* as *Source type*. In this case, a directory can be selected in the dialog after pressing ... Browse.

- With the *Database* source type you can select an existing database connection or create one to the selected database type. Some possible database types are `ODBC`, `Esri Personal Geodatabase`, `MS SQL Server` as well as `PostgreSQL` or `MySQL` .

  Pressing the *New* button opens the *Create a New OGR Database Connection* dialog whose parameters are among the ones you can find in *Criando uma conexão armazenada*. Pressing *Open* lets you select from the available tables.

- O tipo de fonte *Protocolo: HTTP(S), nuvem, etc.* abre dados armazenados localmente ou na rede sejam publicamente acessíveis, ou em armazenamentos em nuvem particulares fornecidos por serviços comerciais. Os tipos de protocolo suportados são:

  - `HTTP/HTTPS/FTP`, with a *URI* and, if required, an *authentication*.
  - Cloud storage such as `AWS S3`, `Google Cloud Storage`, `Microsoft Azure Blob`, `Microsoft Azure Data Lake Storage`, `Alibaba OSS Cloud`, and `Open Stack Swift Storage` supports direct control over VSI *Credential Options* when adding OGR vector or GDAL raster layers. You need to fill in the *Bucket or container* and the *Object key* first. After that, you can add the necessary *Credential Options*.

    When adding OGR vector or GDAL raster layers from the cloud based protocols, you can also set additional *Credential options* for that specific driver and bucket. When credential options are found in a layer's URI, they will also be automatically set. This allows different layers to use different credentials.
  - service supporting OGC `WFS 3` (still experimental), using `GeoJSON` or `GEOJSON - Newline Delimited` format or based on `CouchDB` database. A *URI* is required, with optional *authentication*.

  - For all vector source types it is possible to define the *Encoding* or to use the *Automatic* ► setting.

- The *OGC API* source type allows you to access vector and raster data from servers that implement the OGC API standards. To use this option:

  1. Select *OGC API* from the *Data Source Manager* dialog.
  2. Enter the endpoint of the OGC API service you want to connect to. Note that you don't need to prefix the endpoint with "OGCAPI:".
  3. Click *Connect* to establish a connection to the server.

Carregando uma camada de malha

A mesh is an unstructured grid usually with temporal and other components. The spatial component contains a collection of vertices, edges and faces in 2D or 3D space. More information on mesh layers at *Working with Mesh Data*.

Para adicionar uma camada de malha ao QGIS:

1. Open the *Data Source Manager* dialog, either by selecting it from the *Layer* ► menu or clicking the Open Data Source Manager button.
2. Ative a guia *Malha* no painel da esquerda
3. Pressione o botão … Buscar para selecionar o arquivo. *Vários formatos* são suportados.
4. Select the file and press *Add*. The layer will be added using the native mesh rendering.
5. If the selected file contains many mesh layers, then you'll be prompted with a dialog to choose the sublayers to load. Do your selection and press *OK* and the layers are loaded with the native mesh rendering. It's also possible to load them within a group.

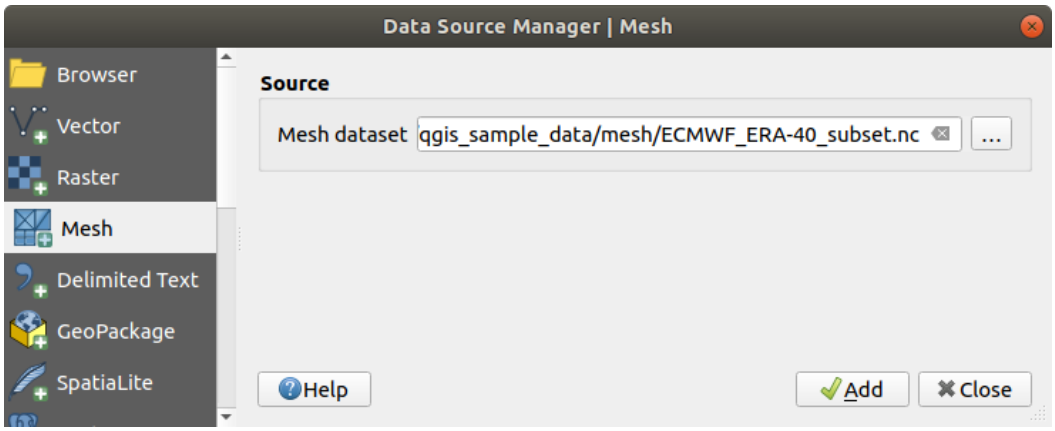


Figura11.8: Mesh tab in Data Source Manager

Importando um arquivo de texto delimitado

Delimited text files (e.g. `.txt`, `.csv`, `.dat`, `.wkt`) can be loaded using the tools described above. This way, they will show up as simple tables. Sometimes, delimited text files can contain coordinates / geometries that you could want to visualize. This is what *Add Delimited Text Layer* is designed for.

1. Click the Open Data Source Manager icon to open the *Data Source Manager* dialog
2. Ative a guia *Texto Delimitado*
3. Select the delimited text file to import (e.g., `qgis_sample_data/csv/elevp.csv`) by clicking on the … Browse button.
4. Configure as configurações para atender seu conjunto de dados e necessidades, conforme explicado abaixo.

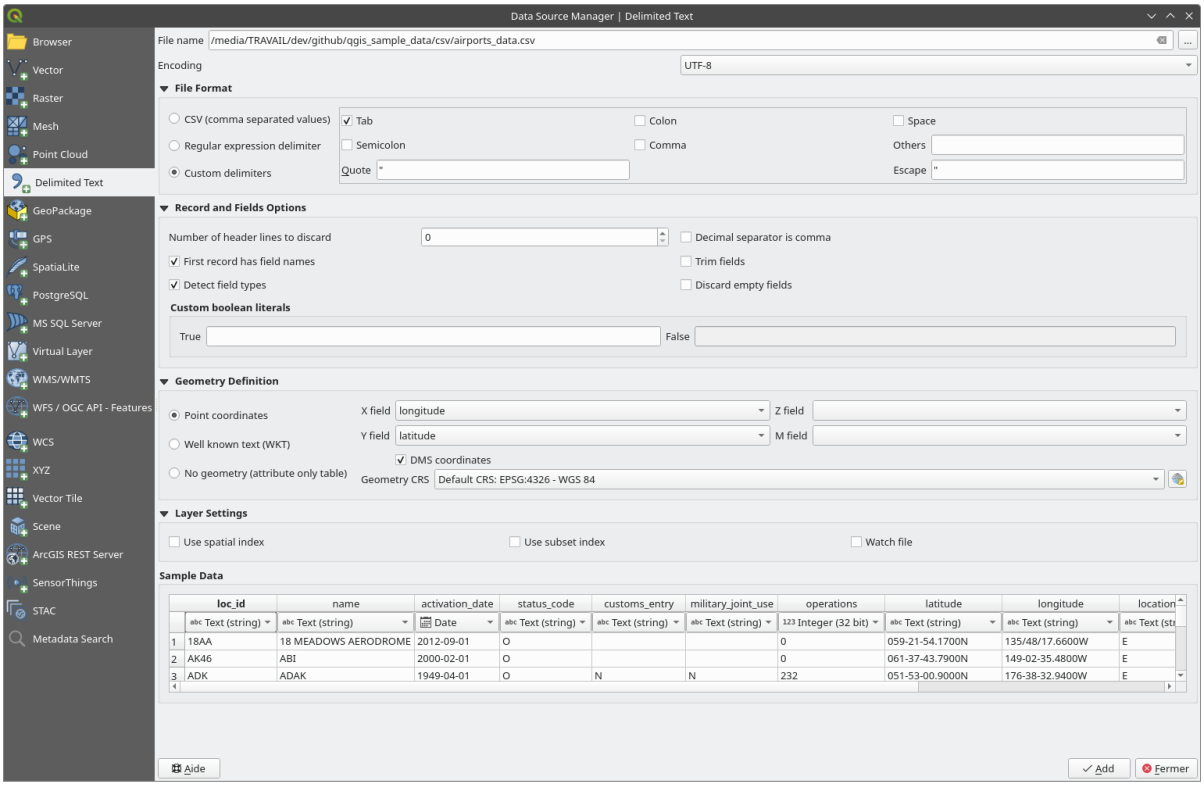


Figura11.9: Delimited Text Dialog

Formato de arquivo

Depois que o arquivo é selecionado, o QGIS tenta analisar o arquivo com o delimitador usado mais recentemente, identificando campos e linhas. Para permitir que o QGIS analise corretamente o arquivo, é importante selecionar o delimitador correto. Você pode especificar um delimitador escolhendo entre:

- *CSV (valores separados por vírgula)* para utilizar o caractere de vírgula.
- *Regular expression delimiter* and enter text into the *Expression* field. For example, to change the delimiter to tab, use `\t` (this is used in regular expressions for the tab character).
- *Custom delimiters*, choosing among some predefined delimiters like `comma`, `space`, `tab`, `semicolon`, … .

Registros e campos

Algumas outras opções convenientes podem ser usadas para reconhecimento de dados:

- *Number of header lines to discard*: convenient when you want to avoid the first lines in the file in the import, either because those are blank lines or with another formatting.
- *First record has field names*: values in the first line are used as field names, otherwise QGIS uses the field names `field_1`, `field_2`...
- *Detect field types*: automatically recognizes the field type. If unchecked then all attributes are treated as text fields.
- *Separador decimal é vírgula*: você pode forçar o separador decimal a ser uma vírgula.
- *Trim fields*: allows you to trim leading and trailing spaces from fields.
- *Discard empty fields*.
- *Custom boolean literals*: allows you to add a custom couple of string that will be detected as boolean values.

Field type detection

QGIS tries to detect the field types automatically (unless *Detect field types* is not checked) by examining the content of an optional sidecar CSVT file (see GeoCSV specification) and by scanning the whole file to make sure that all values can actually be converted without errors, the fall-back field type is text.

The detected field type appears under the field name in sample data preview table and can be manually changed if necessary.

The following field types are supported:

- `Boolean` case-insensitive literal couples that are interpreted as boolean values are `1`/`0`, `true`/`false`, `t`/`f`, `yes`/`no`
- `Whole Number (integer)`
- `Whole Number (integer - 64 bit)`
- `Decimal Number`: double precision floating point number
- `Date`
- `Time`
- `Date and Time`
- `Text`

Definição de geometria

Once the file is parsed, set *Geometry definition* to

- *Point coordinates* and provide the *X field*, *Y field*, *Z field* (for 3-dimensional data) and *M field* (for the measurement dimension) if the layer is of point geometry type and contains such fields. If the coordinates are defined as degrees/minutes/seconds, activate the *DMS coordinates* checkbox. Provide the appropriate *Geometry CRS* using the Select CRS widget.
- *Well known text (WKT)* option if the spatial information is represented as WKT: select the *Geometry field* containing the WKT geometry and choose the approriate *Geometry field* or let QGIS auto-detect it. Provide the appropriate *Geometry CRS* using the Select CRS widget.

- If the file contains non-spatial data, activate *No geometry (attribute only table)* and it will be loaded as an ordinary table.

Configurações de camada

Além disso, você pode habilitar:

- *Use spatial index* to improve the performance of displaying and spatially selecting features.
- *Use subset index* to improve performance of *subset filters* (when defined in the layer properties).
- *Watch file* to watch for changes to the file by other applications while QGIS is running.

At the end, click *Add* to add the layer to the map. In our example, a point layer named `Elevation` is added to the project and behaves like any other map layer in QGIS. This layer is the result of a query on the `.csv` source file (hence, linked to it) and would require *to be saved* in order to get a spatial layer on disk.

Sample Data

As you set the parser properties, the sample data preview updates regarding to the applied settings.

Also in the Sample Data Table it is possible to override the automatically determined column types.

Importando um arquivo DXF ou DWG

`DXF` and `DWG` files can be added to QGIS by simple drag-and-drop from the Browser Panel. You will be prompted to select the sublayers you would like to add to the project. Layers are added with random style properties.

> **Nota**
>
> For DXF files containing several geometry types (point, line and/or polygon), the name of the layers will be generated as *<filename.dxf> entities <geometry type>*.

To keep the dxf/dwg file structure and its symbology in QGIS, you may want to use the dedicated *Project ► Import/Export ► Import Layers from DWG/DXF…* tool which allows you to:

1. import elements from the drawing file into a GeoPackage database.
2. adicionar elementos importados ao projeto.

In the *DWG/DXF Import* dialog, to import the drawing file contents:

1. Input the location of the *Source drawing*, i.e. the DWG/DXF drawing file to import.
2. Especifique o sistema de referência de coordenadas dos dados no arquivo de desenho.
3. Input the location of the *Target package*, i.e. the GeoPackage file that will store the data. If an existing file is provided, then it will be overwritten.
4. Choose how to import `blocks` with the dedicated combobox:
   - *Expand Block Geometries*: imports the blocks in the drawing file as normal elements.
   - *Expand Block Geometries and Add Insert Points*: imports the blocks in the drawing file as normal elements and adds the insertion point as a point layer.
   - *Add Only Insert Points*: adds the blocks insertion point as a point layer.
5. Check *Use curves* to promote the imported layers to a `curved` geometry type.

6. Use the *Import* button to import the drawing into the destination GeoPackage file. The GeoPackage database will be automatically populated with the drawing file content. Depending on the size of the file, this can take some time.

After the `.dwg` or `.dxf` data has been imported into the GeoPackage database, the frame in the lower half of the dialog is populated with the list of layers from the imported file. There you can select which layers to add to the QGIS project:

1. At the top, set a *Group name* to group the drawing files in the project. By default this is set to the filename of the source drawing file.
2. Check layers to show: Each selected layer is added to an ad hoc group which contains vector layers for the point, line, label and area features of the drawing layer. The style of the layers will resemble the look they originally had in *CAD.
3. Escolha se a camada deve estar visível na abertura.
4. Marcar a opção *Mesclar camadas* coloca todas as camadas em um único grupo.
5. Press *OK* to open the layers in QGIS.

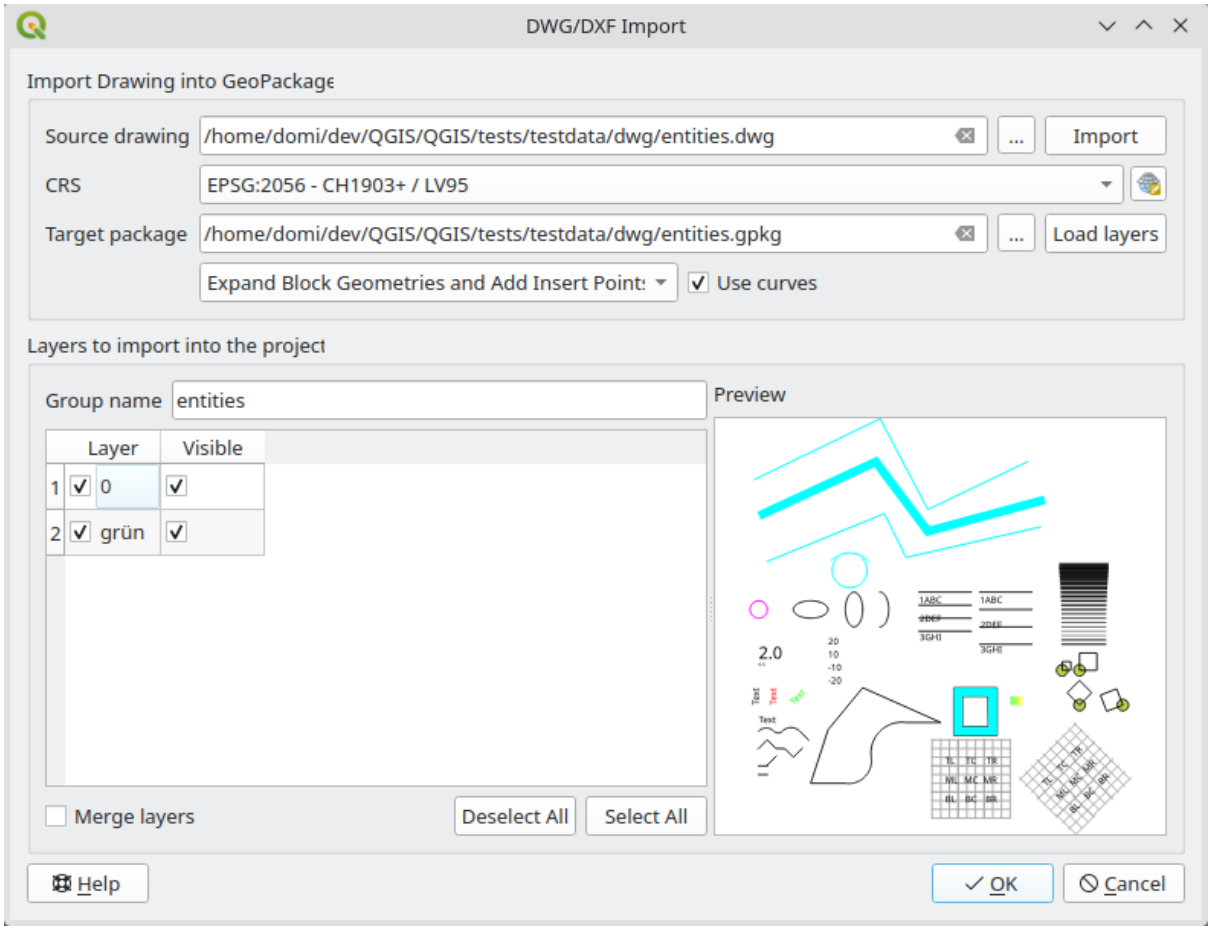


Figura11.10: Import dialog for DWG/DXF files

Importando vetores OpenStreetMap

The OpenStreetMap project is popular because in many countries no free geodata such as digital road maps are available. The objective of the OSM project is to create a free editable map of the world from GPS data, aerial photography and local knowledge. To support this objective, QGIS provides support for OSM data.

Using the *Browser Panel*, you can load an `.osm` file to the map canvas, in which case you'll get a dialog to select sublayers based on the geometry type. The loaded layers will contain all the data of that geometry type in the `.osm` file, and keep the `osm` file data structure.

Camadas SpatiaLite

The first time you load data from a SpatiaLite database, begin by:

- clicking on the Add SpatiaLite Layer toolbar button
- selecting the *Add SpatiaLite Layer…* option from the *Layer* ► *Add Layer* menu
- ou digitando `Ctrl+Shift+L`

This will bring up a window that will allow you either to connect to a SpatiaLite database already known to QGIS (which you choose from the drop-down menu) or to define a new connection to a new database. To define a new connection, click on *New* and use the file browser to point to your SpatiaLite database, which is a file with a `.sqlite` extension.

O QGIS também suporta visualizações editáveis no SpatiaLite.

GPS

Existem dezenas de formatos de arquivo diferentes para armazenar dados GPS. O formato que o QGIS usa é chamado GPX (formato GPS eXchange), que é um formato padrão de intercâmbio que pode conter qualquer número de pontos de passagem, rotas e trilhas no mesmo arquivo.

Use the … Browse button to select the GPX file, then use the check boxes to select the feature types you want to load from that GPX file. Each feature type will be loaded in a separate layer.

More on GPS data manipulation at *Trabalhando com dados GPS*.

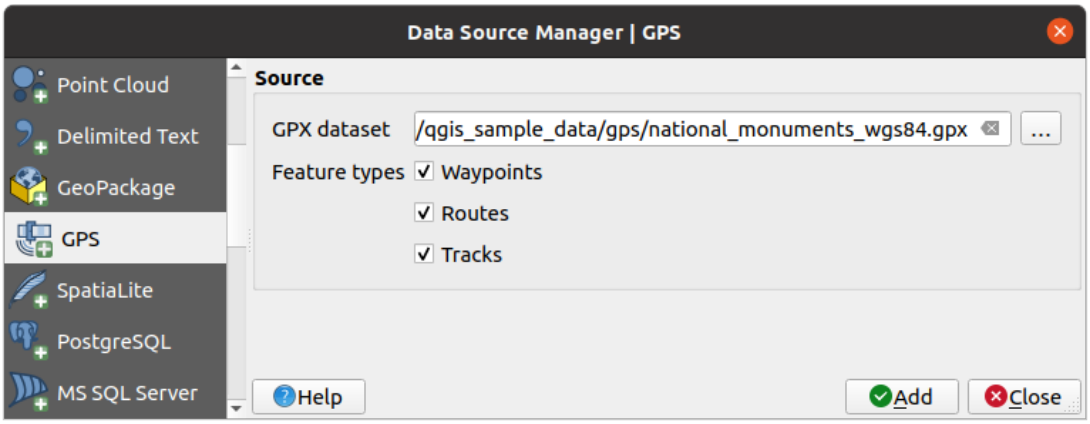


Figura11.11: Carregando a janela de dados do GPS

GRASS

Working with GRASS vector data is described in section *Integração com SIG GRASS*.

Database related tools

Criando uma conexão armazenada

Para ler e gravar tabelas a partir de um formato de banco de dados, o QGIS oferece suporte à criação de uma conexão com esse banco de dados. Embora *Painel do Navegador QGIS* seja a maneira mais simples e recomendada de conectar-se e usar bancos de dados, o QGIS fornece outras ferramentas para conectar-se a cada um deles e carregar suas tabelas:

- *Add PostgreSQL Layer…* or by typing `Ctrl+Shift+D`
- *Add Oracle Spatial Layer…* or by typing `Ctrl+Shift+O`
- *Add MS SQL Server Layer*
- *Add SAP HANA Spatial Layer…* or by typing `Ctrl+Shift+G`

These tools are accessible either from the *Manage Layers Toolbar* and the *Layer* ► *Add Layer* ► menu. Connecting to SpatiaLite database is described at *Camadas SpatiaLite*.

> **Dica**
>
> **Create connection to database from the QGIS Browser Panel**
>
> Selecting the corresponding database format in the Browser tree, right-clicking and choosing connect will provide you with the database connection dialog.

Most of the connection dialogs follow a common structure:

- a section with credentials information to connect to the database
- a section with options to tune which data can be requested in the database

Connecting to PostgreSQL

The first time you use a PostgreSQL data source, you must create a connection to a database that contains the data. Press the appropriate button as exposed above, opening the *PostgreSQL* tab of the *Data Source Manager* dialog. To access the connection manager, click on the *New* button to display the *Create a New PostgreSQL Connection* dialog.

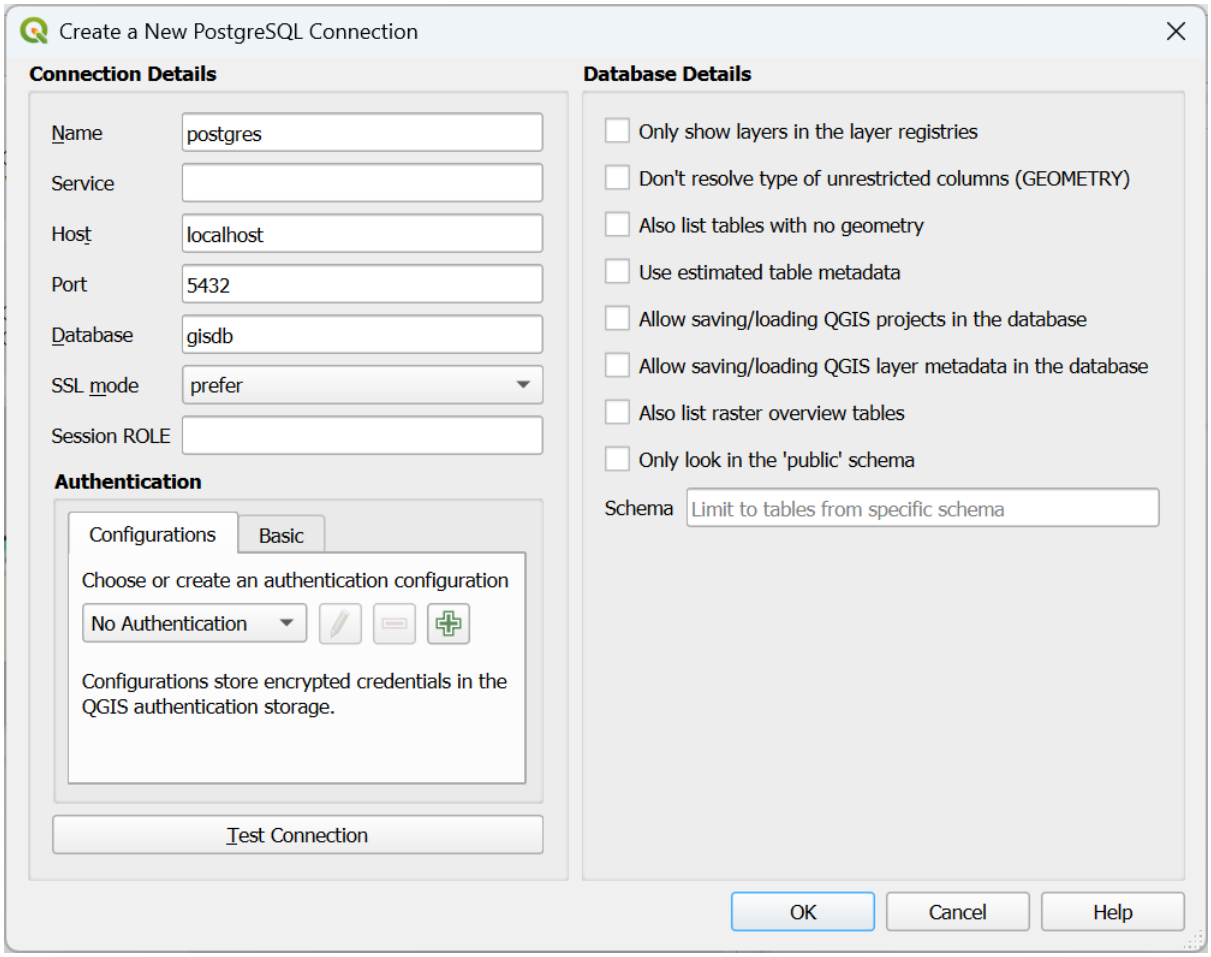


Figura11.12: Create a New PostgreSQL Connection Dialog

- *Name*: A name for this connection. It can be the same as *Database*.
- *Service*: Service parameter to be used alternatively to hostname/port (and potentially database). This can be defined in `pg_service.conf`. Check the *PostgreSQL Service connection file* section for more details.
- *Host*: Name of the database host. This must be a resolvable host name such as would be used to open a TCP/IP connection or ping the host. If the database is on the same computer as QGIS, simply enter *localhost* here.
- *Port*: Port number the PostgreSQL database server listens on. The default port for PostgreSQL is `5432`.
- *Banco de Dados*: Nome do banco de dados.
- *SSL mode*: SSL encryption setup. The following options are available:
  - *Prefer* (the default): I don't care about encryption, but I wish to pay the overhead of encryption if the server supports it.
  - *Require*: I want my data to be encrypted, and I accept the overhead. I trust that the network will make sure I always connect to the server I want.
  - *Verify CA*: I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.

  - Guilabel: 'Verificação Completa': Quero meus dados criptografados, e aceito as despesas gerais. Quero ter certeza de que me conecto a um servidor em que confio, e que é aquele que eu especifico.
  - guilabel: 'Permitir': Eu não me importo com a segurança, mas pagarei as despesas gerais de criptografia se o servidor insistir nisso.
  - guilabel:`Desativar': Eu não me importo com a segurança, e não quero pagar as despesas gerais de criptografia.

- *Session role*: used to set the current user identifier of the current session. This is useful to automatically give the ownership of a new object (table, view, function) to the session_role group and thus share ownership and associated rights with all members of the session_role group. Read more about session role.
- *Authentication*: For general details about the authentication dialog behavior, see *Autenticação*.

Opcionalmente, dependendo do tipo de banco de dados, você pode ativar as seguintes caixas de seleção:

- ☑ *Only show layers in the layer registries*
- ☑ :guilabel:`Não resolva o tipo de colunas irrestritas (GEOMETRIA) `
- ☑ *Also list tables with no geometry*: indicates that tables without geometry should also be listed by default.
- ☑ *Use estimated table metadata*: When initializing layers, various queries may be needed to establish the characteristics of the geometries stored in the database table. When this option is checked, these queries examine only a sample of the rows and use the table statistics, rather than the entire table. This can drastically speed up operations on large datasets, but may result in incorrect characterization of layers (e.g. the feature count of filtered layers will not be accurately determined) and may even cause strange behaviour if columns that are supposed to be unique actually are not.
- ☑ *Permitir salvar/carregar projetos QGIS no banco de dados* - mais detalhes *here*
- ☑ *Allow saving/loading QGIS layer metadata in the database* - more details *here*
- ☑ *Also list raster overview tables*
- ☑ *Only look in the 'public' schema*
- *Schema*: Allows to specify a single schema to limit a connection to. When set, only tables from the matching schema will be included in the browser panel and data source select for the connection. This can be used to limit the database work required to populate tables for a connection pointing to a large database store.

Once all parameters and options are set, you can test the connection by clicking the *Test Connection* button or apply it by clicking the *OK* button.

PostgreSQL Service connection file

The service connection file allows PostgreSQL connection parameters to be associated with a single service name. That service name can then be specified by a client and the associated settings will be used.

It's called `.pg_service.conf` under *nix systems (GNU/Linux, macOS etc.) and `pg_service.conf` on Windows.

O arquivo de serviço pode se parecer com o seguinte:

```
[water_service]
host=192.168.0.45
port=5433
dbname=gisdb
user=paul
password=paulspass

[wastewater_service]
host=dbserver.com
dbname=water
user=waterpass
```

Nota

There are two services in the above example: `water_service` and `wastewater_service`. You can use these to connect from QGIS, pgAdmin, etc. by specifying only the name of the service you want to connect to (without the enclosing brackets). If you want to use the service with `psql`, you can do `psql service=water_service`.

You can find all the PostgreSQL parameters here

Nota

If you don't want to save the passwords in the service file you can use the .pg_pass option.

Nota

QGIS Server and service

When using a service file and QGIS Server, you must configure the service on the server side as well. You can follow the QGIS Server documentation.

On *nix operating systems (GNU/Linux, macOS etc.) you can save the `.pg_service.conf` file in the user's home directory and PostgreSQL clients will automatically be aware of it. For example, if the logged user is `web`, `.pg_service.conf` should be saved in the `/home/web/` directory in order to directly work (without specifying any other environment variables).

You can specify the location of the service file by creating a `PGSERVICEFILE` environment variable (e.g. run the `export PGSERVICEFILE=/home/web/.pg_service.conf` command under your *nix OS to temporarily set the `PGSERVICEFILE` variable).

You can also make the service file available system-wide (all users) either by placing the `.pg_service.conf` file in `pg_config --sysconfdir` or by adding the `PGSYSCONFDIR` environment variable to specify the directory containing the service file. If service definitions with the same name exist in the user and the system file, the user file takes precedence.

Aviso

There are some caveats under Windows:

- The service file should be saved as `pg_service.conf` and not as `.pg_service.conf`.
- The service file should be saved in Unix format in order to work. One way to do it is to open it with Notepad++ and *Edit* ► *EOL Conversion* ► *UNIX Format* ► *File save*.
- Você pode adicionar variáveis ambientais de várias maneiras; um testado, conhecido por funcionar de forma confiável, é **:menuelection:`Painel de controle --> Sistema e segurança --> Sistema --> Configurações avançadas do sistema --> Variáveis de ambiente`** adicionando `PGSERVICEFILE` com o caminho - por exemplo `C:\Users\John\pg_service.conf`
- After adding an environment variable you may also need to restart the computer.

Conectando ao Oracle Spatial

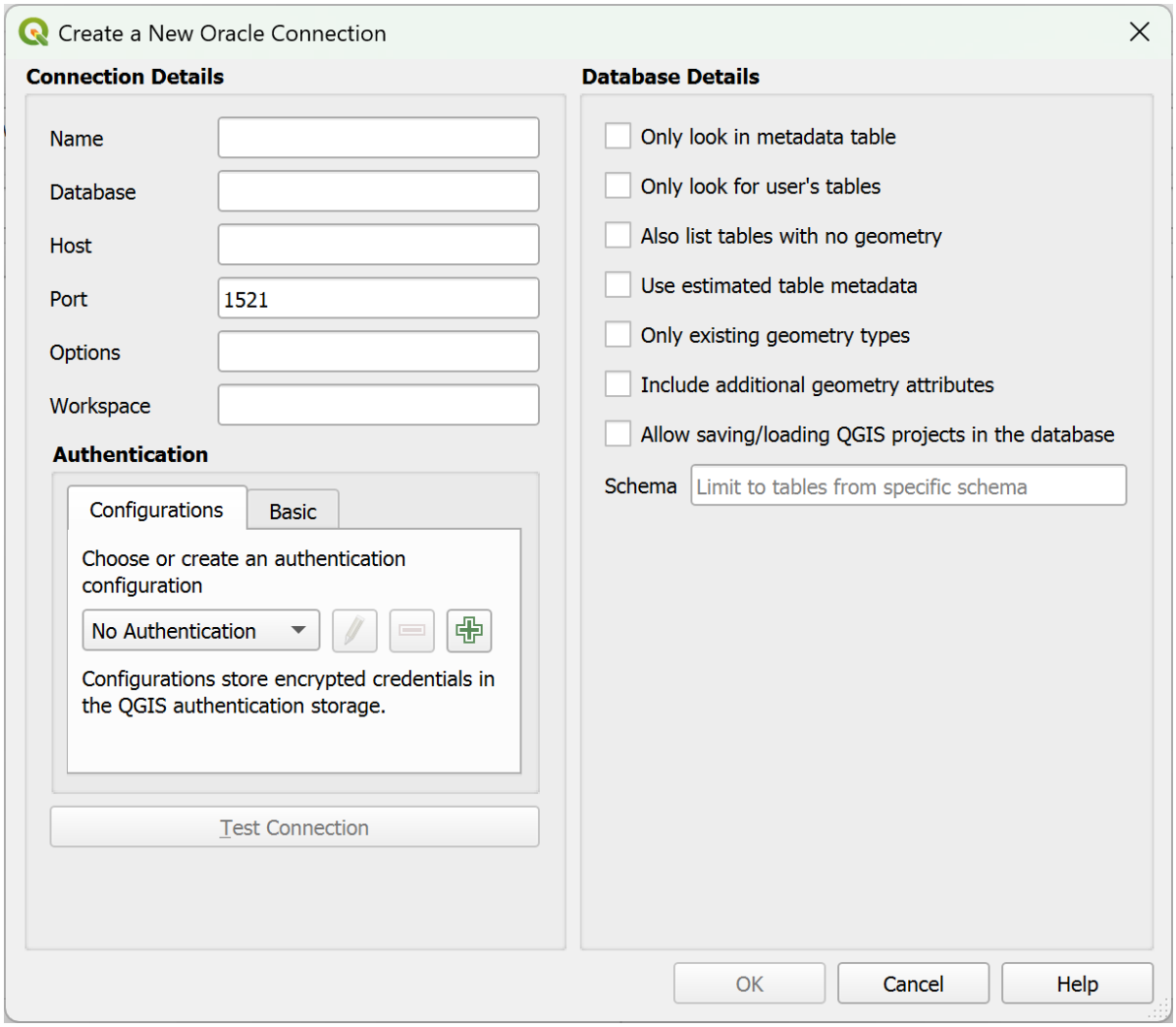


Figura11.13: Create a New Oracle Connection Dialog

The spatial features in Oracle Spatial aid users in managing geographic and location data in a native type within an Oracle database. The connection dialog proposes:

- *Name*: A name for this connection. It can be the same as *Database*;
- *Database*: SID or SERVICE_NAME of the Oracle instance;
- *Host*: The name of the database host;
- *Port*: Port number the Oracle database server listens on. The default port is `1521`;
- *Options*: Oracle connection specific options (e.g. OCI_ATTR_PREFETCH_ROWS, OCI_ATTR_PREFETCH_MEMORY). The format of the options string is a semicolon separated list of option names or option=value pairs;
- *Workspace*: Workspace to switch to;
- *Authentication*: For general details about the authentication dialog behavior, see *Autenticação*.

Opcionalmente, você pode ativar as seguintes caixas de seleção:

- *Only look in metadata table*: restricts the displayed tables to those that are in the `all_sdo_geom_metadata` view. This can speed up the initial display of spatial tables.

- *Only look for user's tables*: when searching for spatial tables, restricts the search to tables that are owned by the user.
- *Also list tables with no geometry*: indicates that tables without geometry should also be listed by default.
- *Use estatísticas estimadas da tabela para os metadados da camada*: quando a camada é configurada, vários metadados são necessários para a tabela Oracle. Isso inclui informações como a contagem de linhas da tabela, tipo de geometria e extensões espaciais dos dados na coluna geometria. Se a tabela contiver um grande número de linhas, a determinação desses metadados poderá ser demorada. Ao ativar esta opção, as seguintes operações rápidas de metadados da tabela são concluídas: A contagem de linhas é determinada em `all_tables.num_rows`. As extensões de tabela são sempre determinadas com a função SDO_TUNE.EXTENTS_OF, mesmo se um filtro de camada for aplicado. A geometria da tabela é determinada a partir das primeiras 100 linhas de geometria não nula da tabela.
- *Only existing geometry types*: only lists the existing geometry types and don't offer to add others.
- :guilabel:`Incluir atributos adicionais de geometria `.
- *Permitir salvar/carregar projetos QGIS no banco de dados* - mais detalhes *here*
- *Schema*: Allows to specify a single schema to limit a connection to. When set, only tables from the matching schema will be included in the browser panel and data source select for the connection. This can be used to limit the database work required to populate tables for a connection pointing to a large database store.

Dica

Camadas Espaciais Oracle

Normally, an Oracle Spatial layer is defined by an entry in the **USER_SDO_METADATA** table.

To ensure that selection tools work correctly, it is recommended that your tables have a **primary key**.

Connecting to MS SQL Server

As mentioned in *Criando uma conexão armazenada* QGIS allows you to create MS SQL Server connection through *Data Source Manager*.

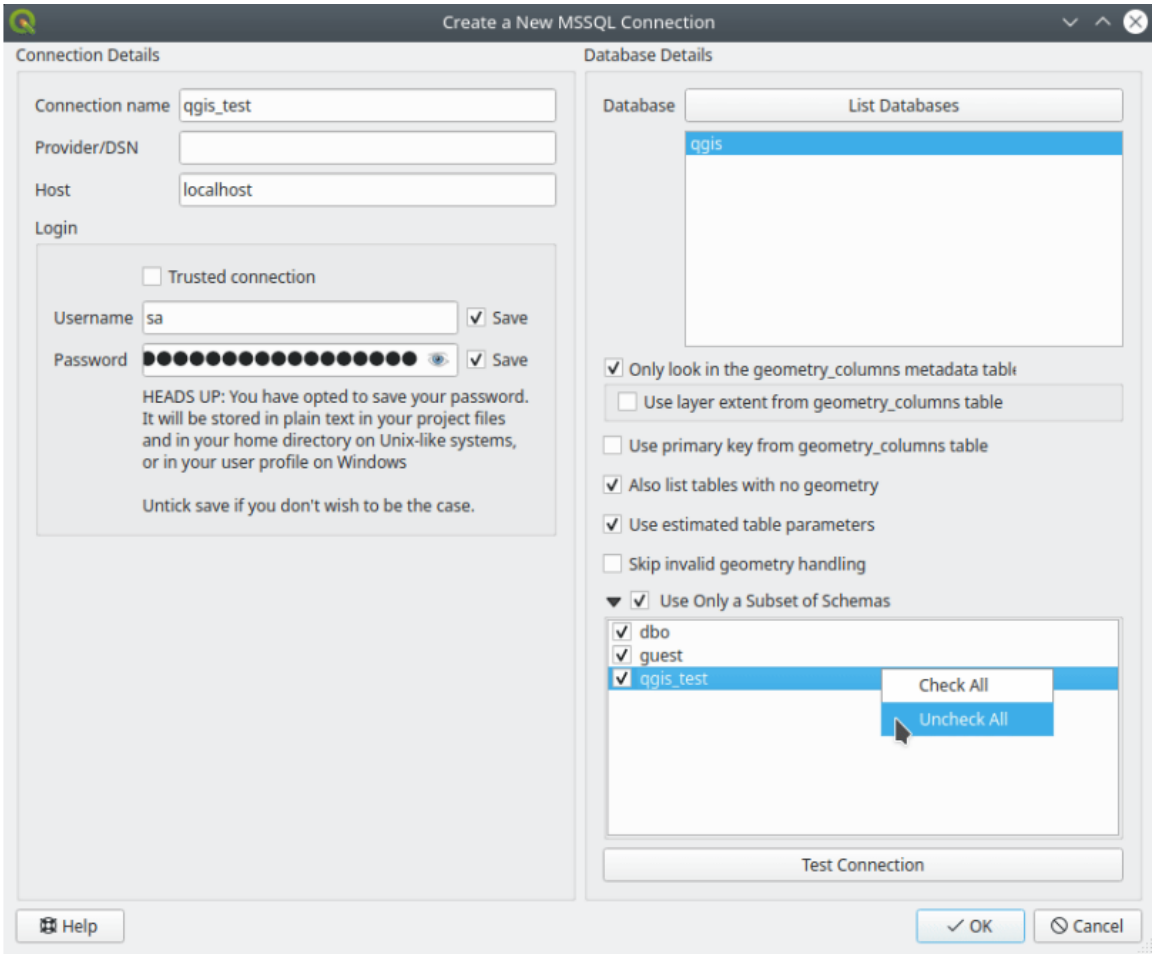


Figura11.14: MS SQL Server Connection

To create a new MS SQL Server connection, you need to provide some of the following information in the *Connection Details* dialog:

- *Connection name*
- *Provider/DSN*
- *Host*
- *Login* information. You can choose to *Save* your credentials.

Navigate to the *Database Details* section and click the *List Databases* button to view the available datasets. Select datasets that you want, then press *OK*. Optionally, you can also perform a *Test Connection*. Once you click *OK* the *Create a New MS SQL Server Connection* dialog will close and in the *Data Source Manager* press *Connect*, select a layer and then click *Add*.

Optionally, you can activate the following options:

- *Only look in the geometry_columns metadata table*: restricts the available tables to the ones in the `geometry_columns` metadata table when scanning for tables. This can speed up the table scanning.
- *Use layer extent from geometry_columns table*: this option, dependent on the previous one, allows QGIS to skip extent calculation when loading layers and thus lowering the amount of time needed to load them. It relies on extent manually specified using additional QGIS-specific columns (`qgis_xmin`, `qgis_xmax`, `qgis_ymin`, `qgis_ymax`) in the `geometry_columns` table.
- *Use primary key from geometry_columns table*: allows QGIS to skip primary key calculation for views when loading them, thus lowering the amount of time needed to load them. It relies on names manually filled

in a QGIS-specific `qgis_pkey` column set in the `geometry_columns` table. If more than one column is used for the primary key, they should be filled as comma separated values.

- *Also list table with no geometry*: tables without a geometry column attached will also be shown in the available table list.
- *Use estimated table parameters*: only estimated table metadata will be used. This avoids a slow table scan, but may result in incorrect layer properties such as layer extent.
- *Skip invalid geometry handling*: all handling of records with invalid geometry will be disabled. This speeds up the provider, however, if any invalid geometries are present in a table then the result is unpredictable and may include missing records. Only check this option if you are certain that all geometries present in the database are valid, and any newly added geometries or tables will also be valid.
- *Use only a Subset of Schemas* will allow you to filter schemas for MS SQL connection. If enabled, only checked schemas will be displayed. You can right-click to *Check* or *Uncheck* any schema in the list.

Connecting to SAP HANA

Nota

You require the SAP HANA Client to connect to a SAP HANA database. You can download the SAP HANA Client for your platform at the SAP Development Tools website.

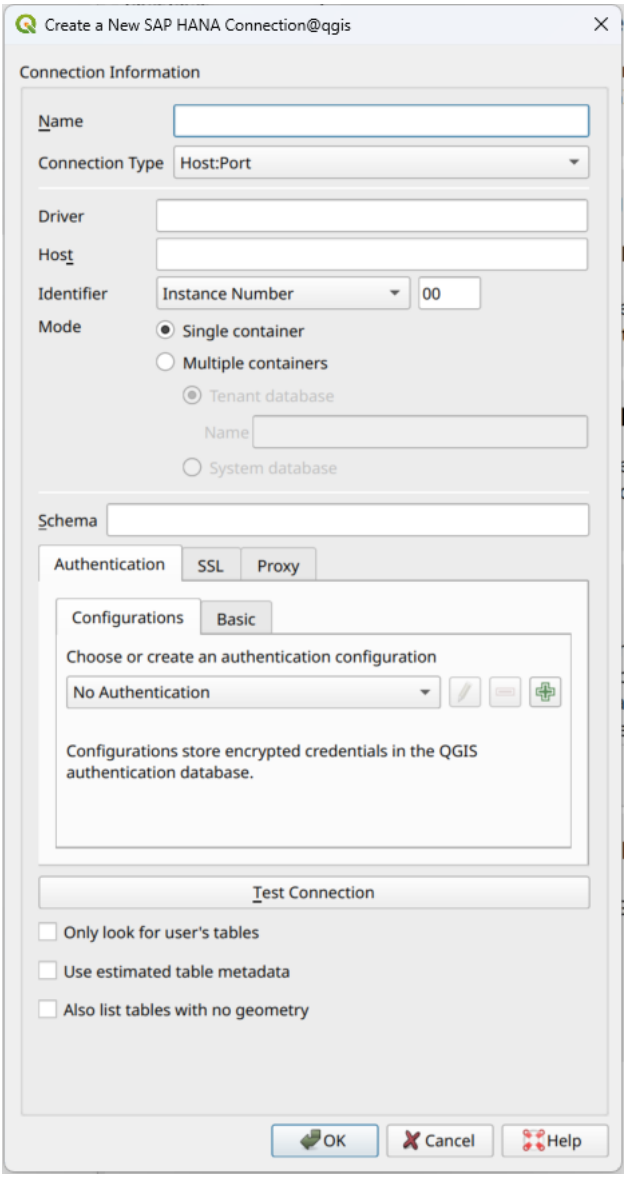


Figura11.15: Create a New SAP HANA Connection Dialog

Os seguintes parâmetros podem ser inseridos:

- *Nome*: Um nome para esta conexão.
- *Driver* : The name of the HANA ODBC driver. It is `HDBODBC` if you are using 64-bit QGIS, `HDBODBC32` if you are using 32-bit QGIS. The appropriate driver name is entered automatically.
- *Driver* : Either the name under which the SAP HANA ODBC driver has been registered in `/etc/odbcinst.ini` or the full path to the SAP HANA ODBC driver. The SAP HANA Client installer will install the ODBC driver to `/usr/sap/hdbclient/libodbcHDB.so` by default.
- *Host*: The name of the database host.
- *Identifier*: Identifies the instance to connect to on the host. This can be either *Instance Number* or *Port Number*. Instance numbers consist of two digits, port numbers are in the range from 1 to 65,535.
- *Mode*: Specifies the mode in which the SAP HANA instance runs. This setting is only taken into account if *Identifier* is set to *Instance Number*. If the database hosts multiple containers, you can either connect to a tenant with the name given at *Tenant database* or you can connect to the system database.

- *Schema*: This parameter is optional. If a schema name is given, QGIS will only search for data in that schema. If this field is left blank, QGIS will search for data in all schemas.
- *Authentication*: For general details about the authentication dialog behavior, see *Autenticação*.
- *SSL Settings*
  - *Enable TLS/SSL encryption*: Enables TLS 1.1 - TLS1.2 encryption. The server will choose the highest available.
  - *Provider*: Specifies the cryptographic library provider used for SSL communication. *sapcrypto* should work on all platforms, *openssl* should work on , *mscrypto* should work on and *commoncrypto* requires CommonCryptoLib to be installed.
  - *Validate SSL certificate*: If checked, the SSL certificate will be validated using the truststore given in *Trust store file with public key*.
  - *Override hostname in certificate*: Specifies the host name used to verify server's identity. The host name specified here verifies the identity of the server instead of the host name with which the connection was established. If you specify `*` as the host name, then the server's host name is not validated. Other wildcards are not permitted.
  - *Keystore file with private key*: Currently ignored. This parameter might allow to authenticate via certificate instead via user and password in future.
  - *Trust store file with public key*: Specifies the path to a trust store file that contains the server's public certificates if using OpenSSL. Typically, the trust store contains the root certificate or the certificate of the certification authority that signed the server's public certificates. If you are using the cryptographic library CommonCryptoLib or msCrypto, then leave this property empty.
- *Only look for user's tables*: If checked, QGIS searches only for tables and views that are owned by the user that connects to the database.
- *Use estimated table metadata*: If checked, estimated table metadata will be used if available. For large tables, this avoids slow table loads and potentially expensive computations, but may result in incorrect layer properties such as layer extent. The fast extent estimation is available starting with QRC1/2024 and SP8 in HANA Cloud and HANA On-Premise respectively.
- *Also list tables with no geometries*: If checked, QGIS searches also for tables and views that do not contain a spatial column.

Dica

Connecting to SAP HANA Cloud

If you'd like to connect to an SAP HANA Cloud instance, you usually must set *Port Number* to `443` and check *Enable TLS/SSL encryption*.

Loading a Database Layer

Once you have one or more connections defined to a database (see section *Criando uma conexão armazenada*), you can load layers from it. Of course, this requires that data are available. See section *Importing Data into PostgreSQL* for a discussion on importing data into a PostgreSQL database.

Para carregar uma camada de um banco de dados, você pode executar as seguintes etapas:

1. Open the corresponding tab of the database in the *Data Source Manager* dialog.
2. Choose the connection name from the drop-down list and press *Connect*.
3. The table below will be filled with your data grouped by schema, with a number of metadata information helpful for loading.

4. Select or unselect *Also list tables with no geometry*.
5. Optionally, use some *Search Options* to reduce the list of tables to those matching your search. You can also set this option before you hit the *Connect* button, speeding up the database fetching.
6. Encontre a(s) camada(s) que deseja adicionar na lista de camadas disponíveis.
7. Select it by clicking on it. You can select multiple layers by holding down the `Shift` or `Ctrl` key while clicking.

   Layers can be selected only if they have no warning icon at the left-hand side of their first column. This may indicate an issue to detect:

   - features geometry type: in which case you can select the appropriate one in the drop-dowmn list of the *Spatial Type* column
   - layer CRS: you can enter the correct code in the *SRID* column
   - layer's primary key, in order to unequivocally identify each feature: this can be fixed by selecting one or more attributes in the drop-down list at the corresponding *Feature id* column.

   > **Dica**
   >
   > **Use the first column to store "primary keys" for views**
   >
   > Since PostgreSQL views don't support primary keys, a unique attribute or combination of attributes should always be selected. To help users to speed-up workflows, QGIS automatically selects the first attribute in the view. Therefore, users can define their views in a way that a unique column is in the first position of the view's definition. In this way, the view will be loaded with no extra interaction and the warning icon will never appear.

8. If applicable, use the *Set Filter* button (or double-click the layer) to start the *Query Builder* dialog (see section *Criador de consultas*) and define which features to load from the selected layer. The filter expression appears in the `sql` column. This restriction can be removed or edited in the *Layer Properties* ► *General* ► *Provider Feature Filter* frame.
9. The checkbox in the `Select at id` column that is activated by default gets the feature ids without the attributes and generally speeds up the data loading.
10. Click on the *Add* button to add the layer to the map.

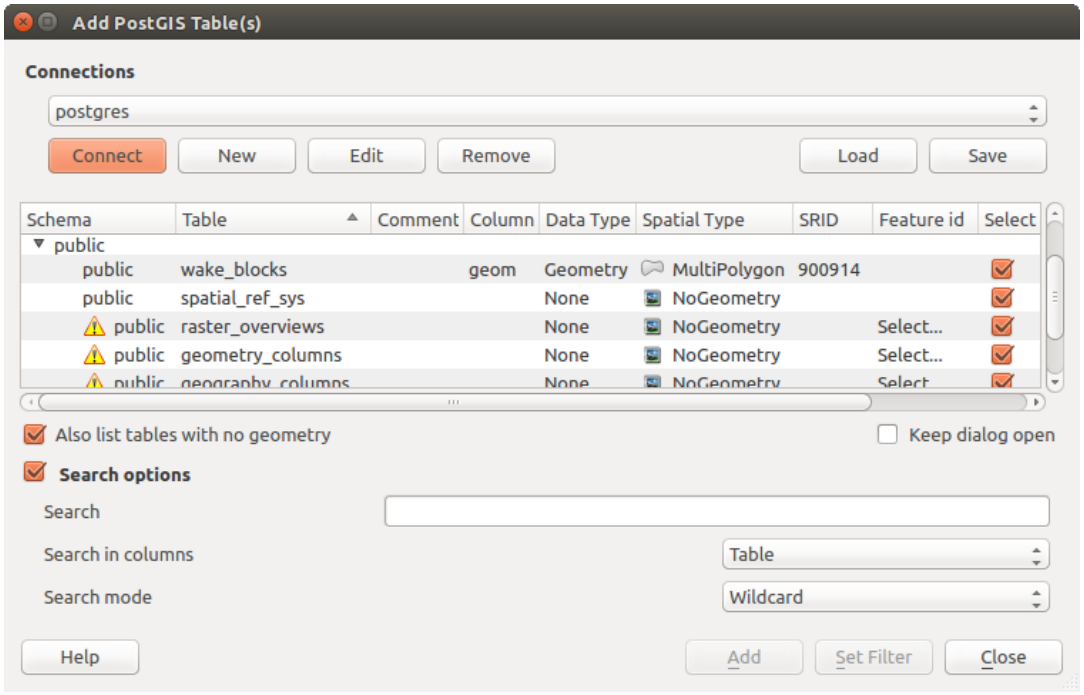


Figura11.16: Add PostgreSQL Table(s) Dialog

Dica

Use the Browser Panel to speed up loading of database table(s)

A adição de tabelas de BD do *Gerenciador de Fonte de Dados* às vezes pode ser demorado, pois o QGIS busca estatísticas e propriedades (por exemplo, tipo e campo de geometria, SRC, número de recursos) para cada tabela com antecedência. Para evitar isso, uma vez que *the connection is set*, é melhor usar o *Browser Panel* ou o *DB Manager* para arrastar e soltar as tabelas do banco de dados na tela do mapa.

11.1.4 The Layer Metadata Search Panel

By default, QGIS can retrieve layers metadata from the connections or data providers that allow metadata storage (more details on *saving metadata to the database*). The *Metadata search* panel allows to browse the layers by their metadata and add them to the project (either with a double-click or the *Add* button). The list can be filtered:

- by text, watching a set of metadata properties (identifier, title, abstract)
- by spatial extent, using the current *project extent* or the map canvas extent
- by the layer (geometry) type

Nota

The sources of metadata are implemented through a layer metadata provider system that can be extended by plugins.

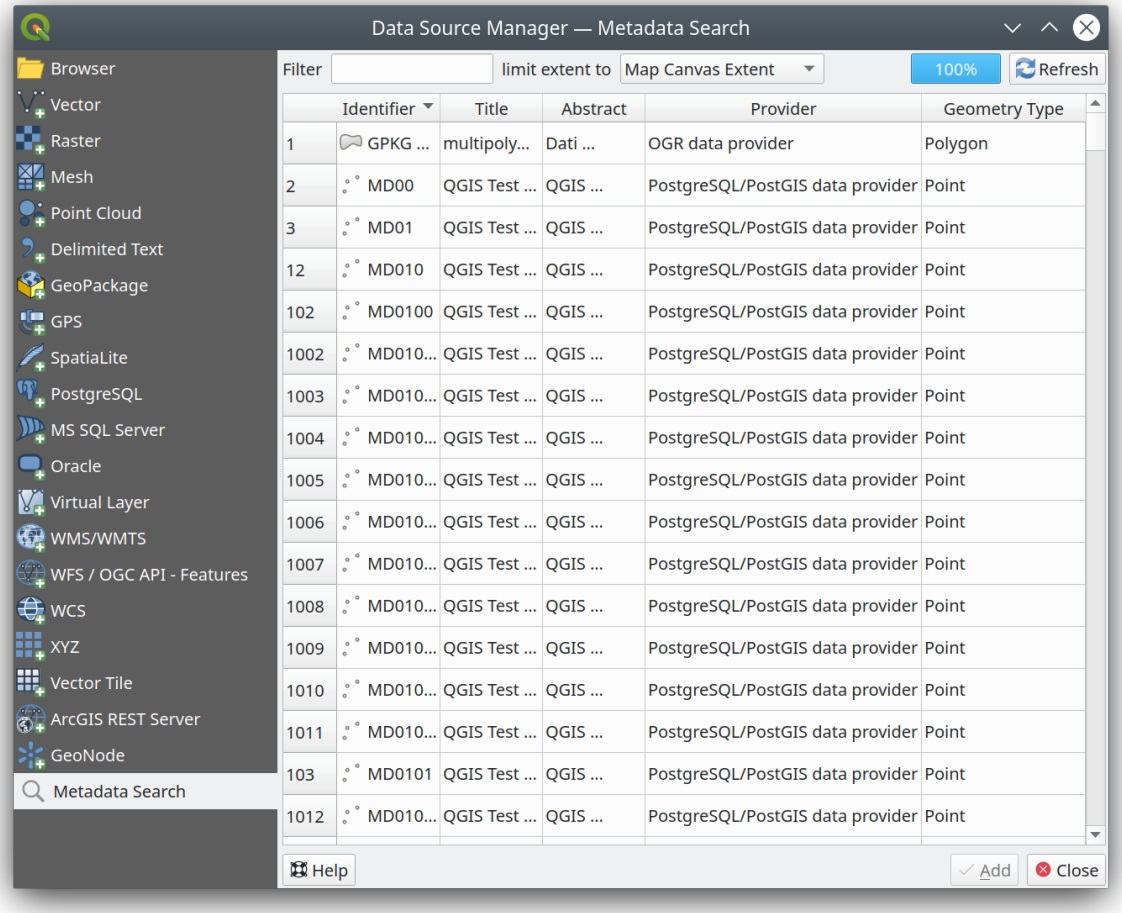


Figura11.17: Layer Metadata Search Panel

11.1.5 Formatos personalizados QGIS

O QGIS propõe dois formatos personalizados:

- Temporary Scratch Layer: a memory layer that is bound to the project (see *Creating a new Temporary Scratch Layer* for more information)
- Virtual Layers: a layer resulting from a query on other layer(s) (see *Criando camadas virtuais* for more information)

11.1.6 QLR - QGIS Layer Definition File

Layer definitions can be saved as a *Layer Definition File* (QLR - `.qlr`) using *Export* ► *Save As Layer Definition File…* in the layer context menu.

The QLR format makes it possible to share “complete” QGIS layers with other QGIS users. QLR files contain links to the data sources and all the QGIS style information necessary to style the layer.

QLR files are shown in the Browser Panel and can be used to add layers (with their saved styles) to the Layers Panel. You can also drag and drop QLR files from the system file manager into the map canvas.

Available actions for QLR files in the Browser Panel are:

- *Export Layer* ► *To file*

- *Adicionar camada ao projeto*
- *Propriedades da Camada…*

11.1.7 Conectando-se a serviços da web

With QGIS you can get access to different types of OGC web services (WM(T)S, WFS(-T), WCS, CSW, …). Thanks to QGIS Server, you can also publish such services. QGIS-Server-manual contains descriptions of these capabilities.

Using Vector Tiles services

Vector Tile services can be added via the *Vector Tiles* tab of the *Data Source Manager* dialog or the contextual menu of the *Vector Tiles* entry in the *Browser* panel. Services can be either a *New Generic Connection…* or a *New ArcGIS Vector Tile Service Connection…*.

Você cria um serviço, adicionando:

- um *Nome*
- a *Style URL*: a URL to a MapBox GL JSON style configuration. If provided, then that style will be applied whenever the layers from the connection are added to QGIS. In the case of Arcgis vector tile service connections, the URL overrides the default style configuration specified in the server configuration.

  You can load vector tiles directly from a *Style URL*. The data source is automatically parsed from the style, and URLs with multiple sources are supported. That makes *Source URL* optional.
- the *Source URL*: of the type `http://example.com/{z}/{x}/{y}.pbf` for generic services and `http://example.com/arcgis/rest/services/Layer/VectorTileServer` for ArcGIS based services. The service must provide tiles in `.pbf` format.
- the *Min. Zoom Level* and the *Max. Zoom Level*. Vector Tiles have a pyramid structure. By using these options you have the opportunity to individually generate layers from the tile pyramid. These layers will then be used to render the Vector Tile in QGIS.

  For Mercator projection (used by OpenStreetMap Vector Tiles) Zoom Level 0 represents the whole world at a scale of 1:500.000.000. Zoom Level 14 represents the scale 1:35.000.
- the *authentication* configuration if necessary
- a *Referer*

Fig.11.18 shows the dialog with the Vector Tiles service configuration.

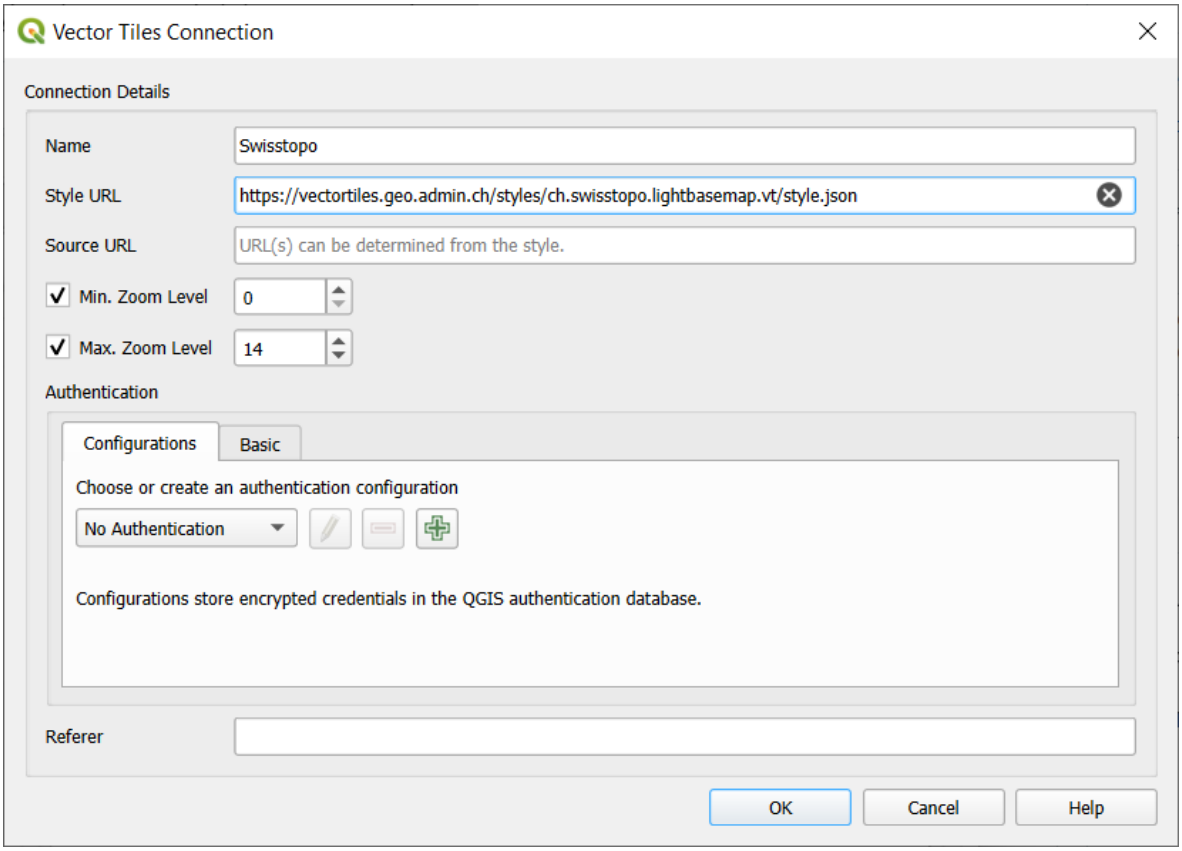


Figura11.18: Vector Tiles - Service configuration

Configurations can be saved to `.XML` file (*Save Connections*) through the *Vector Tiles* entry in *Data Source Manager* dialog or its context menu in the *Browser* panel. Likewise, they can be added from a file (*Load Connections*).

Once a connection to a vector tile service is set, it's possible to:

- *Edit* the vector tile connection settings
- *Remover* a conexão
- From the *Browser* panel, right-click over the entry and you can also:
  - *Add layer to project*: a double-click also adds the layer
  - View the *Layer Properties...* and get access to metadata and a preview of the data provided by the service. More settings are available when the layer has been loaded into the project.

Using XYZ Tile services

XYZ Tile services can be added via the *XYZ* tab of the *Data Source Manager* dialog or the contextual menu of the *XYZ Tiles* entry in the *Browser* panel. By default, QGIS provides some default and ready-to-use XYZ Tiles services:

- *Mapzen Global Terrain*, allowing an immediate access to global DEM source for the projects. More details and resources at https://registry.opendata.aws/terrain-tiles/
- *OpenStreetMap* to access the world 2D map. Fig.11.19 shows the dialog with the OpenStreetMap XYZ Tile service configuration.

To add a new service, press *New* (respectively *New Connection* from the Browser panel) and provide:

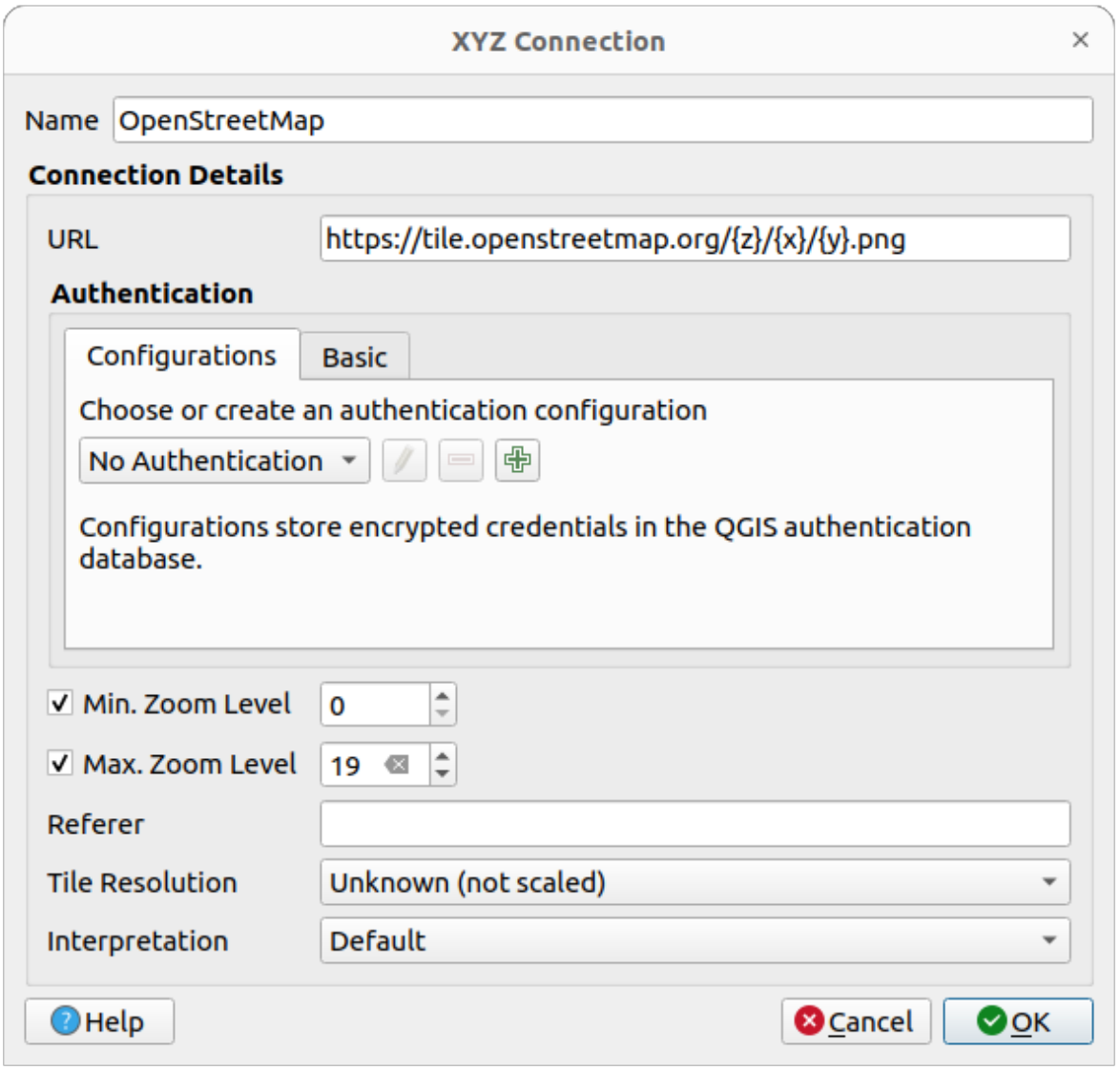


Figura11.19: XYZ Tiles - OpenStreetMap configuration

- um *Nome*
- the *URL*, you can add `http://example.com/{z}/{x}/{y}.png` or `file:///local_path/{z}/{x}/{y}.png`
- the *authentication* configuration if necessary
- the *Min. Zoom level* and *Max. Zoom level*
- a *Referer*
- the *Tile Resolution*: possible values are *Unknown (not scaled)*, *Standard (256x256 / 96DPI)* and *High (512x512 / 192DPI)*
- *Interpretation*: converts WMTS/XYZ raster datasets to a raster layer of single band float type following a predefined encoding scheme. Supported schemes are *Default* (no conversion is done), *MapTiler Terrain RGB* and *Terrarium Terrain RGB*. The selected converter will translate the RGB source values to float values for each pixel. Once loaded, the layer will be presented as a single band floating point raster layer, ready for styling using QGIS usual *raster renderers*.

Press *OK* to establish the connection. It will then be possible to:

- *Add* the new layer to the project; it is loaded with the name given in the settings.

- *Edit* the XYZ connection settings
- *Remover* a conexão
- From the *Browser* panel, right-click over the entry and you can also:
  - *Exportar camada…* ► *Para Arquivo*, *saving it as a raster*
  - *Add layer to project*: a double-click also adds the layer
  - View the *Layer Properties…* and get access to metadata and a preview of the data provided by the service. More settings are available when the layer has been loaded into the project.

Configurations can be saved to `.XML` file (*Save Connections*) through the *XYZ* entry in *Data Source Manager* dialog or its contextual menu in the *Browser* panel. Likewise, they can be added from a file (*Load Connections*).

The XML file for OpenStreetMap looks like this:

```
<!DOCTYPE connections>
<qgsXYZTilesConnections version="1.0">
  <xyztiles url="https://tile.openstreetmap.org/{z}/{x}/{y}.png"
   zmin="0" zmax="19" tilePixelRatio="0" password="" name="OpenStreetMap"
   username="" authcfg="" referer=""/>
</qgsXYZTilesConnections>
```

Dica

Loading XYZ tiles without creating a connection

It is also possible to add XYZ tiles to a project without necessarily storing its connection settings in you user profile (e.g. for a dataset you may need once). In the *Data Source Manager* ► *XYZ* tab, edit any properties in the *Connection Details* group. The *Name* field above should turn into `Custom`. Press *Add* to load the layer in the project. It will be named by default `XYZ Layer`.

Examples of XYZ Tile services:

- OpenStreetMap Monochrome: *URL*: `http://tiles.wmflabs.org/bw-mapnik/{z}/{x}/{y}.png`, *Min. Zoom Level*: 0, *Max. Zoom Level*: 19.
- Google Maps: *URL*: `https://mt1.google.com/vt/lyrs=m&x={x}&y={y}&z={z}`, *Min. Zoom Level*: 0, *Max. Zoom Level*: 19.
- Temperatura do Mapa Meteorológico Aberto: *URL*: `http://tile.openweathermap.org/map/temp_new/{z}/{x}/{y}.png?appid={api_key}` *Nível Mín. de Zoom*: 0, : guilabel: Nível Máx. de Zoom`: 19.

Using ArcGIS REST Servers

An ArcGIS REST Server can host many different types of web services (feature service, map service, image service, …). ArcGIS REST Servers can be added via the *ArcGIS REST Server* tab of the *Data Source Manager* dialog or the contextual menu of the *ArcGIS REST Servers* entry in the *Browser* panel:

1. Press *New* (respectively *New Connection*) and provide:
   - a *Name*: A name for the connection.
   - the *URL*: Main address of the ArcGIS REST Server.
   - a *Prefix*: Used to specify the proxy prefix in the URL, which is necessary for some ArcGIS servers that use web proxy prefixes.
   - a *Community endpoint URL*: Endpoint URL for ArcGIS Online or Portal for ArcGIS, used to access content groups.

- a *Content endpoint URL*: Endpoint URL for the content service. This is used to access items in ArcGIS Online or Portal for ArcGIS.
- the *authentication* credentials if necessary.
- a *Referer*: The referer URL to be sent in the HTTP headers when making requests to the server. This may be required by some servers for authentication purposes.

> **Nota**
>
> ArcGIS Feature Service connections which have their corresponding Portal endpoint URLS set can be explored by content groups in the browser panel.
>
> Se uma conexão tiver os terminais do Portal definidos, expandir a conexão no navegador mostrará uma pasta “Grupos” e “Serviços”, em vez da lista completa de serviços normalmente exibida. Expandir a pasta de grupos mostrará uma lista de todos os grupos de conteúdo dos quais o usuário é membro, cada um dos quais pode ser expandido para mostrar os itens de serviço pertencentes a esse grupo.

2. Then press *OK* to validate the configuration settings. These configurations can be saved to `.XML` file (*Save*). Likewise, they can be added from a file (*Load*).
3. Once a connection to an ArcGIS REST Server is set, it is possible to:
   - *Edit* the ArcGIS REST Server connection settings
   - *Remover* a conexão
   - *Refresh* the connection
4. Press *Connect* to request the server and display its contents. They are organized in a tree structure whose nodes depend on the connection’s endpoint.

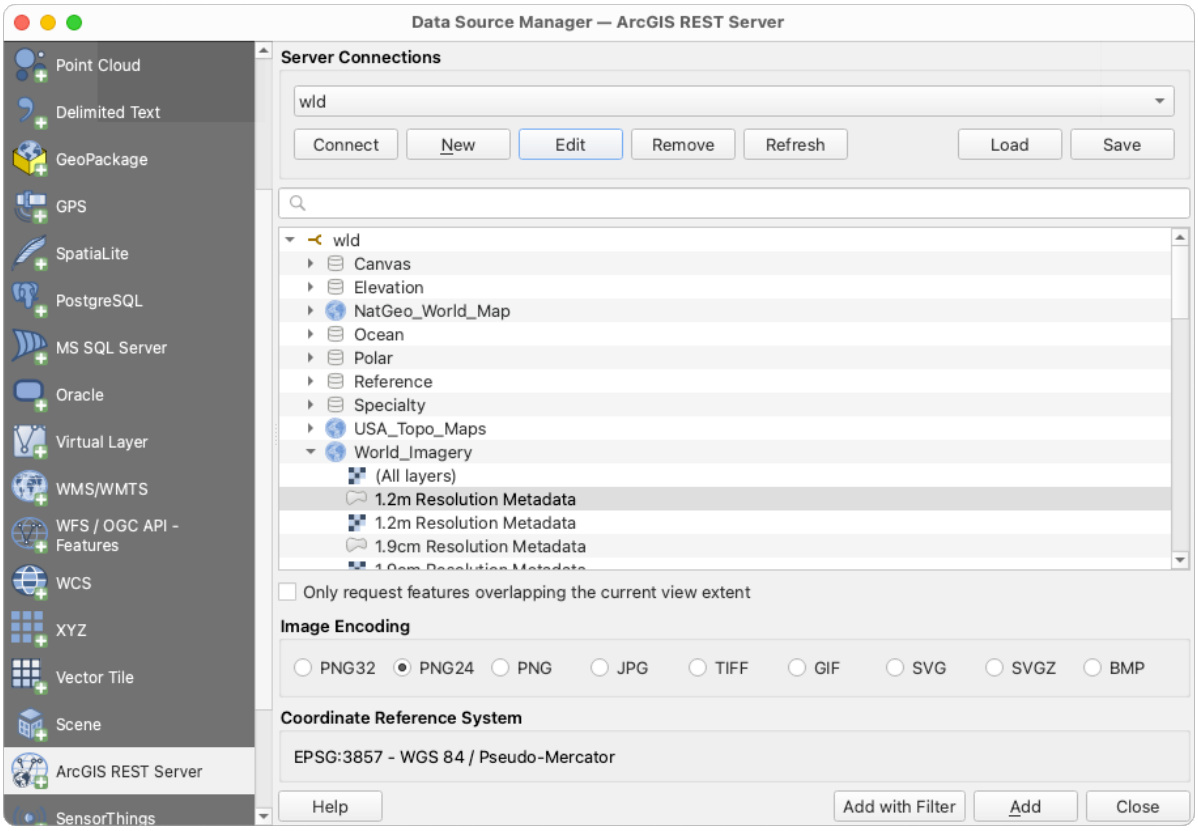


Figura11.20: Data Source Manager - ArcGIS REST Server

5. Expand the tree to find and select the layers of interest. Their *Coordinate Reference System* is displayed at the bottom of the dialog.
6. For raster-based layers, you can select the *Image encoding* to use among a number of image formats advertized by the target service (e.g., `PNG`, `JPG`, `GIF`, `SVG`, `SVGZ`, ...).
7. To add the selected layers to the map canvas, press *Add* button.

   Because layers can sometimes load and render slowly on the client side, applying a filter to restrict the features retrieved from the service can significantly improve performance, since only the filtered features are requested from the server. This can be done by:

   - checking *Only request features overlapping the current view extent*
   - for vector layers (feature service), pressing *Add with filter* to apply attribute-based filters to the layer with *O Construtor de string de expressão* functions.

     In the *Browser* panel, right-click the ArcGIS REST Server layer and select *Add Filtered Layer to Project* will also open the builder dialog.

Most of the above tools to connect and access layers on an ArcGIS REST Server are also available from the *Browser* panel, within the contextual menu of the target node item. Depending on the node, you can also:

- *View Service Info* which will open the default web browser and display information on the requested service.
- View the *Layer Properties...* and get access to the metadata and a preview of the data provided by the service. More settings are available when the layer has been loaded into the project.

Using 3D tiled scene services

QGIS supports multiple formats of 3D tiled datasets, grouped together as “tiled scenes”. These include Cesium 3D Tiles and Quantized Mesh tiles.

To load a tiled scene dataset into QGIS, use the *Scene* tab in the *Data Source Manager* dialog.

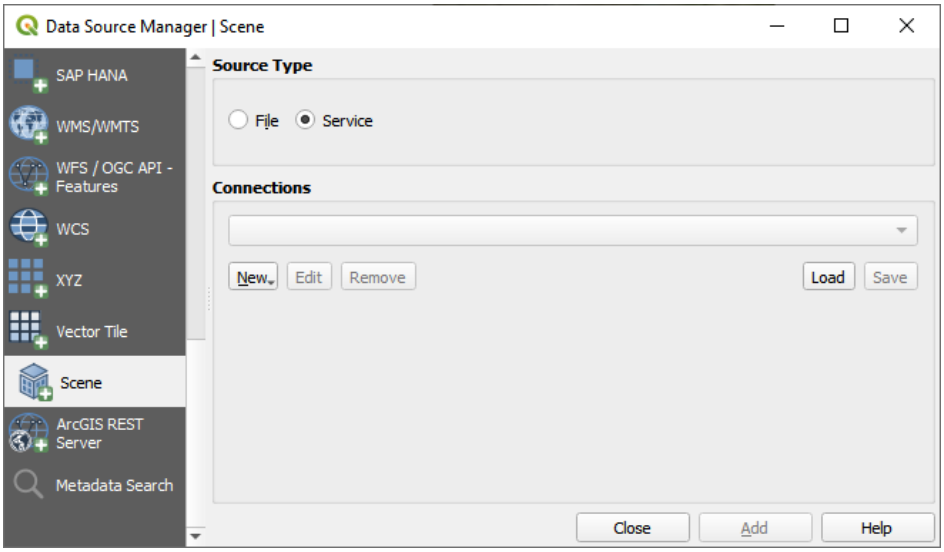


Figura11.21: Data Source Manager - Scene

Create a connection by clicking on *New*. You can add a *New Cesium 3D Tiles Connection* or a *New Quantized Mesh Connection*.

Choose a *Name* and set the *URL* to the URL of a layer description JSON file.

The URL may be remote (e.g. `http://example.com/tileset.json`) or local (e.g. `file:///path/to/tiles/tileset.json`).

The *authentication* configuration if necessary.

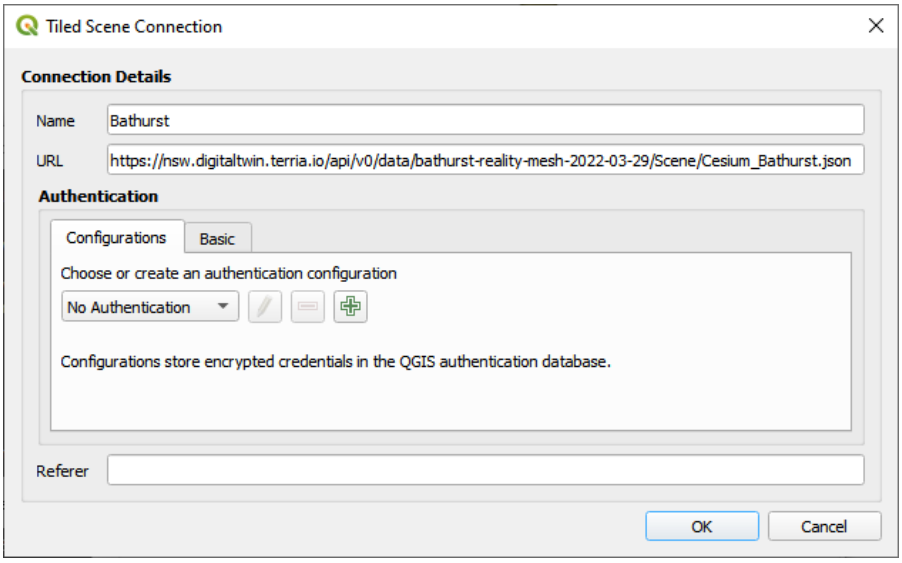


Figura11.22: Tiled Scene Connection

You can also add the service from *Browser Panel*.

After creating new connection you are able to *Add* the new layer to your map.

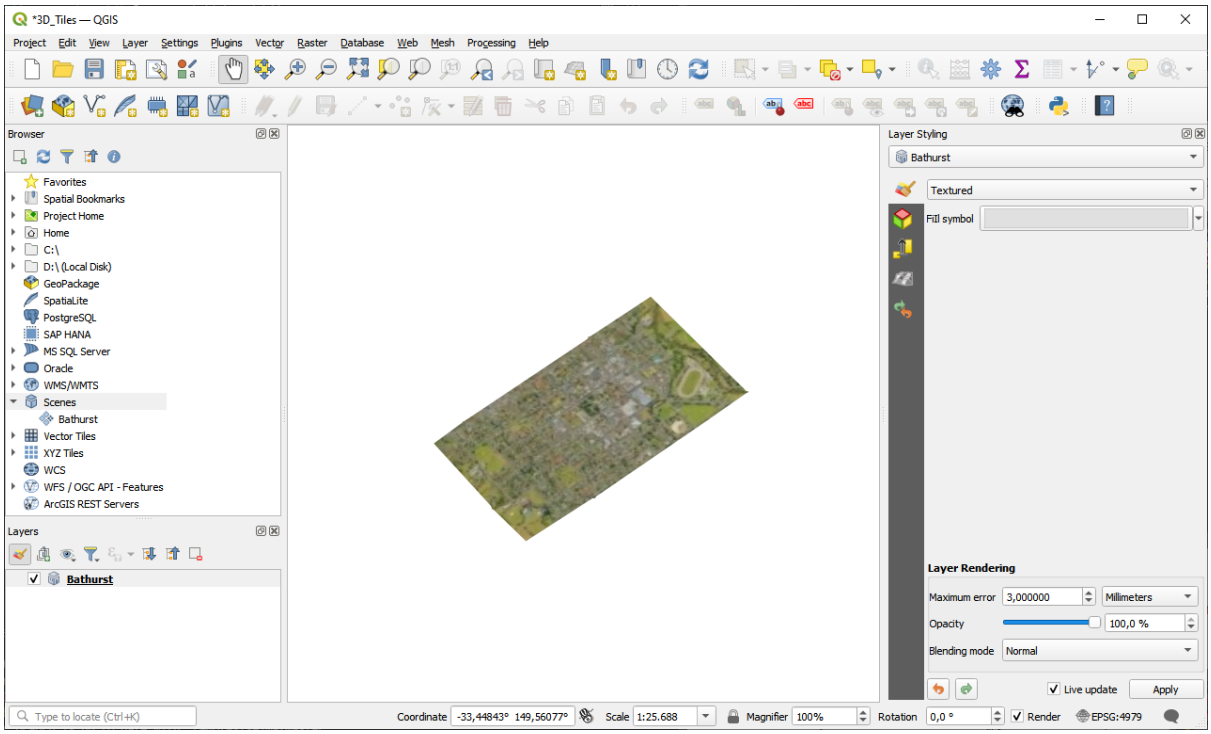


Figura11.23: 3D Tiles Layer - Textured

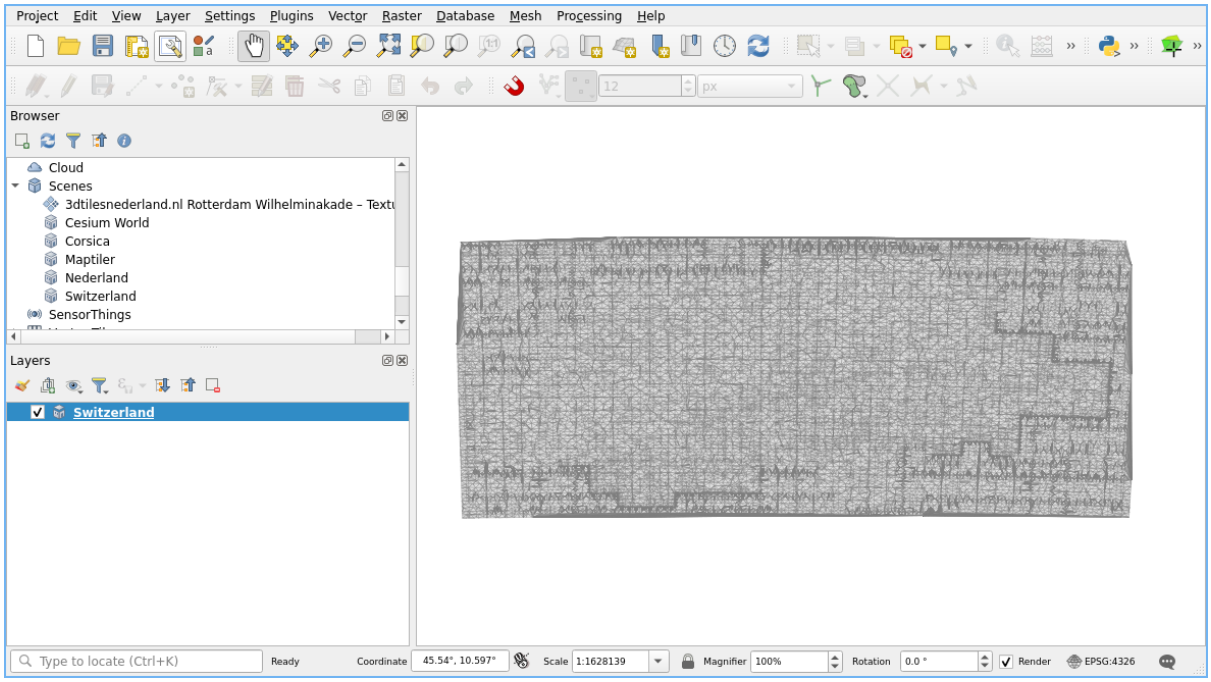


Figura11.24: Quantized Mesh layer

Using Cloud Connections

QGIS supports connections to cloud services like Alibaba Cloud OSS, Amazon S3, Google Cloud Storage, Microsoft Azure Blob Storage, Microsoft Azure Data Lake Storage, and OpenStack Swift Object Storage. You can load vector and raster data from these services into QGIS. Set up a new *Cloud* connection in the *Browser* panel by right-clicking on the *Cloud* entry and selecting *New Connection*. You will see a drop-down list of available cloud services. Select the service you want to connect to and fill in the required fields:

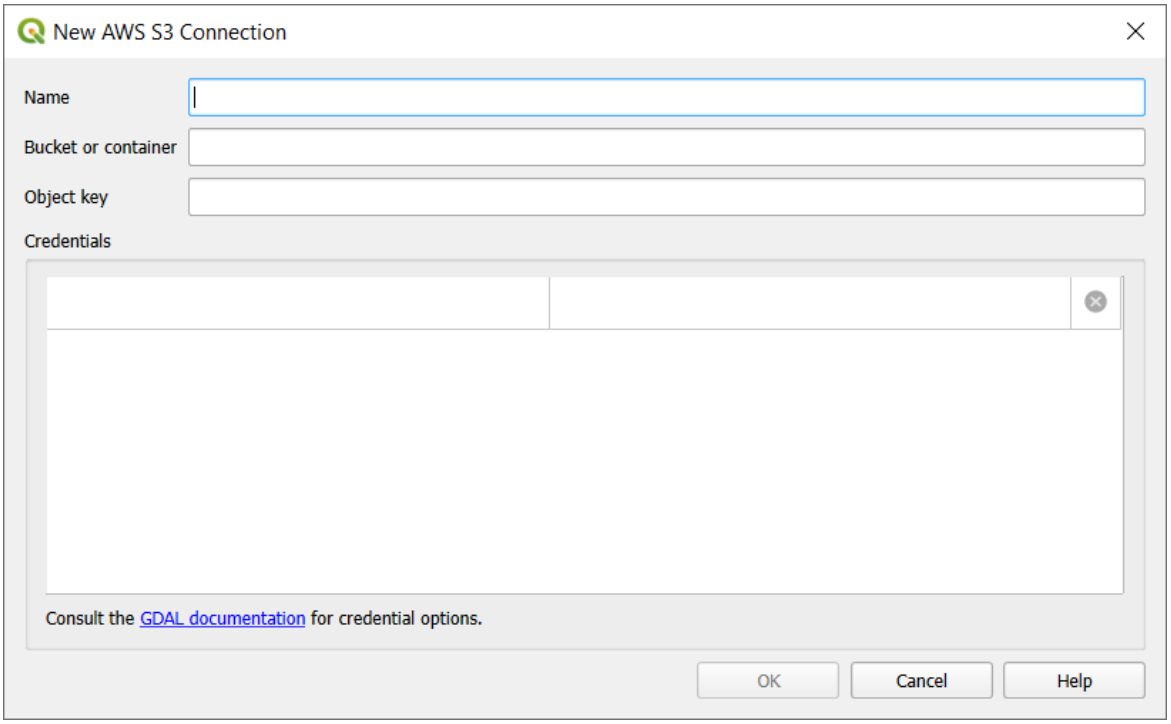


Figura11.25: Cloud Connection Dialog

- *Name*: A name for the connection.
- *Bucket or Container*: The name of the bucket or container in the cloud service.
- *Object Key* (optional): The key of the object in the bucket or container.
- *Credentials*: The credentials to access the cloud service.

You can also choose to *Save Connection* to an XML file or *Load Connection* from an XML file.

11.2 Criando Camadas

As camadas podem ser criadas de várias maneiras, incluindo:

- empty layers from scratch
- camadas de camadas existentes
- layers from the clipboard
- layers as a result of an SQL-like query based on one or many layers (*virtual layers*)

QGIS also provides tools to import/export from/to different formats.

11.2.1 Criando novas camadas vetoriais

QGIS allows you to create new layers in different formats. It provides tools for creating GeoPackage, Shapefile, SpatiaLite, GPX format and Temporary Scratch layers (aka memory layers). Creation of a *new GRASS layer* is supported within the GRASS plugin.

Criando uma nova camada GeoPackage

To create a new GeoPackage layer, press the *New GeoPackage Layer…* button in the *Layer* ► *Create Layer* ► menu or from the *Data Source Manager* toolbar. You can also create a new GeoPackage layer through the *Browser Panel* by selecting the *Create Database and Layer…*. The *New GeoPackage Layer* dialog will be displayed as shown in Fig.11.26.

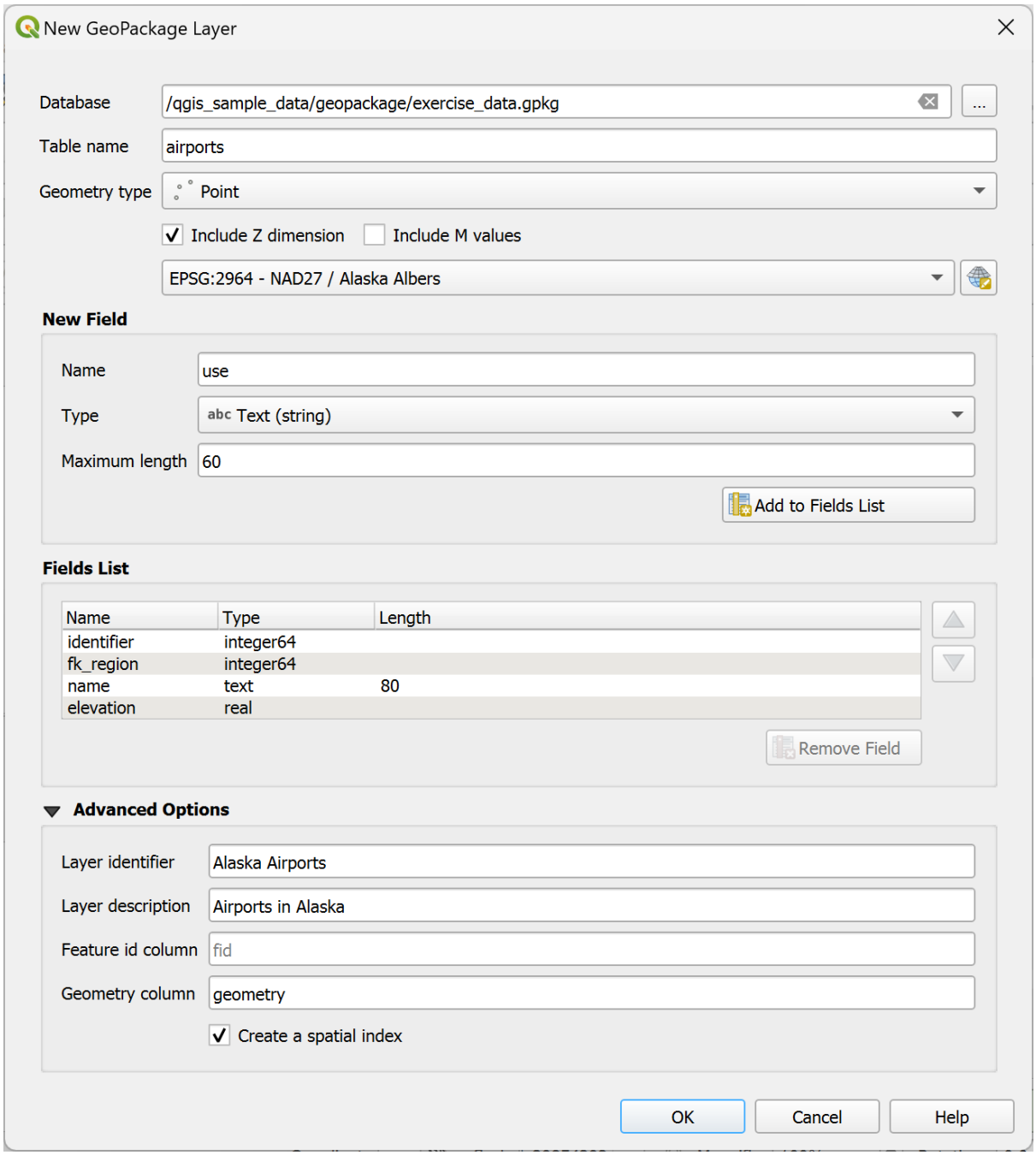


Figura11.26: Creating a New GeoPackage layer dialog

1. The first step is to indicate the database file location. This can be done by pressing the … button to the right of the *Database* field and select an existing GeoPackage file or create a new one. QGIS will automatically add the right extension to the name you provide.
2. Give the new layer / table a name (*Table name*)
3. Define the *Geometry type*. If not a geometryless layer, you can specify whether it should *Include Z dimension* and/or *Include M values*.
4. Specify the coordinate reference system using the button

Para adicionar campos à camada que você está criando:

1. Enter the *Name* of the field
2. Select the data *Type*. Supported types are *Text data*, *Whole number* (both integer and integer64), *Decimal number*, *Date* and *Date and time*, *Binary (BLOB)* and *Boolean*.
3. Depending on the selected data format, enter the *Maximum length* of values.
4. Click on the *Add to Fields List* button
5. Reproduzir as etapas acima para cada campo que você precisa adicionar
6. You can later change the fields order using the Move Up and Move Down buttons
7. Once you are happy with the attributes, click *OK*. QGIS will add the new layer to the legend, and you can edit it as described in section *Digitalizar uma camada existente*.

By default, when creating a GeoPackage layer, QGIS generates a *Feature id column* called `fid` which acts as the primary key of the layer. The name can be changed. The geometry field, if availabe, is named `geometry`, and you can choose to *Create a spatial index* on it. These options can be found under the *Advanced Options* together with the *Layer identifier* (short human readable name of the layer) and the *Layer description*.

Further management of GeoPackage layers can be done with the *DB Manager*.

Criando uma nova camada Shapefile

To create a new ESRI Shapefile format layer, press the *New Shapefile Layer…* button in the *Layer* ► *Create Layer* ► menu or from the *Data Source Manager* toolbar. The *New Shapefile Layer* dialog will be displayed as shown in Fig.11.27.

1. Provide a path and file name using the … button next to *File name*. QGIS will automatically add the right extension to the name you provide.
2. Next, indicate the *File encoding* of the data
3. Choose the *Geometry type* of the layer: No Geometry (resulting in a `.DBF` format file), point, multipoint, line or polygon
4. Specify whether the geometry should have additional dimensions: *None*, *Z (+ M values)* or *M values*
5. Specify the coordinate reference system using the button

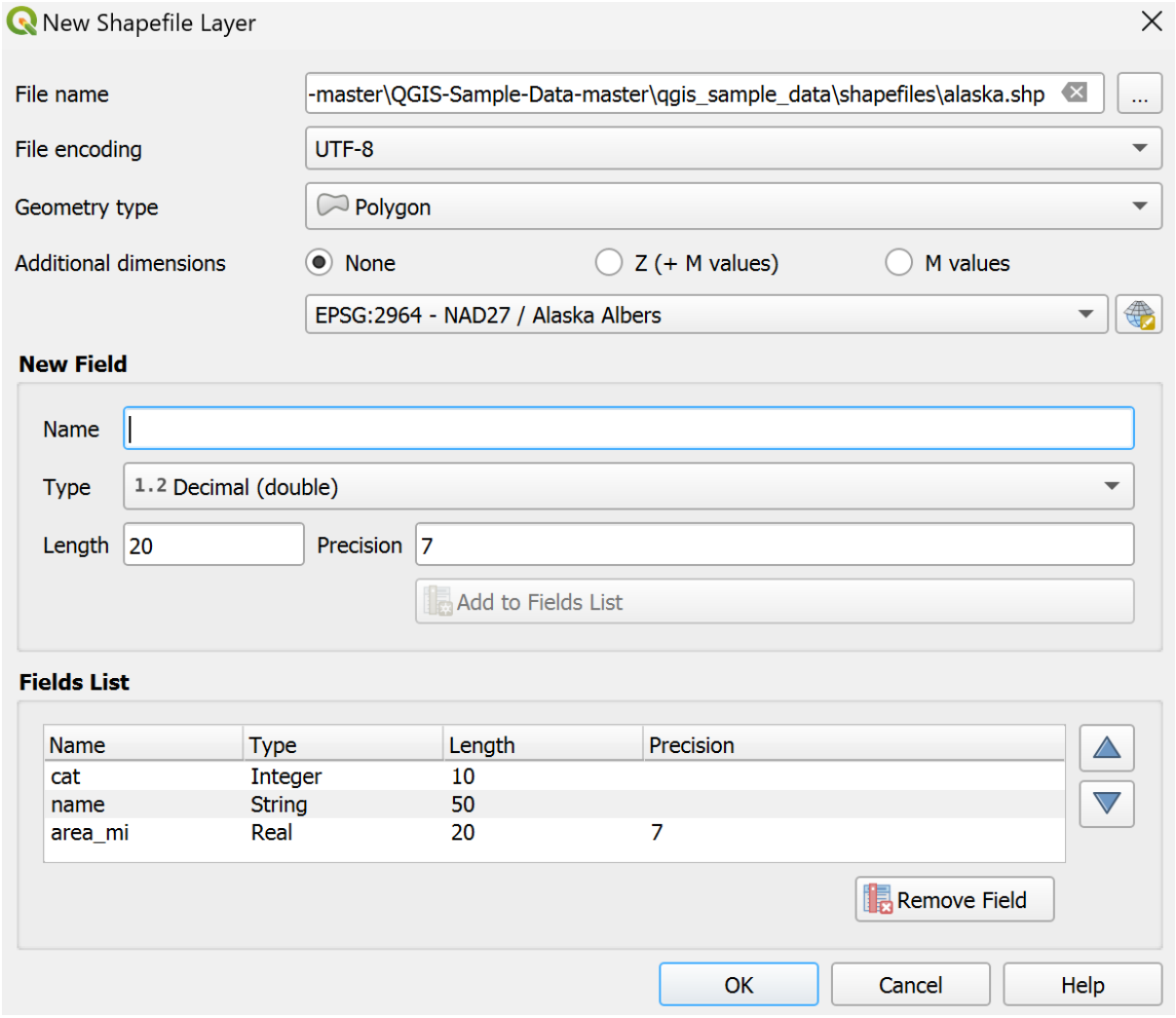


Figura11.27: Creating a new Shapefile layer dialog

Para adicionar campos à camada que você está criando:

1. Enter the *Name* of the field
2. Select the data *Type*. Only *Decimal number*, *Whole number*, *Text data*, *Date*, and *Boolean* data types attributes are supported.
3. Depending on the selected data format, enter the *Length* and *Precision*.
4. Click on the *Add to Fields List* button
5. Reproduzir as etapas acima para cada campo que você precisa adicionar
6. You can later change the fields order using the Move Up and Move Down buttons
7. Once you are happy with the attributes, click *OK*. QGIS will add the new layer to the legend, and you can edit it as described in section *Digitalizar uma camada existente*.

Por padrão, uma primeira coluna inteira `id` é adicionada, mas pode ser removida.

Creating a new SpatiaLite layer

To create a new SpatiaLite layer, press the *New SpatiaLite Layer…* button in the *Layer* ► *Create Layer* ► menu or from the *Data Source Manager* toolbar. The *New SpatiaLite Layer* dialog will be displayed as shown in Fig.11.28.

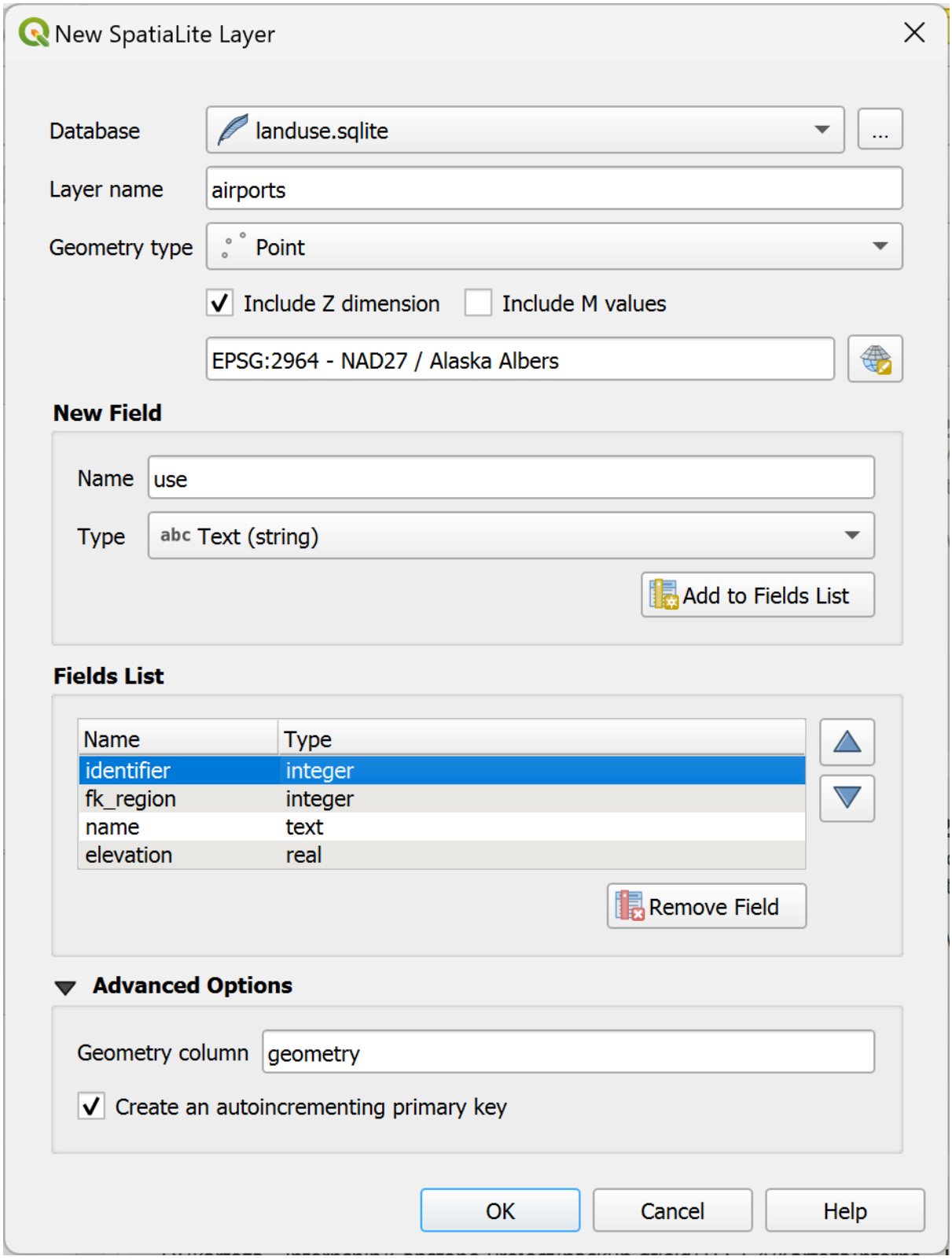


Figura11.28: Creating a New SpatiaLite layer dialog

1. The first step is to indicate the database file location. This can be done by pressing the … button to the right of the *Database* field and select an existing SpatiaLite file or create a new one. QGIS will automatically add the right extension to the name you provide.
2. Provide a name (*Layer name*) for the new layer
3. Define the *Geometry type*. If not a geometryless layer, you can specify whether it should *Include Z dimension* and/or *Include M values*.
4. Specify the coordinate reference system using the button.

Para adicionar campos à camada que você está criando:

1. Enter the *Name* of the field
2. Select the data *Type*. Supported types are *Text data*, *Whole number*, *Decimal number*, *Date* and *Date time*.
3. Click on the *Add to Fields List* button
4. Reproduzir as etapas acima para cada campo que você precisa adicionar
5. You can later change the fields order using the Move Up and Move Down buttons
6. Once you are happy with the attributes, click *OK*. QGIS will add the new layer to the legend, and you can edit it as described in section *Digitalizar uma camada existente*.

If desired, you can select *Create an autoincrementing primary key* under the *Advanced Options* section. You can also rename the *Geometry column* (`geometry` by default).

Further management of SpatiaLite layers can be done with *DB Manager*.

Criando uma nova camada de malha

To create a new Mesh layer, press the *New Mesh Layer…* button in the *Layer* ► *Create Layer* ► menu or from the *Data Source Manager* toolbar. The *New Mesh Layer* dialog will be displayed as shown in Fig.11.29.

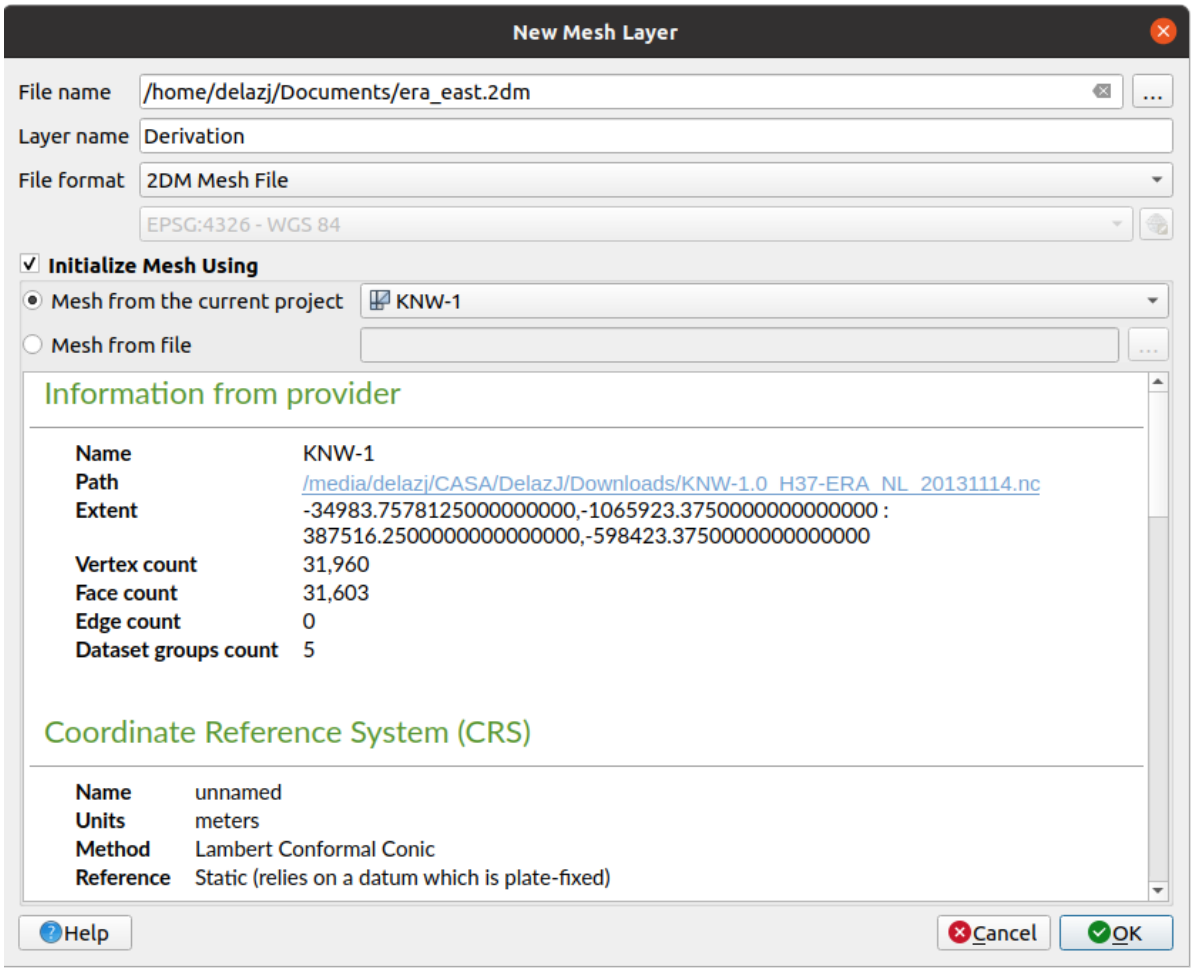


Figura11.29: Creating a New Mesh layer dialog

1. The first step is to indicate the mesh file location. This can be done by pressing the … button to the right of the *File name* field and select an existing mesh file or create a new one.
2. Provide a name (*Layer name*), i.e. the name the layer is displayed with in the *Layers* panel
3. Select the *File format*: currently supported mesh file formats are `2DM Mesh File (*.2dm)`, `Selafin File (*.slf)` and `UGRID (*.nc)`.
4. Indicate the *Coordinate Reference System* to assign to the dataset
5. The above steps will generate an empty layer that you can afterwards digitize vertices and add dataset groups to. It's however also possible to initialize the layer with an existing mesh layer, i.e. populate the new layer with vertices or faces from the other. To do so:

   1. Check *Initialize Mesh using*
   2. and select either a *Mesh from the current project* or *Mesh from a file*. Informations on the selected mesh file are displayed for checkup.

   Note that only the frame of the mesh layer is transferred to the new layer; their datasets are not copied.

Criando uma nova camada GPX

Para criar um novo arquivo GPX:

1. Select *Create Layer* ► *New GPX Layer…* from the *Layer* menu.
2. In the dialog, choose where to save the new file, name it and press *Save*.
3. Three new layers are added to the *Layers Panel*:
   - a point layer to digitize locations (`waypoints`) with fields storing the name, elevation, comment, description, source, url and url name
   - a line layer to digitize sequences of locations that make up a planned route (`routes`) with fields storing the name, symbol, number, comment, description, source, url, url name
   - and a line layer to track the receiver's movement over time (`tracks`) with fields storing the name, symbol, number, comment, description, source, url, url name.
4. You can now edit any of them as described in section *Digitalizar uma camada existente*.

Creating a new Temporary Scratch Layer

Temporary Scratch Layers are in-memory layers, meaning that they are not saved on disk and will be discarded when QGIS is closed. They can be handy for storing features you temporarily need or as intermediate layers during geoprocessing operations.

To create a new Temporary Scratch layer, choose the *New Temporary Scratch Layer…* entry in the *Layer* ► *Create Layer* ► menu or in the *Data Source Manager* toolbar. The *New Temporary Scratch Layer* dialog will be displayed as shown in Fig.11.30. Then:

1. Provide the *Layer name*
2. Select the *Geometry type*. Here you can create a:
   - `No geometry` type layer, served as simple table,
   - Camada de``Ponto`` ou `MultiPonto`,
   - `LineString/CompoundCurve` or `MultiLineString/MultiCurve` layer,
   - `Polygon/CurvePolygon` or `MultiPolygon/MultiSurface` layer.
3. For geometric types, specify the dimensions of the dataset: check whether it should *Include Z dimension* and/or *Include M values*
4. Specify the coordinate reference system using the button.
5. Adicionar campos à camada. Observe que, ao contrário de muitos formatos, as camadas temporárias podem ser criadas sem nenhum campo. Esta etapa é, portanto, opcional.
   1. Enter the *Name* of the field
   2. Select the data *Type*: *Text*, *Whole number*, *Decimal number*, *Boolean*, *Date*, *Time*, *Date & Time* and *Binary (BLOB)* are supported.
   3. Depending on the selected data format, enter the *Length* and *Precision*
   4. Click on the *Add to Fields List* button
   5. Repita os passos acima para cada campo que você precisa adicionar
   6. You can later change the fields order using the Move Up and Move Down buttons
6. Once you are happy with the settings, click *OK*. QGIS will add the new layer to the *Layers* panel, and you can edit it as described in section *Digitalizar uma camada existente*.

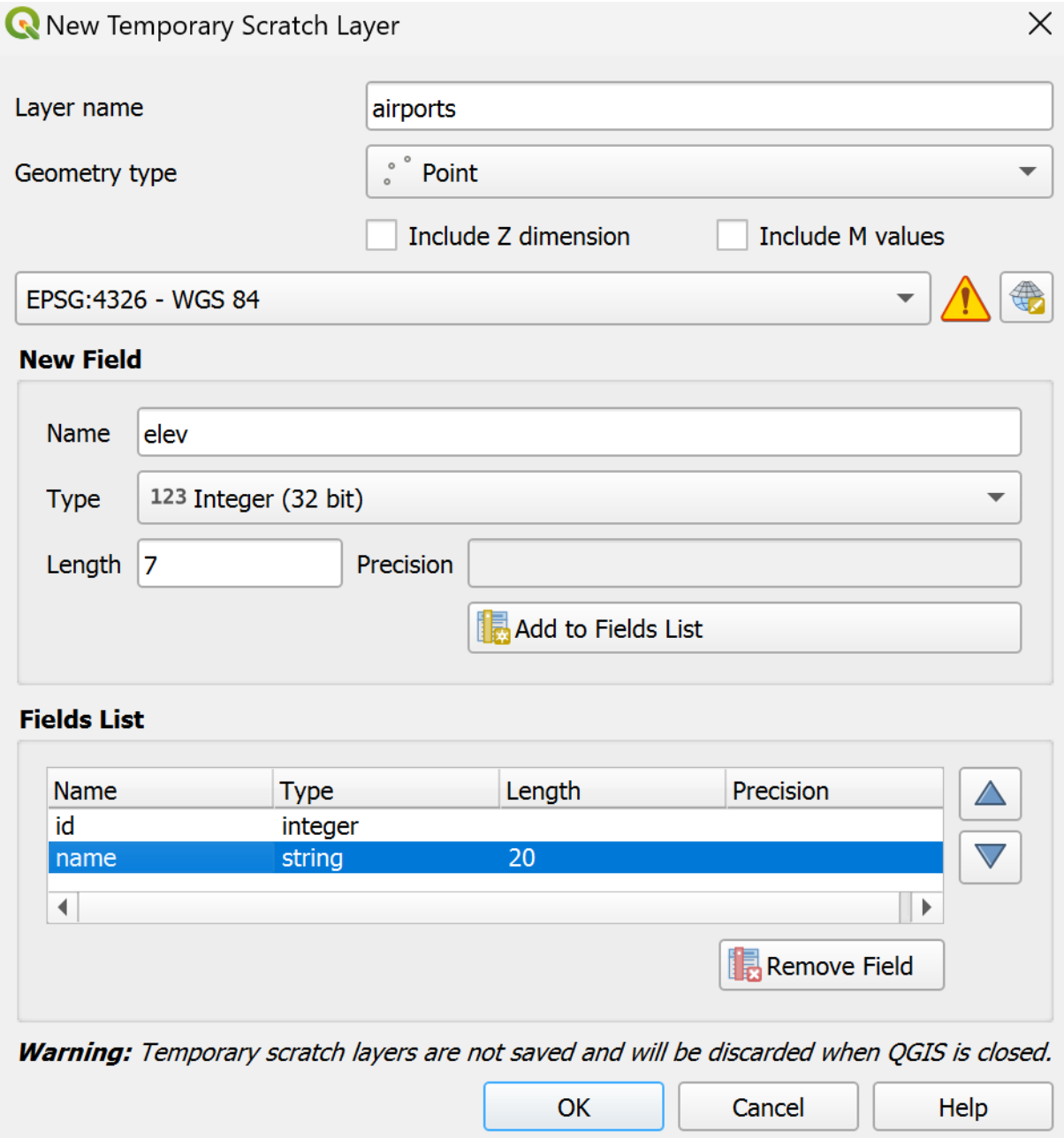


Figura11.30: Creating a new Temporary Scratch layer dialog

You can also create prepopulated temporary scratch layers using e.g. the clipboard (see *Creating new layers from the clipboard*) or as a result of a *Processing algorithm*.

Dica

Armazene permanentemente a camada de memória no disco

To avoid data loss when closing a project with temporary scratch layers, you can save these layers to any vector format supported by QGIS:

- clicking the indicator icon next to the layer;
- selecting the *Make permanent* entry in the layer contextual menu;
- using the *Export* ► entry from the contextual menu or the *Layer* ► *Save As...* menu.

Each of these commands opens the *Save Vector Layer as* dialog described in the *Criando novas camadas a partir de uma camada existente* section and the saved file replaces the temporary one in the *Layers* panel.

11.2.2 Criando novas camadas a partir de uma camada existente

Layers (raster, vector and point cloud) can be saved in a different format and/or reprojected to a different coordinate reference system (CRS) using the *Layer* ► *Save As…* menu or right-clicking on the layer in the *Layers panel* and selecting:

- *Export* ► *Save As…* for raster and point cloud layers
- *Export* ► *Save Features As…* or *Export* ► *Save Selected Features As…* for vector layers.
- Drag and drop the layer from the layer tree to the PostgreSQL entry in the *Browser Panel*. Note that you must have a PostgreSQL connection in the *Browser Panel*.

Parâmetros comuns

The *Save Layer as…* dialog shows several parameters to change the behavior when saving the layer. Among the common parameters for raster and vector are:

- *File name*: the location of the file on the disk. It can refer to the output layer or to a container that stores the layer (for example database-like formats such as GeoPackage, SpatiaLite or Open Document Spreadsheets).
- *CRS*: can be changed to reproject the data
- *Extent*: restricts the extent of the input that is to be exported using the *extent_selector* widget
- *Add saved file to map*: to add the new layer to the canvas

However, some parameters are specific to certain formats:

Parâmetros específicos do raster

Dependendo do formato de exportação, algumas destas opções podem não estar disponíveis:

- *Output mode* (it can be **raw data** or **rendered image**)
- *Format*: exports to any raster format GDAL can write to, such as GeoTiff, GeoPackage, MBTiles, Geospatial PDF, SAGA GIS Binary Grid, Intergraph Raster, ESRI .hdr Labelled…
- *Resolution*
- *Create Options*: use advanced options (file compression, block sizes, colorimetry…) when generating files, either from the *predefined create profiles* related to the output format or by setting each parameter.
- *Pyramids* creation
- *VRT Tiles* in case you opted to *Create VRT*
- *No data values*

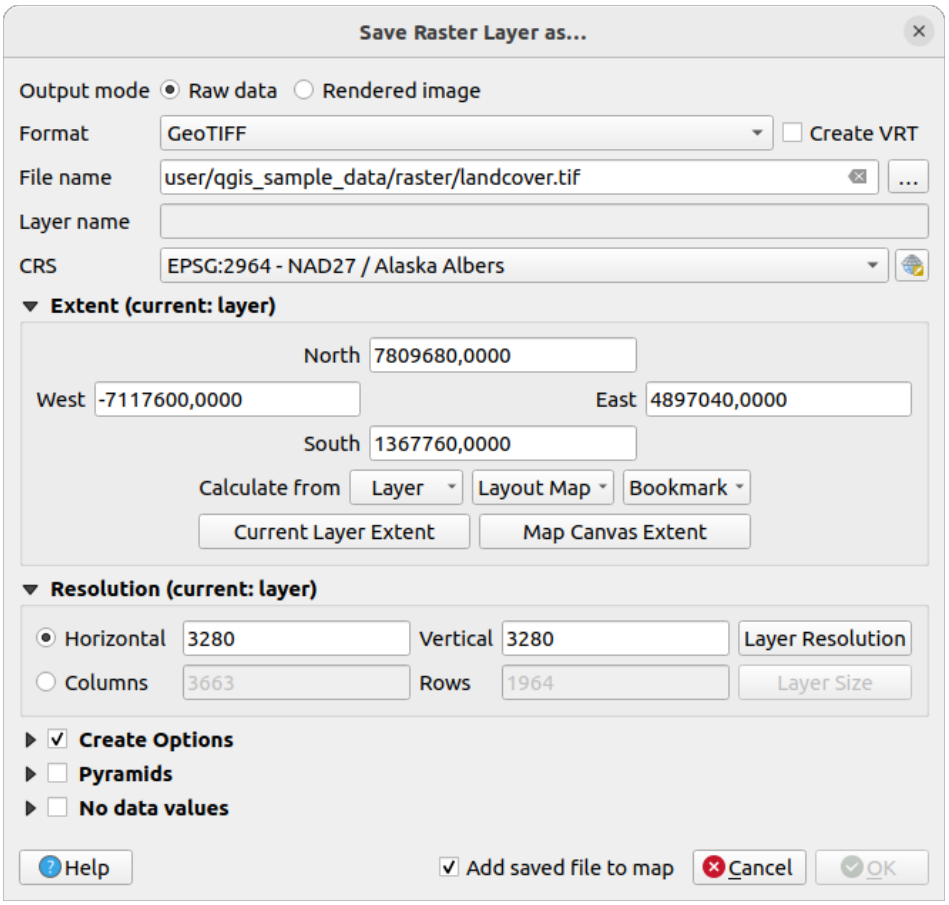


Figura11.31: Salvando como uma nova camada matricial

Parâmetros específicos do vetor

Dependendo do formato de exportação, algumas destas opções podem estar disponíveis:

- *Format*: exports to any vector format GDAL can write to, such as GeoPackage, GML, ESRI Shapefile, AutoCAD DXF, ESRI FileGDB, Mapinfo TAB or MIF, SpatiaLite, CSV, KML, ODS, …
- *Layer name*: available when the *File name* refers to a container-like format, this entry represents the output layer.
- *Encoding*
- *Save only selected features*
- *Select fields to export and their export options*: provides means to export fields with custom names and *form widget* settings:
  - Check rows under the *Name* column to choose fields to keep in the output layer, or press *Select All* or *Deselect All* buttons
  - Toggle the *Use aliases for exported name* checkbox to populate the *Export name* column with corresponding field aliases or reset to the original field name. Double-clicking a cell will also edit the name.
  - Depending on whether attribute form custom widgets are in use, you can *Replace all selected raw field values by displayed values*. E.g. if a `value map` widget is applied to a field, the output layer will contain the description values instead of the original values. The replacement can also be done on a field by field basis, in the *Replace with displayed values* column.
- *Persist layer metadata*: ensures that any layer *metadata* present in the source layer will be copied and stored:
  - in the newly created layer, if the output is of GeoPackage format

  - as a `.qmd` file along with the output layer, for other formats. Note that file-based formats supporting more than one dataset (e.g. SpatiaLite, DXF,…) may have unintended behavior.

- *Symbology export*: can be used mainly for DXF export and for all file formats who manage OGR feature styles (see note below) as DXF, KML, tab file formats:
  - **No symbology**: default style of the application that reads the data
  - **Feature symbology**: save style with OGR Feature Styles (see note below)
  - **Symbol Layer symbology**: save with OGR Feature Styles (see note below) but export the same geometry multiple times if there are multiple symbology symbol layers used
  - Um valor de **Escala** pode ser aplicado às opções mais recentes

Nota

OGR Feature Styles are a way to store style directly in the data as a hidden attribute. Only some formats can handle this kind of information. KML, DXF and TAB file formats are such formats. For advanced details, you can read the OGR Feature Styles specification document.

- *Geometry*: you can configure the geometry capabilities of the output layer
  - *geometry type*: keeps the original geometry of the features when set to **Automatic**, otherwise removes or overrides it with any type. You can add an empty geometry column to an attribute table and remove the geometry column of a spatial layer.
  - *Force multi-type*: forces creation of multi-geometry features in the layer.
  - *Include z-dimension* to geometries.

Dica

Overriding layer geometry type makes it possible to do things like save a geometryless table (e.g. `.csv` file) into a shapefile WITH any type of geometry (point, line, polygon), so that geometries can then be manually added to rows with the Add Part tool.

- *Datasource Options*, *Layer Options* or *Custom Options* which allow you to configure advanced parameters depending on the output format. Some are described in *Explorando Formatos de Dados e Campos* but for full details, see the GDAL driver documentation. Each file format has its own custom parameters, e.g. for the `GeoJSON` format have a look at the GDAL GeoJSON documentation.

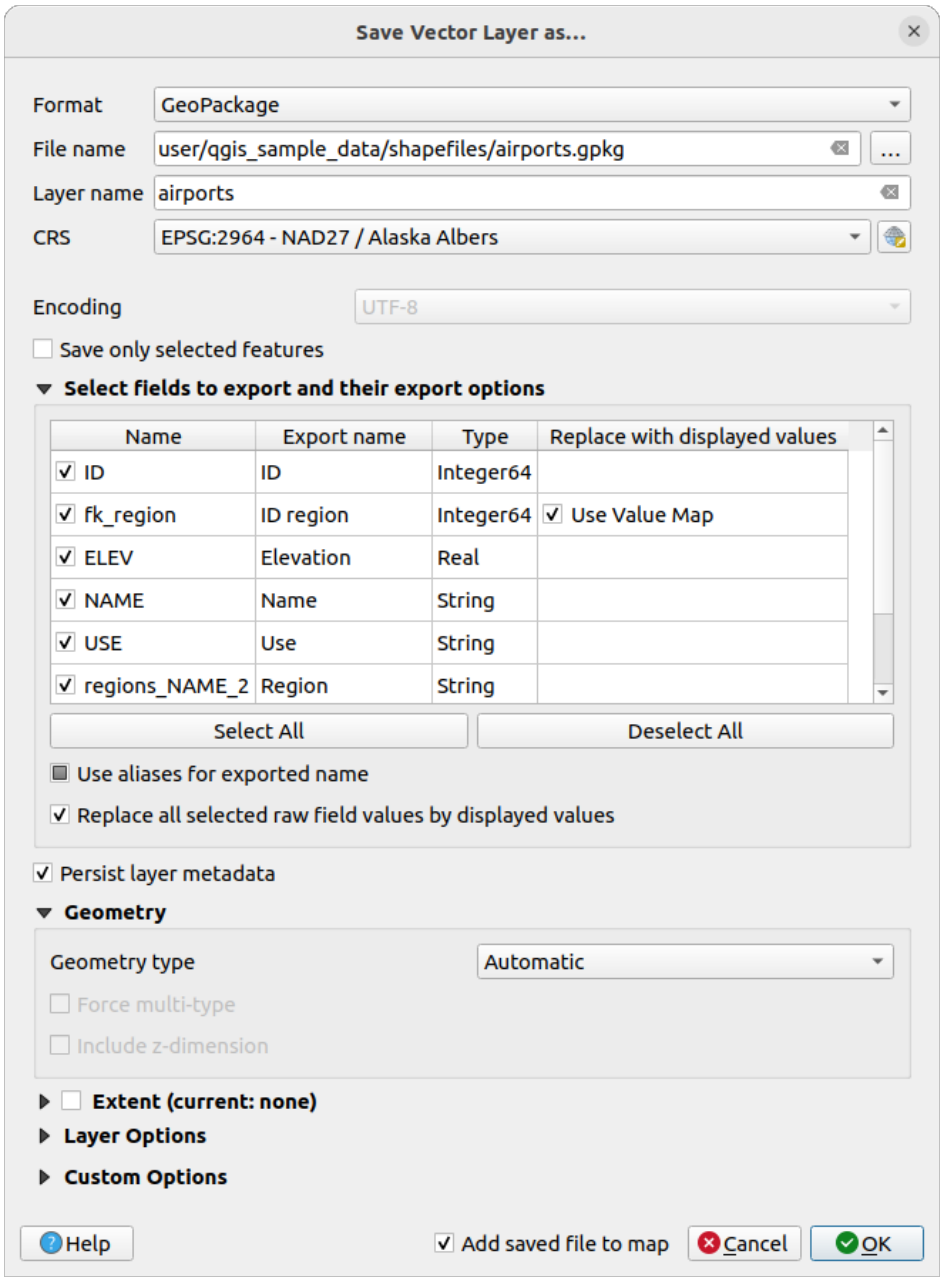


Figura11.32: Salvando como uma nova camada vetorial

When saving a vector layer into an existing file, depending on the capabilities of the output format (Geopackage, SpatiaLite, FileGDB…), the user can decide whether to:

- overwrite the whole file
- overwrite only the target layer (the layer name is configurable)
- append features to the existing target layer
- append features, add new fields if there are any.

For formats like ESRI Shapefile, MapInfo .tab, feature append is also available.

Point Cloud specific parameters

Similar to raster and vector layers, point cloud layers can be saved in a different format and/or reprojected to a different coordinate reference system (CRS). This allows you to export a point cloud layer to vector or point cloud formats. Current supported formats are: Temporary scratch (memory layer), GeoPackage, ESRI Shapefile, DXF and LAS/LAZ point cloud. In addition to the common parameters listed above, exporting point cloud layers includes the following options:

- *Filter by Polygon Layer*: Allows you to filter the point cloud data based on a polygon layer.
- *Elevation Range*: Enables filtering of the point cloud data based on a specified Z range.
- *Limit number of points*: Provides an option to limit the number of points exported from the point cloud layer.

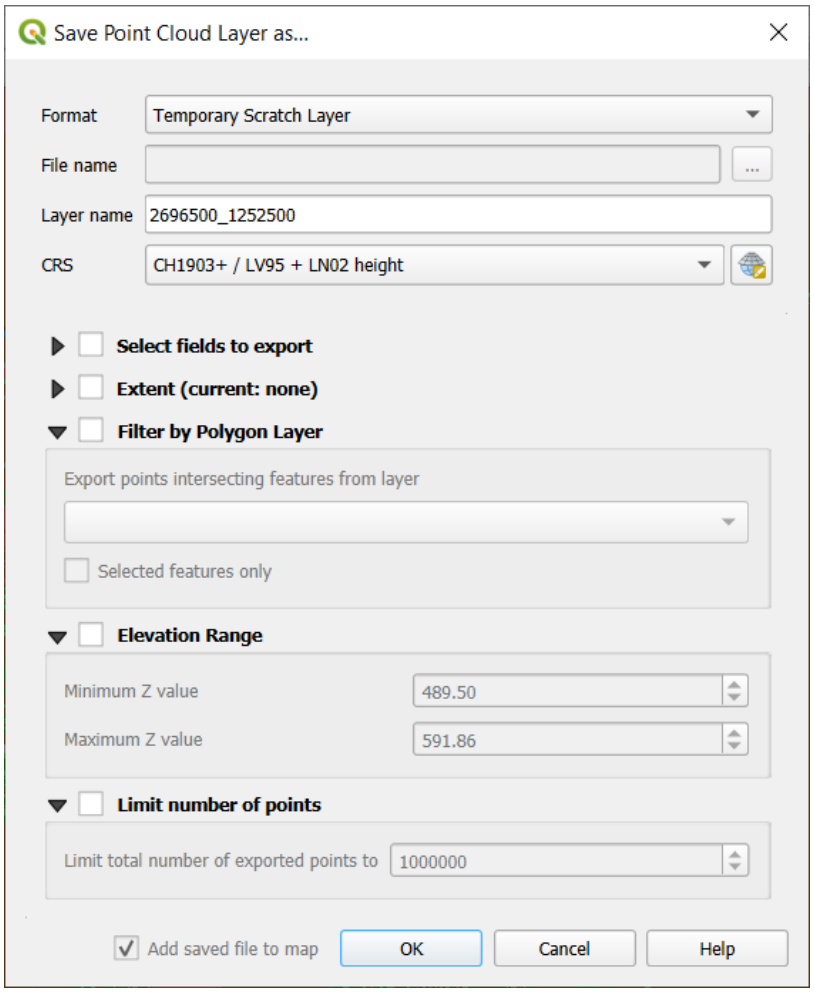


Figura11.33: Saving a point cloud layer as a new layer

11.2.3 Criando novos arquivos DXF

Besides the *Save As...* dialog which provides options to export a single layer to another format, including `*.DXF`, QGIS provides another tool to export multiple layers as a single DXF layer. It's accessible in the *Project* ► *Import/Export* ► *Export Project to DXF...* menu.

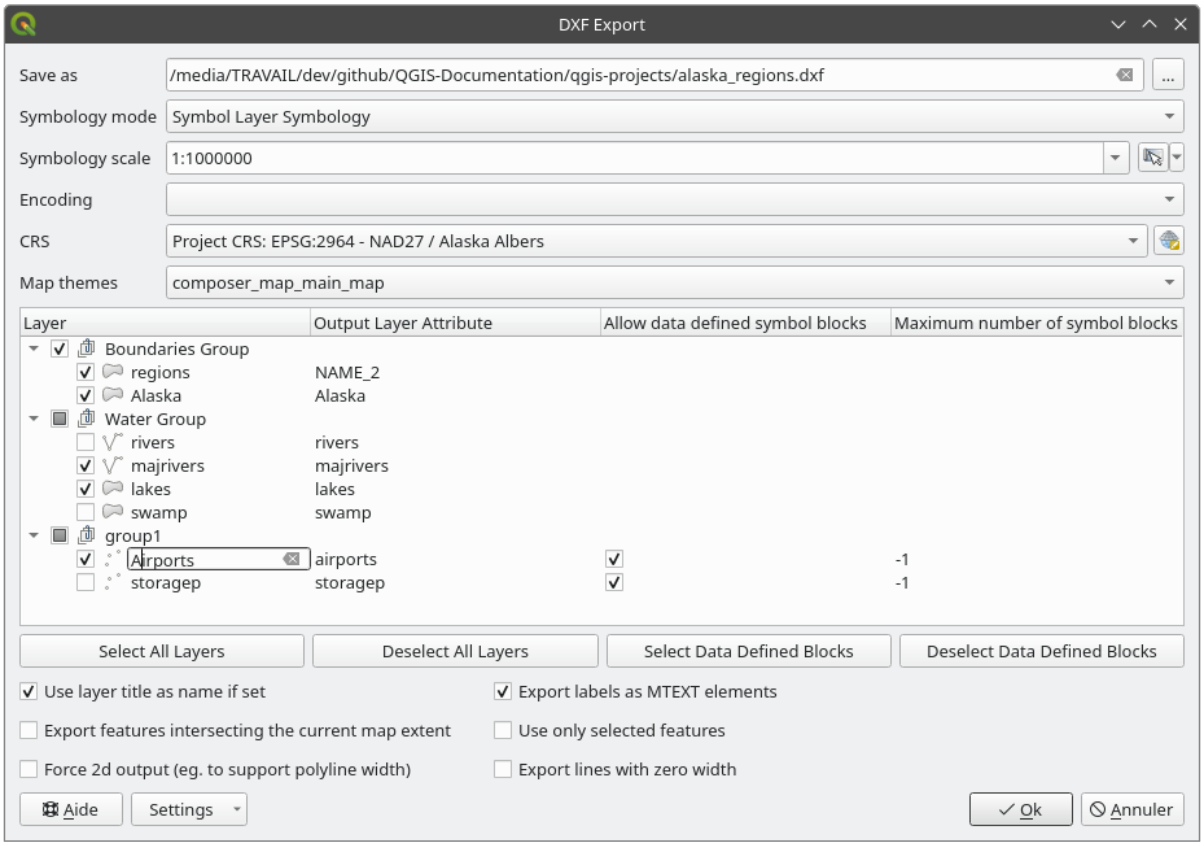


Figura11.34: Exporting a project to DXF dialog

In the *DXF Export* dialog:

1. Fornecer o arquivo de destino.
2. Choose the symbology mode and scale (see the *OGR Feature Styles* note), if applicable.
3. Select the data *Encoding*.
4. Select the *CRS* to apply: the selected layers will be reprojected to the given CRS.
5. Select the layers to include in the DXF files either by checking them in the table widget or automatically picking them from an existing *map theme*. The *Select All* and *Deselect All* buttons can help to quickly set the data to export.

   For each layer, you can:

   - Override the output layer name without altering the original project layer. For this, click on the *Layer* name in the dialog and write the output name to use.
   - *Output layer attribute*: Choose whether to export all the features in a single DXF layer or rely on a field whose values are used to split the features into layers in the DXF output. In the latter case, each layer will take its name from the corresponding field value.
   - *Allow data defined symbol blocks*:

- *Maximum number of symbol blocks*: creates symbol blocks up to the specified limit, starting with the ones containing the highest number of references. The other symbols are written as they are. `-1` means no limitation.

Opcionalmente, você também pode escolher por:

- ☐ *Use the layer title as name if set* instead of the layer name itself: the title is taken from the *metadata* or *server* properties of the layer;
- ☐ *Export features intersecting the current map extent*;
- ☐ *Force 2d output (eg. to support polyline width)*;
- ☑ *Export label as MTEXT elements* or TEXT elements;
- ☐ *Use only selected features*;
- ☐ *Export lines with zero width*: all the lines are exported with minimal width `0` (hairline) if enabled. This helps keep the lines minimal in the file regardless of the zoom level, and can be handy for doing further CAD-editing with the exported dxf, especially if there are many features next to each other on the map.

Nota

The precedence for defining the output layer name is as follows:

1. The field value from *Output layer attribute*
2. The overridden name in *Layer* column
3. The *Use the layer title as name if set* option
4. The layer name

Current settings defined in the *DXF Export* dialog may be stored in an XML file for reusing them in other sessions. For this, the *Settings* combo box has two options: *Load Settings from File…* and *Save Settings to File…*.

11.2.4 Creating new layers from the clipboard

As feições que estão na área de transferência podem ser colados em uma nova camada. Para fazer isso, selecione algumas feições, copie-as para a área de transferência e cole-as em uma nova camada usando *Editar* ► *Colar Feições como* ► e escolhendo:

- *New Vector Layer…*: the *Save vector layer as…* dialog appears (see *Criando novas camadas a partir de uma camada existente* for parameters)
- or *Temporary Scratch Layer…*: you need to provide a name for the layer

A new layer, filled with selected features and their attributes is created (and added to map canvas).

Nota

Creating layers from the clipboard is possible with features selected and copied within QGIS as well as features from another application, as long as their geometries are defined using well-known text (WKT).

11.2.5 Creating SQL Query Layers

Beside loading an entire layer in a project or creating new layers from scratch or pasted features, you can also load layers generated on the fly from other layer(s). They are the result of a more or less advanced filter using SQL language, applied to stored layers regardless of their data provider or their availability in the active project. This means that temporary layers are not compatible with this feature. Depending on the provider, one or more layers can be used to write the query. The generated layer remains dependent on the layer(s) involved in the query and is loaded with the Filter icon next to it.

This feature is accessible:

- From the *Browser* panel, right-click on a supported data (plain layer, database connection, schema or table) and select *Execute SQL…* entry in the contextual menu.
- From the *Layers* panel, select a loaded layer, right-click and select *Execute SQL…* entry in the contextual menu.

This opens a window with a central text box widget where you can write SQL queries.

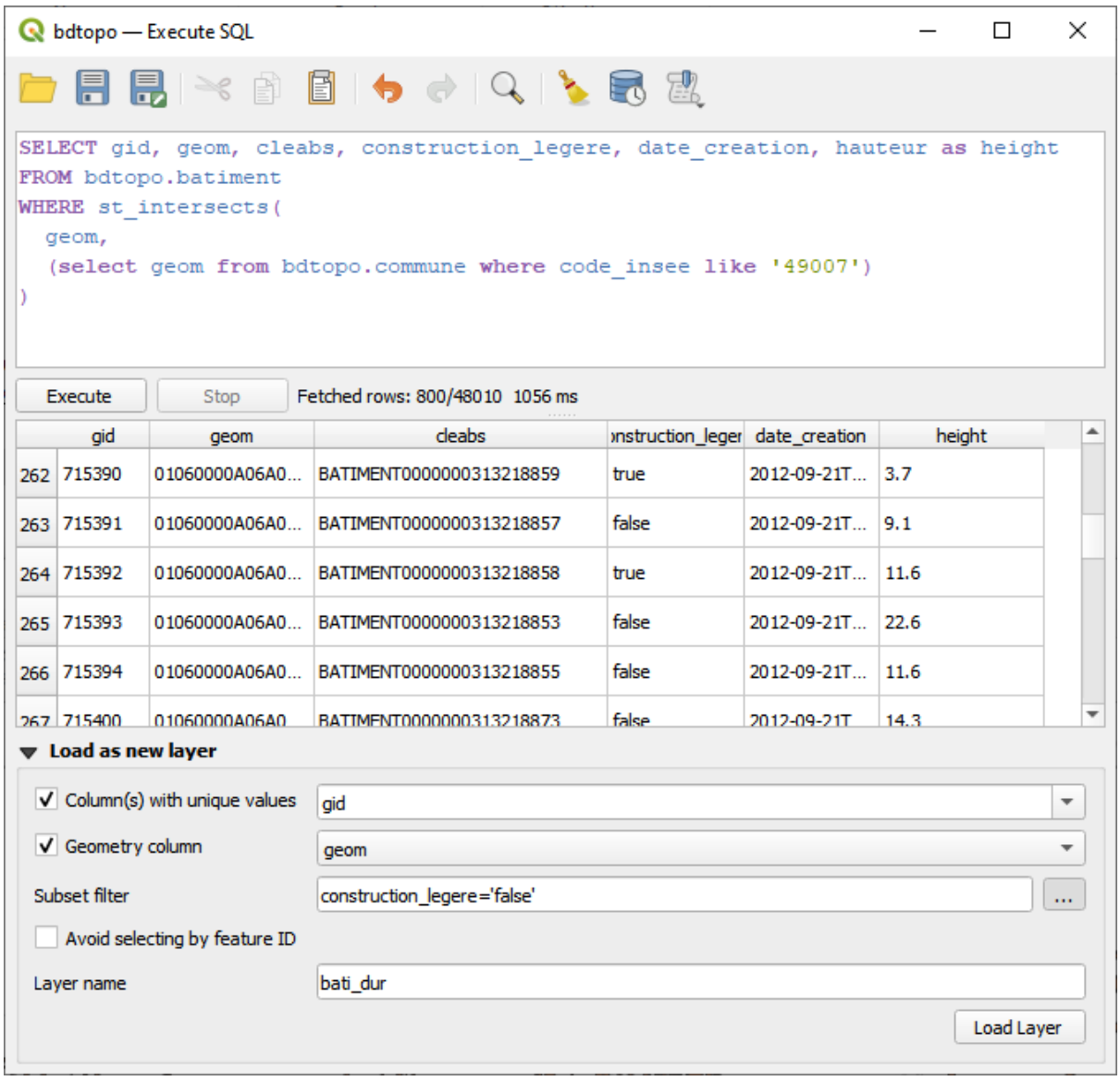


Figura11.35: Executing SQL queries in the Execute SQL window

At the top of the dialog, a toolbar provides a set of tools to create, store and manipulate your queries:

- Open Queries…: fills the text editor widget with contents from an existing `.sql` file
- Save Queries… and Save Queries as… help you store the written query to a `.sql` file
- Statements of the query can be adjusted using the Cut, Copy and Paste buttons. Likewise, you can Undo or Redo your changes.
- The Find & Replace enables, at the bottom of the dialog, a widget allowing to look for a particular string in your SQL code. The search can be case sensitive, affect partial or whole word, rely on a regular expression. It is then possible to navigate through the found strings, replacing them one by one or all in a row.
- Use Clear SQL Editor to wipe the text editor.
- The History button opens a dialog storing previously run queries. More at *Query History*.
- As previously mentioned, queries can be saved as an `.sql` file stored on disk. Using the Store Current Query button, they can also be stored:
  - In the active *User Profile*, in the associated `QGIS3.ini` file, thus accessible in subsequent projects
  - or as part of the *Current Project*.

  Clicking an entry from the stored queries drop-down menu inserts that query in the expression being written. The stored entry can also be deleted, from the drop-down menu.

Running and loading queries as layer

In the central part of the *Execute SQL* dialog, you build your query using the SQL syntax supported by the underlying provider (e.g., OGR, GeoPackage, PostgreSQL).

By default, if opened from a layer entry, a sample SQL query is provided. Editing tools to select, cut, copy, paste, undo and redo are available as well from the contextual menu.

> **Dica**
>
> **Finding the right SQL syntax for your dataset**
>
> To find the appropriate SQL dialect for your dataset, press the … button next to the *Subset filter* option in the bottom part of the dialog.

When ready, pressing the *Execute* button below the text area will run the query. It is possible to highlight a portion of the SQL to only execute that portion when pressing `Ctrl+R` or clicking the *Execute selection* button. Use the *Stop* button to abort the execution.

A successful query execution will display a table at the bottom of the dialog with returned features. You can select specific cells in the result set. Use the `Ctrl+C` shortcut to copy the selected cells to the clipboard. The copied data is available as a formatted table. This allows you to paste the data into other applications, such as spreadsheet where it will show up as a table.

The returned table can be loaded in QGIS expanding the *Load as new layer* group and configuring parameters (their availability depends on the layer provider):

- *Column(s) with unique values* to indicate primary key of the data,
- *Geometry column*: check the box to load the layer as a spatial one, and indicate the geometry field name
- *Subset filter*: allows to filter the results using a `WHERE` clause. It can be written in the text box, or built using the *Query builder* after you pressed …. Make sure to use fields that are available in the SQL layer.
- *Avoid selecting by feature ID*

- *Layer name* in the project

At any time, in the *Layers panel*, you can adjust the output layer by right-clicking and selecting *Update SQL expression…*. The *Update SQL* dialog opens, prefilled with the applied query that you can edit as you wish. Once ready, press *Update layer* and the layer will be modified in-place.

Query History

The *Query History* dialog displays all the previously run queries, sorted by date and data provider type. It is accessible from the *Database* ► *Query History…* menu, or pressing the History button from the *Execute SQL* dialog.

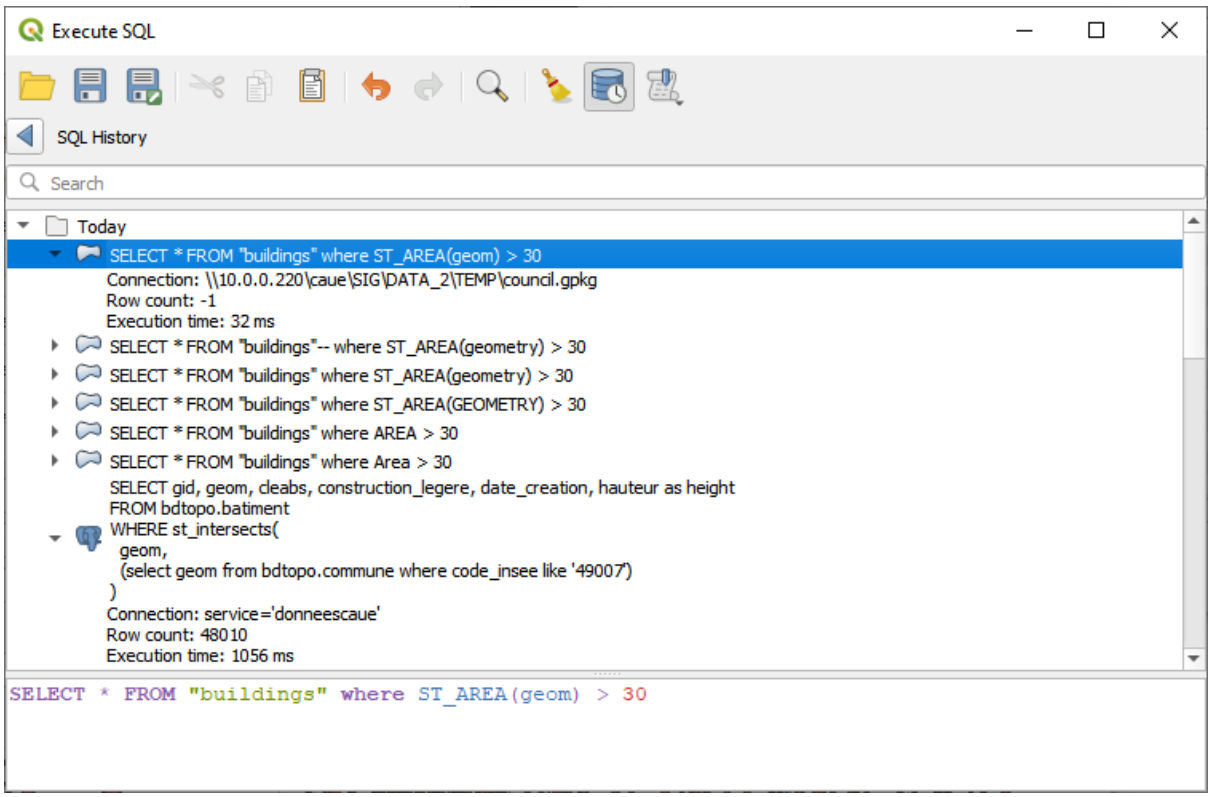


Figura11.36: History of the executed SQL queries

At the top of the dialog, a search box can be used to filter out the queries. The search is done through every strings displayed in the query history. If the dialog is opened from within the *Execute SQL* dialog, the search box is preceded by the *Execute SQL toolbar*.

Below, collapsible and sorted date entries list their related queries. Each query is identifiable by its SQL command, and when expanded, reveals:

- The *Connection*: the source of the data the query is based on
- The *Row count*: number of returned features
- The *Execution time*

Hover over an entry and a tooltip displays the full query. Likewise, selecting an entry also displayed it in the bottom part of the dialog. You can interact with the text, copying all or part of it.

Double-click the SQL command entry and the *Execute SQL* dialog opens, filled with the clicked query. Any query being written is overwritten. Right-click the SQL command entry and you can either:

- *Load SQL Command…*, loads the target command into the *Execute SQL* dialog, replacing any existing query. It is the same as double clicking the entry.

- *Copy SQL Command* and paste it wherever you want.

11.2.6 Criando camadas virtuais

A virtual layer is a special kind of vector layer. It allows you to define a layer as the result of an SQL query involving any number of other vector layers that QGIS is able to open. Virtual layers do not carry data by themselves and can be seen as views.

Para criar uma camada virtual, abra a caixa de diálogo de criação de camada virtual:

- choosing the *Add/Edit Virtual Layer* entry in the *Layer* ► *Add Layer* ► menu;
- enabling the *Add Virtual Layer* tab in the *Data Source Manager* dialog;
- or using the *DB Manager* dialog tree.

The dialog allows you to specify a *Layer name* and an SQL *Query*. The query can use the name (or id) of loaded vector layers as tables, as well as their field names as columns.

For example, if you have a layer called `airports`, you can create a new virtual layer called `public_airports` with an SQL query like:

```
SELECT *
FROM airports
WHERE USE = "Civilian/Public"
```

The SQL query will be executed, regardless of the underlying provider of the `airports` layer, even if this provider does not directly support SQL queries.

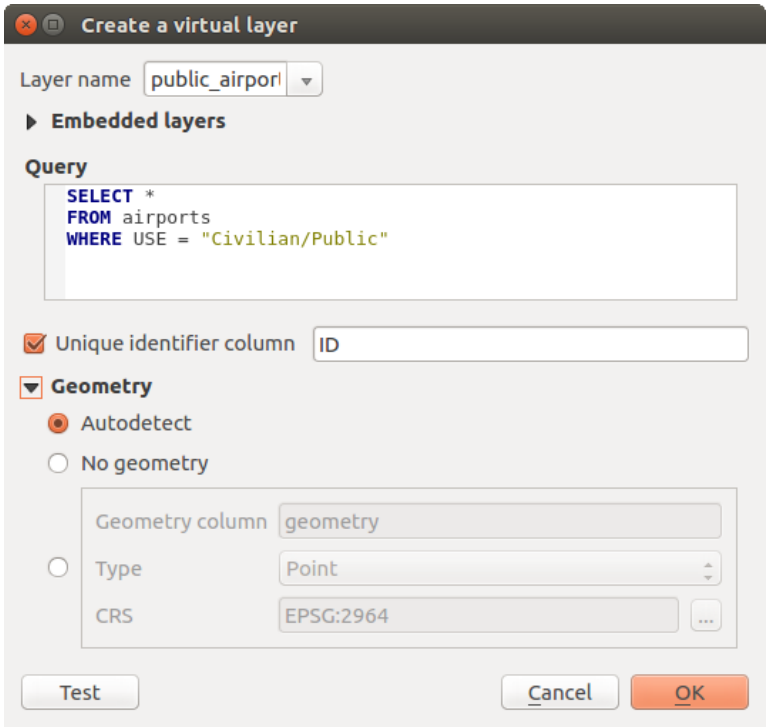


Figura11.37: Criar caixa de diálogo de camadas virtuais

Uniões e consultas complexas também podem ser criadas, por exemplo, para unir informações de aeroportos e países:

```
SELECT airports.*, country.population
FROM airports
JOIN country
ON airports.country = country.name
```

Nota

It's also possible to create virtual layers using the SQL window of *Plugin do gerenciador de banco de dados*.

Embedding layers for use in queries

Besides the vector layers available in the map canvas, the user can add layers to the *Embedded layers* list, which can be used in queries without the need to have them showing in the map canvas or Layers panel.

To embed a layer, click *Add* and provide the *Local name*, *Provider*, *Encoding* and the path to the *Source*.

The *Import* button allows adding layers in the map canvas into the Embedded layers list. Those layers can then be removed from the Layers panel without breaking existent queries.

Supported query language

The underlying engine uses SQLite and SpatiaLite to operate.

It means you can use all of the SQL your local installation of SQLite understands.

Functions from SQLite and spatial functions from SpatiaLite can also be used in a virtual layer query. For instance, creating a point layer out of an attribute-only layer can be done with a query similar to:

```
SELECT id, MakePoint(x, y, 4326) as geometry
FROM coordinates
```

Functions of QGIS expressions can also be used in a virtual layer query.

To refer the geometry column of a layer, use the name `geometry`.

Contrary to a pure SQL query, all the fields of a virtual layer query must be named. Don't forget to use the `as` keyword to name your columns if they are the result of a computation or a function call.

Performance issues

With default parameters, the virtual layer engine will try its best to detect the type of the different columns of the query, including the type of the geometry column if one is present.

This is done by introspecting the query when possible or by fetching the first row of the query (LIMIT 1) as a last resort. Fetching the first row of the result just to create the layer may be undesirable for performance reasons.

The creation dialog parameters:

- *Unique identifier column*: specifies a field of the query that represents unique integer values that QGIS can use as row identifiers. By default, an autoincrementing integer value is used. Defining a unique identifier column speeds up the selection of rows by id.
- *No geometry*: forces the virtual layer to ignore any geometry field. The resulting layer is an attribute-only layer.
- Geometry *Column*: specifies the name of the geometry column.
- Geometry *Type*: specifies the type of the geometry.
- Geometry *CRS*: specifies the coordinate reference system of the virtual layer.

Comentários especiais

The virtual layer engine tries to determine the type of each column of the query. If it fails, the first row of the query is fetched to determine column types.

The type of a particular column can be specified directly in the query by using some special comments.

The syntax is the following: `/*:type*/`. It has to be placed just after the name of a column. `type` can be either `int` for integers, `real` for floating point numbers or `text`.

Por exemplo:

```
SELECT id+1 as nid /*:int*/
FROM table
```

The type and coordinate reference system of the geometry column can also be set thanks to special comments with the following syntax `/*:gtype:srid*/` where `gtype` is the geometry type (`point`, `linestring`, `polygon`, `multipoint`, `multilinestring` or `multipolygon`) and `srid` an integer representing the EPSG code of a coordinate reference system.

Uso de índices

Ao solicitar uma camada por meio de uma camada virtual, os índices da camada de origem serão utilizados das seguintes formas:

- if an = predicate is used on the primary key column of the layer, the underlying data provider will be asked for a particular id (FilterFid)
- for any other predicates (>, <=, !=, etc.) or on a column without a primary key, a request built from an expression will be used to request the underlying vector data provider. It means indexes may be used on database providers if they exist.

A specific syntax exists to handle spatial predicates in requests and triggers the use of a spatial index: a hidden column named `_search_frame_` exists for each virtual layer. This column can be compared for equality to a bounding box. Example:

```
SELECT *
FROM vtab
WHERE _search_frame_=BuildMbr(-2.10,49.38,-1.3,49.99,4326)
```

Spatial binary predicates like `ST_Intersects` are sped up significantly when used in conjunction with this spatial index syntax.

11.3 Georreferenciador

The Georeferencer is a tool for aligning unreferenced raster or vector layers to known coordinate systems using Ground Control Points (GCPs). It supports exporting transformed rasters (GeoTIFF) and vectors (shapefiles, geopackages, etc.) or writing accompanying world files (for raster only). The basic approach to georeferencing a layer is to locate points on it for which you can accurately determine coordinates.

Feições

Tabela11.1: Georeferencer Tools

Ícone	Finalidade	Ícone	Finalidade
	Abrir raster		Open vector
	Iniciar georreferenciamento		Generate GDAL Script
	Load GCP Points		Save GCP Points As
	Configurações de transformação		
	Add GCP Point		Delete GCP Point
	Move GCP Point		Snapping Type
	Show Advanced Digitizing Dock		Pan
	Zoom In		Zoom Out
|zoomParaCamada|	Aproximar para camada		Zoom Last
	Zoom Next		Link Georeferencer to QGIS
	Link QGIS to Georeferencer		Full histogram stretch
	Local histogram stretch		

11.3.1 Georeferencing raster layer

As X and Y coordinates (DMS (dd mm ss.ss), DD (dd.dd) or projected coordinates (mmmm.mm)), which correspond with the selected point on the image, two alternative procedures can be used:

- The raster itself sometimes provides crosses with coordinates "written" on the image. In this case, you can enter the coordinates manually.
- Using already georeferenced layers. This can be either vector or raster data that contain the same objects/features that you have on the image that you want to georeference and with the projection that you want for your image. In this case, you can enter the coordinates by clicking on the reference dataset loaded in the QGIS map canvas.

The usual procedure for georeferencing an image involves selecting multiple points on the raster, specifying their coordinates, and choosing a relevant transformation type. Based on the input parameters and data, the Georeferencer will compute the world file parameters. The more coordinates you provide, the better the result will be.

The first step is to start QGIS and click on *Layer* ► *Georeferencer*, which appears in the QGIS menu bar. The Georeferencer dialog appears as shown in Fig.11.38.

For this example, we are using a topo sheet of South Dakota from SDGS. It can later be visualized together with the data from the GRASS `spearfish60` location. You can download the topo sheet here: https://grass.osgeo.org/sampledata/spearfish_toposheet.tar.gz.

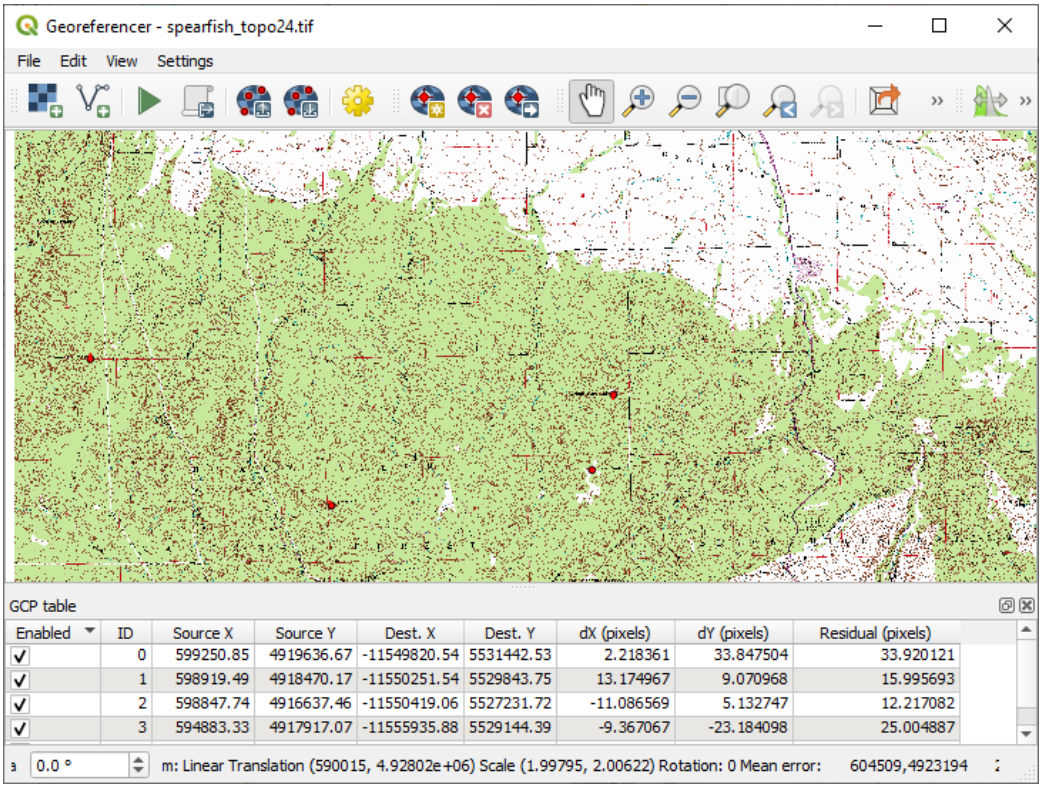


Figura11.38: Georeferencer Dialog

Entering ground control points (GCPs)

1. To start georeferencing an unreferenced raster, we must load it using the button. The raster will show up in the main working area of the dialog. Once the raster is loaded, we can start to enter reference points.

2. Using the Add GCP Point button, add points to the main working area and enter their coordinates (see Figure Fig.11.39). For this procedure you have the following options:

   - Click on a point in the raster image and enter the X and Y coordinates manually, along with the CRS of the point.
   - Click on a point in the raster image and choose the From map canvas button to add the X and Y coordinates with the help of a georeferenced map already loaded in the QGIS map canvas. The CRS will be set automatically.
   - When entering GCPs from the main map canvas, you have the option to hide the georeferencer window while selecting points from the main canvas. If the *Automatically hide georeferencer window* checkbox is ticked, after clicking *From Map Canvas*, the main georeferencer window will be hidden until a point is added on the map canvas. The *Enter Map Coordinates* dialog will remain open. If the box is unchecked, both windows will remain open while selecting a point on the map canvas. This option only takes effect when the georeferencer window is not docked in the main interface.

3. Continue entering points. You should have at least four points, and the more coordinates you can provide, the better the result will be. There are additional tools for zooming and panning the working area in order to locate a relevant set of GCP points.

> **Dica**

To avoid constant switching between Pan, Add GCP point and Move GCP point buttons, you may use the keyboard arrow keys for moving and the mouse wheel for scaling the georeferenced map conveniently.

4. After you provide a few points, you can use the Link QGIS to Georeferencer and/or Link Georeferencer to QGIS buttons that will adjust, respectively, the map extent of the main QGIS window to the present view in Georeferencer and/or vice versa.
5. With the tool, you can move the GCPs in both the canvas and the georeferencing window, if you need to correct them.

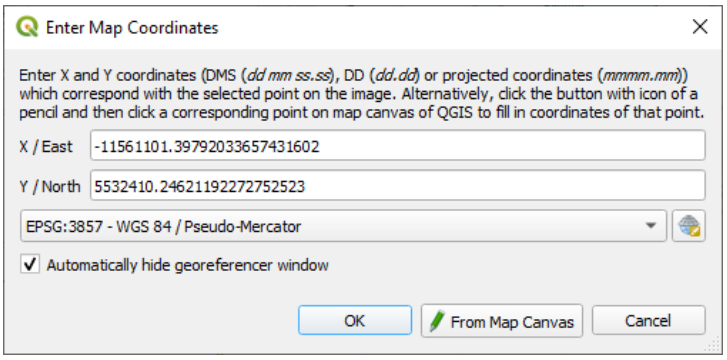


Figura11.39: Adicionar pontos à imagem matricial

The points that are added to the map will be stored in a separate text file (`[filename].points`) usually together with the raster image. This allows us to reopen the Georeferencer at a later date and add new points or delete existing ones to optimize the result. The points file contains values of the form: `mapX, mapY, pixelX, pixelY`. You can use the Load GCP points and Save GCP points as buttons to manage the files.

Definindo as configurações de transformação

After you have added your GCPs to the raster image, you need to define the transformation settings for the georeferencing process.

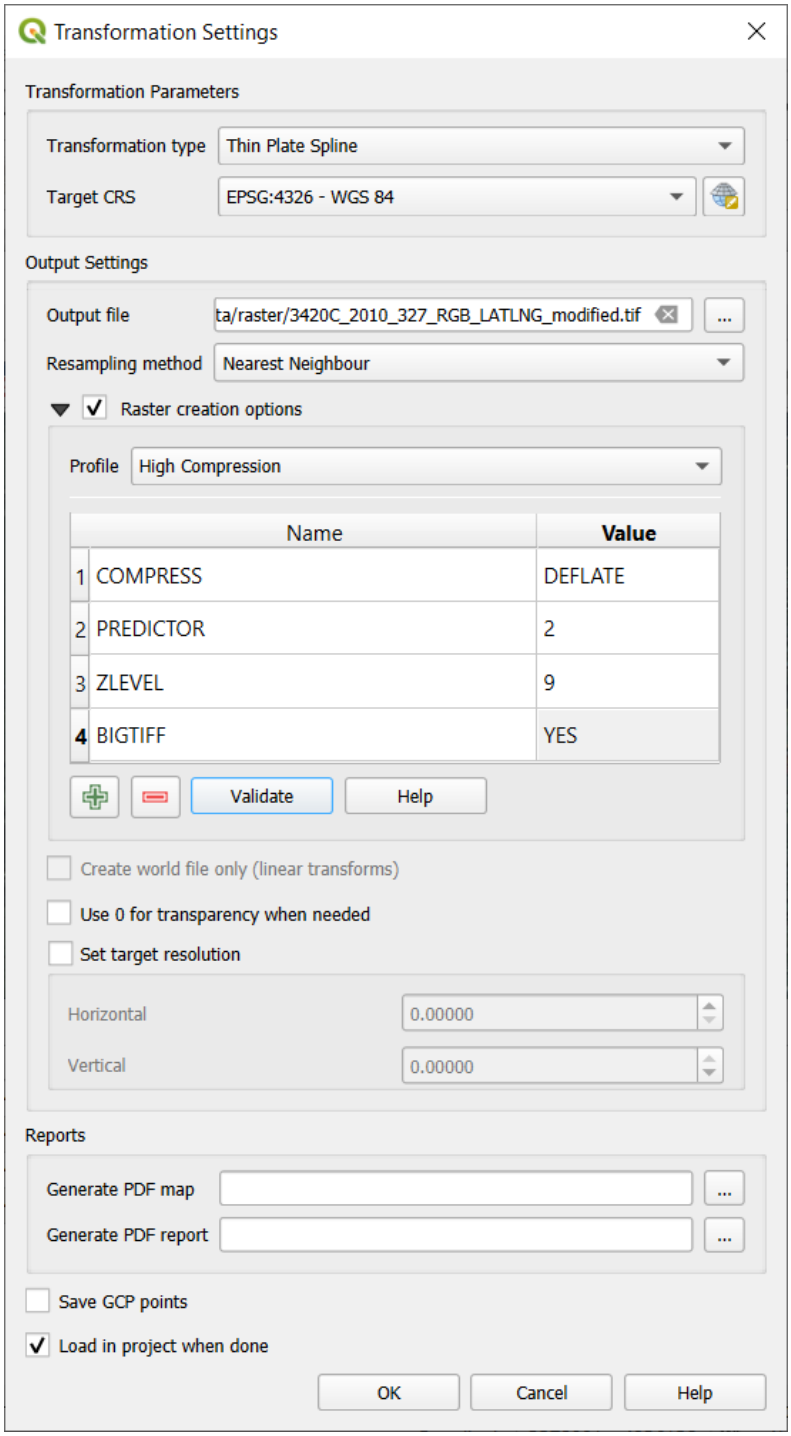


Figura11.40: Definindo as configurações de transformação do georreferenciador

Algoritmos de Transformação disponíveis

A number of transformation algorithms are available, dependent on the type and quality of input data, the nature and amount of geometric distortion that you are willing to introduce to the final result, and the number of ground control points (GCPs).

Currently, the following *Transformation types* are available:

- The **Linear** algorithm is used to create a world file and is different from the other algorithms, as it does not actually transform the raster pixels. It allows positioning (translating) the image and uniform scaling, but no rotation or other transformations. It is the most suitable if your image is a good quality raster map, in a known CRS, but is just missing georeferencing information. At least 2 GCPs are needed.
- The **Helmert** transformation also allows rotation. It is particularly useful if your raster is a good quality local map or orthorectified aerial image, but not aligned with the grid bearing in your CRS. At least 2 GCPs are needed.
- The **Polynomial 1** algorithm allows a more general affine transformation, in particular also a uniform shear. Straight lines remain straight (i.e., collinear points stay collinear) and parallel lines remain parallel. This is particularly useful for georeferencing data cartograms, which may have been plotted (or data collected) with different ground pixel sizes in different directions. At least 3 GCP's are required.
- The **Polynomial** algorithms 2-3 use more general 2nd or 3rd degree polynomials instead of just affine transformation. This allows them to account for curvature or other systematic warping of the image, for instance photographed maps with curving edges. At least 6 (respectively 10) GCP's are required. Angles and local scale are not preserved or treated uniformly across the image. In particular, straight lines may become curved, and there may be significant distortion introduced at the edges or far from any GCPs arising from extrapolating the data-fitted polynomials too far.
- The **Projective** algorithm generalizes Polynomial 1 in a different way, allowing transformations representing a central projection between 2 non-parallel planes, the image and the map canvas. Straight lines stay straight, but parallelism is not preserved and scale across the image varies consistently with the change in perspective. This transformation type is most useful for georeferencing angled photographs (rather than flat scans) of good quality maps, or oblique aerial images. A minimum of 4 GCPs is required.
- Finally, the **Thin Plate Spline** (TPS) algorithm "rubber sheets" the raster using multiple local polynomials to match the GCPs specified, with overall surface curvature minimized. Areas away from GCPs will be moved around in the output to accommodate the GCP matching, but will otherwise be minimally locally deformed. TPS is most useful for georeferencing damaged, deformed, or otherwise slightly inaccurate maps, or poorly orthorectified aerials. It is also useful for approximately georeferencing and implicitly reprojecting maps with unknown projection type or parameters, but where a regular grid or dense set of ad-hoc GCPs can be matched with a reference map layer. It technically requires a minimum of 10 GCPs, but usually more to be successful.

In all of the algorithms except TPS, if more than the minimum GCPs are specified, parameters will be fitted so that the overall residual error is minimized. This is helpful to minimize the impact of registration errors, i.e. slight imprecisions in pointer clicks or typed coordinates, or other small local image deformations. Absent other GCPs to compensate, such errors or deformations could translate into significant distortions, especially near the edges of the georeferenced image. However, if more than the minimum GCPs are specified, they will match only approximately in the output. In contrast, TPS will precisely match all specified GCPs, but may introduce significant deformations between nearby GCPs with registration errors.

Definir o método de Reamostragem

The type of resampling you choose will likely depend on your input data and the ultimate objective of the exercise. If you don't want to change statistics of the raster (other than as implied by nonuniform geometric scaling if using other than the Linear, Helmert, or Polynomial 1 transformations), you might want to choose 'Nearest neighbour'. In contrast, 'cubic resampling', for instance, will usually generate a visually smoother result.

É possível escolher entre cinco métodos de reamostragem diferentes:

1. Vizinho mais próximo
2. Bilinear (2x2 kernel)
3. Cubic (4x4 kernel)
4. Cubic B-Spline (4x4 kernel)
5. Lanczos (6x6 kernel)

Define the Raster creation options

When exporting a raster, *Raster creation options* allows you to define additional options that control how the output file is structured and compressed. See more at *Raster driver options*.

Dica

Select an empty entry if you want to create your own custom combination of parameters.

Definir as configurações de transformação

Existem várias opções que precisam ser definidas para a matriz de saída georreferenciada.

- The *Create world file* checkbox is only available if you decide to use the linear transformation type, because this means that the raster image actually won't be transformed. In this case, the *Output raster* field is not activated, because only a new world file will be created.
- For all other transformation types, you have to define an *Output raster*. As default, a new file ([file-name]_modified) will be created in the same folder together with the original raster image.
- As a next step, you have to define the *Target CRS* (Coordinate Reference System) for the georeferenced raster (see *Trabalhando com Projeções*).
- If you like, you can **generate a pdf map** and also **a pdf report**. The report includes information about the used transformation parameters, an image of the residuals and a list with all GCPs and their RMS errors.
- Furthermore, you can activate the *Set Target Resolution* checkbox and define the pixel resolution of the output raster. Default horizontal and vertical resolution is 1.
- The *Use 0 for transparency when needed* can be activated, if pixels with the value 0 shall be visualized transparent. In our example toposheet, all white areas would be transparent.
- The *Save GCP Points* will store GCP Points in a file next to the output raster.
- Finally, *Load in project when done* loads the output raster automatically into the QGIS map canvas when the transformation is done.

Executando a transformação

After all GCPs have been collected and all transformation settings are defined, just press the Start georeferencing button to create the new georeferenced raster.

11.3.2 Georeferencing vector layer

Georeferencing vector layers works similarly to raster georeferencing, but instead of matching image pixels, you match vector geometries (points, lines, or polygons) to known spatial references.

The standard procedure starts the same as for raster georeferencing: open QGIS and add a layer to the map canvas to use as a reference. This can be a georeferenced raster or vector layer, or a WMS layer.

Open the Georeferencer dialog from *Layer* ► *Georeferencer*.

Start georeferencing by following these steps (in this example, we use an unreferenced `alaska.shp`):

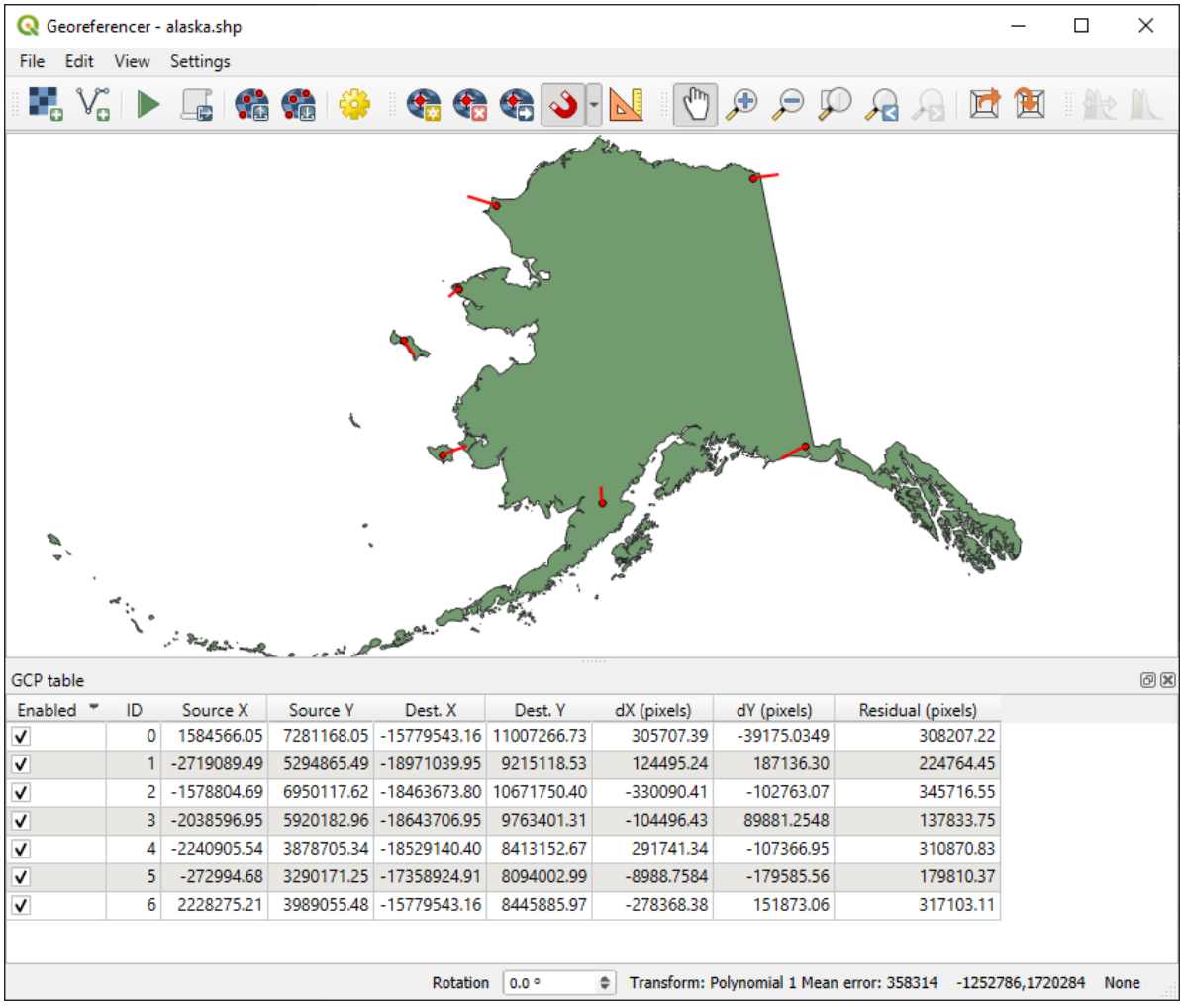


Figura11.41: Vector Georeferencer Dialog

1. Load the unreferenced vector layer using the button. The vector layer will appear in the main working area of the dialog.

2. Activate snapping by clicking the button and selecting the desired snapping type(s). This enables snapping to the reference layer when placing GCPs.

You can also use the Advanced Digitizing dock to ensure high-precision point selection. For more information, refer to the *Advanced Digitizing* section.

3. Use the Add GCP Point button to add a point to the working area. Enter its coordinates manually and set the CRS, or click the From map canvas button to pick the coordinates from a georeferenced layer in the main QGIS map canvas. In that case, the CRS will be set automatically.
4. Define the transformation settings:
   - Select the *Transformation type*. The transformation algorithms are the same as those for raster georeferencing. See *Definindo as configurações de transformação* for more details.
   - Define the *Target CRS* (Coordinate Reference System) for the georeferenced vector (see *Trabalhando com Projeções*).
   - Set the output file format and path (e.g., GeoPackage, Shapefile). By default, a new file with suffix `_modified` will be created in the same folder as the original vector file.
   - Optionally, enable **Generate PDF map** and **Generate PDF report**. The report includes transformation parameters, GCP residuals, and a summary of RMS errors.
   - Enable *Save GCP Points* to store GCPs in a file alongside the output vector layer.
   - Enable *Load in project when done* to add the result directly to the map canvas.
5. Click Start georeferencing to run the transformation and generate the georeferenced vector layer.

11.3.3 Show and adapt layer properties

Clicking on the *Source properties* option in the *Settings* menu opens the *Raster Layer properties* or *Vector Layer properties* depending on the type of layer you are georeferencing.

11.3.4 Configurar o georreferenciador

You can customize the behavior of the georeferencer in *Settings* ► *Configure Georeferencer* (or use keyboard shortcut `Ctrl+P`).

- Under *Point Tip* you can use the checkboxes to toggle displaying GCP IDs and X/Y coordinates in both the Georeferencer window and the main map canvas.
- *Residual Units* controls whether residual units are given in pixels or map units
- *PDF Report* allows you to set margin size in mm for the report export
- *PDF Map* allows you to choose a paper size for the map export
- Finally, you can activate to *Show Georeferencer window docked*. This will dock the Georeferencer window in the main QGIS window rather than showing it as a separate window that can be minimized.

11.4 Explorando Formatos de Dados e Campos

11.4.1 Dados raster

GIS raster data are matrices of discrete cells that represent features / phenomena on, above or below the earth's surface. Each cell in the raster grid has the same size, and cells are usually rectangular (in QGIS they will always be rectangular). Typical raster datasets include remote sensing data, such as aerial photography, or satellite imagery and modelled data, such as elevation or temperature.

Ao contrário dos dados vetoriais, os dados matriciais normalmente não possuem um registro de banco de dados associado para cada célula. Eles são geocodificados pela resolução do pixel e pela coordenada X/Y de um pixel de canto da camada matricial. Isso permite que o QGIS posicione os dados corretamente na tela do mapa.

The GeoPackage format is convenient for storing raster data when working with QGIS. The popular and powerful GeoTiff format is a good alternative.

QGIS makes use of georeference information inside the raster layer (e.g., GeoTiff) or an associated *world file* to properly display the data.

11.4.2 Dado Vetorial

Many of the features and tools available in QGIS work the same, regardless of the vector data source. However, because of the differences in format specifications (GeoPackage, ESRI Shapefile, MapInfo and MicroStation file formats, AutoCAD DXF, PostgreSQL, SpatiaLite, Oracle Spatial, MS SQL Server, SAP HANA Spatial databases and many more), QGIS may handle some of their properties differently. Support is provided by the GDAL vector drivers. This section describes how to work with some of these specifics.

> **Nota**
>
> QGIS supports (multi)point, (multi)line, (multi)polygon, CircularString, CompoundCurve, CurvePolygon, MultiCurve, MultiSurface feature types, all optionally with Z and/or M values.
>
> You should also note that some drivers don't support some of these feature types, like CircularString, CompoundCurve, CurvePolygon, MultiCurve, MultiSurface feature type. QGIS will convert them.

GeoPackage

The GeoPackage (GPKG) format is platform-independent, and is implemented as a SQLite database container, and can be used to store both vector and raster data. The format was defined by the Open Geospatial Consortium (OGC), and was published in 2014.

GeoPackage can be used to store the following in a SQLite database:

- **vector** features
- **tile matrix sets of imagery** and **raster** maps
- atributos (dados não espaciais)
- extensões

Since QGIS version 3.8, GeoPackage can also store QGIS projects. GeoPackage layers can have JSON fields.

GeoPackage é o formato padrão para dados vetoriais no QGIS.

ESRI Shapefile format

The ESRI Shapefile format is still one of the most used vector file formats, even if it has some limitations compared to for instance GeoPackage and SpatiaLite.

An ESRI Shapefile format dataset consists of several files. The following three are required:

1. `.shp` file containing the feature geometries
2. `.dbf` arquivo que contém os atributos no formato dBase
3. `.shx` index file

An ESRI Shapefile format dataset can also include a file with a `.prj` suffix, which contains projection information. While it is very useful to have a projection file, it is not mandatory. A Shapefile format dataset can contain additional files. For further details, see the the ESRI technical specification.

GDAL has read-write support for compressed ESRI Shapefile format (`shz` and `shp.zip`).

Improving Performance for ESRI Shapefile format datasets

To improve the drawing performance for an ESRI Shapefile format dataset, you can create a spatial index. A spatial index will improve the speed of both zooming and panning. Spatial indexes used by QGIS have a `.qix` extension.

Usar estas etapas para criar o índice:

1. Load an ESRI Shapefile format dataset (see *The Browser Panel*)
2. Abra a caixa de diálogo *Propriedades da Camada* clicando duas vezes no nome da camada na legenda ou clicando com o botão direito do mouse e escolhendo *Propriedades…* no menu de contexto
3. In the *Source* tab, click the *Create Spatial Index* button

Problema ao carregar um arquivo .prj

If you load an ESRI Shapefile format dataset with a `.prj` file and QGIS is not able to read the coordinate reference system from that file, you will need to define the proper projection manually in the *Layer Properties* ► *Source* tab of the layer by clicking the Select CRS button. This is due to the fact that `.prj` files often do not provide the complete projection parameters as used in QGIS and listed in the *CRS* dialog.

For the same reason, if you create a new ESRI Shapefile format dataset with QGIS, two different projection files are created: a `.prj` file with limited projection parameters, compatible with ESRI software, and a `.qpj` file, providing all the parameters of the CRS. Whenever QGIS finds a `.qpj` file, it will be used instead of the `.prj`.

Delimited Text Files

Arquivos de texto delimitados são muito comuns e amplamente utilizados devido à sua simplicidade e legibilidade - os dados podem ser visualizados e editados em um editor de texto simples. Um arquivo de texto delimitado é um dado de tabela com colunas separadas por um caractere definido e linhas separadas por quebras de linha. A primeira linha geralmente contém os nomes das colunas. Um tipo comum de arquivo de texto delimitado é um CSV (Comma Separated Values), com colunas separadas por vírgulas. Arquivos de texto delimitados também podem conter informações posicionais (veja *Armazenando informações de geometria em arquivos de texto delimitados*).

QGIS allows you to load a delimited text file as a layer or an ordinary table (see *The Browser Panel* or *Importando um arquivo de texto delimitado*). First check that the file meets the following requirements:

1. O arquivo deve ter uma linha de cabeçalho delimitada de nomes de campo. Esta deve ser a primeira linha dos dados (de preferência a primeira linha no arquivo de texto).
2. Se a geometria deve ser habilitada, o arquivo deve conter campos que definem a geometria. Esses campos podem ter qualquer nome.
3. Os campos de coordenadas X e Y (se a geometria for definida por coordenadas) devem ser especificados como números. O sistema de coordenadas não é importante.
4. If you have a CSV file with non-string columns, you can have an accompanying CSVT file (see section *Using CSVT file to control field formatting*).

The elevation point data file `elevp.csv` in the QGIS sample dataset (see section *Baixando dados de amostra*) is an example of a valid text file:

```
X;Y;ELEV
-300120;7689960;13
-654360;7562040;52
1640;7512840;3
[...]
```

Algumas coisas a serem observadas sobre o arquivo de texto:

1. O arquivo de texto de exemplo usa `;` (ponto e vírgula) como delimitador (qualquer caractere pode ser usado para delimitar os campos).
2. A primeira linha é a linha de cabeçalho. Ela contém os campos `X`, `Y` e `ELEV`.
3. No quotes (`"`) are used to delimit text fields
4. As coordenadas X estão contidas no campo `X`
5. As coordenadas Y estão contidas no campo `Y`

Armazenando informações de geometria em arquivos de texto delimitados

Arquivos de texto delimitados podem conter informações de geometria em duas formas principais:

- Como coordenadas em colunas separadas (ex. `Xcol`, `Ycol`...), para dados de geometria de ponto;
- As well-known text (WKT) representation of geometry in a single column, for any geometry type.

Features with curved geometries (CircularString, CurvePolygon and CompoundCurve) are supported. Here are some examples of geometry types in a delimited text file with geometries coded as WKT:

```
Label;WKT_geom
LineString;LINESTRING(10.0 20.0, 11.0 21.0, 13.0 25.5)
CircularString;CIRCULARSTRING(268 415,227 505,227 406)
CurvePolygon;CURVEPOLYGON(CIRCULARSTRING(1 3, 3 5, 4 7, 7 3, 1 3))
CompoundCurve;COMPOUNDCURVE((5 3, 5 13), CIRCULARSTRING(5 13, 7 15,
  9 13), (9 13, 9 3), CIRCULARSTRING(9 3, 7 1, 5 3))
```

Arquivos de texto delimitados também suportam coordenadas Z e M em geometrias:

```
LINESTRINGZ(10.0 20.0 30.0, 11.0 21.0 31.0, 11.0 22.0 30.0)
```

Using CSVT file to control field formatting

When loading CSV files, the GDAL driver assumes all fields are strings (i.e. text) unless it is told otherwise. You can create a CSVT file to tell GDAL (and QGIS) the data type of the different columns:

Tipo	Nome	Exemplo
Número inteiro	Número inteiro	4
Booleano	Integer(Boolean)	true
Número decimal	Real	3.456
Data	Data (AAAA-MM-DD)	2016-07-28
Hora	Hora (HH:MM:SS+nn)	18:33:12+00
Data & Hora	DateTime (YYYY-MM-DD HH:MM:SS+nn)	2016-07-28 18:33:12+00
CoordX	CoordX	8.8249
CoordY	CoordY	47.2274
Point(X)	Point(X)	8.8249
Point(Y)	Point(Y)	47.2274
WKT	WKT	POINT(15 20)

The CSVT file is a **ONE line** plain text file with the data types in quotes and separated by commas, e.g.:

```
"Integer","Real","String"
```

Você pode até especificar a largura e a precisão de cada coluna, por exemplo:

```
"Integer(6)","Real(5.5)","String(22)"
```

Este arquivo é salvo na mesma pasta que o arquivo `.csv`, com o mesmo nome, mas `.csvt` como a extensão.

You can find more information at GDAL CSV Driver.

Dica

Detect Field Types

Instead of using a CSVT file to tell the data types, QGIS provides the possibility to automatically detect the field types and to change the assumed field types.

PostgreSQL Database

In order to process geographical information in a PostgreSQL database, the PostGIS extension must first be installed. PostGIS extends the capabilities of the PostgreSQL relational database by adding support for storing, indexing, and querying geospatial data.

To enable PostGIS in your database you have to activate the extension in your database (open database and run `CREATE EXTENSION postgis;`). PostGIS enabled databases are also often named “PostGIS layer”. Using PostGIS, vector functions such as select and identify work more accurately than they do with GDAL layers in QGIS.

Dica

Camadas PostGIS

Normalmente, uma camada PostGIS é identificada por uma entrada na tabela de geometria_colunas. QGIS pode carregar camadas que não têm uma entrada na tabela de geometria_colunas. Isto inclui tanto as tabelas quanto as vistas. Consulte seu manual PostgreSQL para obter informações sobre como criar vistas.

This section contains some details on how QGIS accesses PostgreSQL layers. Most of the time, QGIS should simply provide you with a list of database tables that can be loaded, and it will load them on request. However, if you have trouble loading a PostgreSQL table into QGIS, the information below may help you understand QGIS messages and give you directions for modifying the PostgreSQL table or view definition to allow QGIS to load it.

> **Nota**
>
> Um banco de dados PostgreSQL também pode armazenar projetos QGIS.

Chave primária

QGIS requires that PostgreSQL layers contain a column that can be used as a unique key for the layer. For tables, this usually means that the table needs a primary key, or a column with a unique constraint on it. In QGIS, this column needs to be of type int4 (an integer of size 4 bytes). Alternatively, the ctid column can be used as primary key. If a table lacks these items, the oid column will be used instead. Performance will be improved if the column is indexed (note that primary keys are automatically indexed in PostgreSQL).

QGIS offers a checkbox **Select at id** that is activated by default. This option gets the ids without the attributes, which is faster in most cases.

Ver

Se a camada do PostgreSQL é uma visualização, o mesmo requisito existe, mas as visualizações nem sempre possuem chaves ou colunas primárias com restrições exclusivas. Você precisa definir um campo de chave primária (deve ser inteiro) na caixa de diálogo QGIS antes de poder carregar a exibição. Se uma coluna adequada não existir na exibição, o QGIS não carregará a camada. Se isso ocorrer, a solução é alterar a exibição para incluir uma coluna adequada (um tipo de número inteiro e uma chave primária ou com uma restrição exclusiva, preferencialmente indexada).

As for table, a checkbox **Select at id** is activated by default (see above for the meaning of the checkbox). It can make sense to disable this option when you use expensive views.

> **Nota**
>
> **PostgreSQL foreign table**
>
> PostgreSQL foreign tables are not explicitely supported by the PostgreSQL provider and will be handled like a view.

QGIS layer_style table and database backup

If you want to make a backup of your PostgreSQL database using the `pg_dump` and `pg_restore` commands, and the default layer styles as saved by QGIS fail to restore afterwards, you need to set the XML option to `DOCUMENT` before the restore command:

1. Fazer um backup SIMPLES da tabela `layer_style`
2. Abrir o arquivo dentro de um editor de texto
3. Mude a linha `SET xmloption = content;` para `SET XML OPTION DOCUMENT;`
4. Salvar o arquivo
5. Use psql para restaurar a tabela no novo banco de dados

Filter database side

QGIS allows to filter features already on server side. Check *Settings* ► *Options* ► *Data Sources* ► *Execute expressions on server-side if possible* to do so. Only supported expressions will be sent to the database. Expressions using unsupported operators or functions will gracefully fallback to local evaluation.

Support of PostgreSQL data types

Data types supported by the PostgreSQL provider include: integer, float, boolean, binary object, varchar, geometry, timestamp, array, hstore and json.

Importing Data into PostgreSQL

Data can be imported into PostgreSQL using several tools, including the Browser, DB Manager plugin and the command line tools shp2pgsql and ogr2ogr.

DB Manager

O QGIS vem com um complemento principal chamado Gerenciador de BD. Ele pode ser usado para carregar dados e inclui suporte para esquemas. Veja a seção *Plugin do gerenciador de banco de dados* para mais informações.

shp2pgsql

PostGIS includes a utility called **shp2pgsql**, that can be used to import Shapefile format datasets into a PostGIS-enabled database. For example, to import a Shapefile format dataset named `lakes.shp` into a PostgreSQL database named `gis_data`, use the following command:

```
shp2pgsql -s 2964 lakes.shp lakes_new | psql gis_data
```

This creates a new layer named `lakes_new` in the `gis_data` database. The new layer will have a spatial reference identifier (SRID) of 2964. See section *Trabalhando com Projeções* for more information about spatial reference systems and projections.

> **Dica**
>
> **Exporting datasets from PostgreSQL**
>
> There is also a tool for exporting postgreSQL datasets wth geographic data to Shapefile format: **pgsql2shp**. It is shipped within your PostGIS distribution.

ogr2ogr

In addition to **shp2pgsql** and **DB Manager**, there is another tool for feeding geographical data in PostgreSQL: **ogr2ogr**. It is part of your GDAL installation.

To import a Shapefile format dataset into PostgreSQL, do the following:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres
password=topsecret" alaska.shp
```

This will import the Shapefile format dataset `alaska.shp` into the database *postgis* using the user *postgres* with the password *topsecret* on the host server *myhost.de*.

Note that GDAL must be built with PostgreSQL. You can verify this by typing (in):

```
ogrinfo --formats | grep -i post
```

If you prefer to use the PostgreSQL's **COPY** command instead of the default **INSERT INTO** method, you can export the following environment variable (at least available on and):

```
export PG_USE_COPY=YES
```

ogr2ogr does not create spatial indexes like **shp2pgsl** does. You need to create them manually, using the normal SQL command **CREATE INDEX** afterwards, as an extra step (as described in the next section *Melhorando a Performance*).

Melhorando a Performance

Retrieving features from a PostgreSQL database can be time-consuming, especially over a network. You can improve the drawing performance of PostgreSQL layers by ensuring that a spatial index exists on each layer in the database. PostGIS supports creation of a GiST (Generalized Search Tree) index to speed up spatial searching (GiST index information is taken from the PostGIS documentation available at https://postgis.net).

Dica

You can use the DBManager to create an index for your layer. You should first select the layer and click on *Table* ► *Edit table*, go to *Indexes* tab and click on *Add Spatial Index*.

The syntax for creating a GiST index is:

```
CREATE INDEX [indexname] ON [tablename]
  USING GIST ( [geometryfield] GIST_GEOMETRY_OPS );
```

Note that for large tables, creating the index can take a long time. Once the index is created, you should perform a `VACUUM ANALYZE`. See the PostGIS documentation (POSTGIS-PROJECT in *Referências Bibliográficas e Web*) for more information.

The following example creates a GiST index:

```
gsherman@madison:~/current$ psql gis_data
Welcome to psql 8.3.0, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit

gis_data=# CREATE INDEX sidx_alaska_lakes ON alaska_lakes
gis_data-# USING GIST (geom GIST_GEOMETRY_OPS);
CREATE INDEX
gis_data=# VACUUM ANALYZE alaska_lakes;
VACUUM
gis_data=# \q
gsherman@madison:~/current$
```

Camadas SpatiaLite

If you want to save a vector layer using the SpatiaLite format, you can do this by following instructions at *Criando novas camadas a partir de uma camada existente*. You select `SpatiaLite` as *Format* and enter both *File name* and *Layer name*.

Also, you can select `SQLite` as format and then add `SPATIALITE=YES` in the *Custom Options* ► *Data source* field. This tells GDAL to create a SpatiaLite database. See also https://gdal.org/en/latest/drivers/vector/sqlite.html.

QGIS also supports editable views in SpatiaLite. For SpatiaLite data management, you can also use the core plugin *DB Manager*.

Se você deseja criar uma nova camada SpatiaLite, por favor consulte a seção *Creating a new SpatiaLite layer*.

GeoJSON specific parameters

When *exporting layers* to GeoJSON, there are some specific *Layer Options* available. These options come from GDAL which is responsible for the writing of the file:

- *COORDINATE_PRECISION* the maximum number of digits after the decimal separator to write in coordinates. Defaults to 15 (note: for Lat Lon coordinates 6 is considered enough). Truncation will occur to remove trailing zeros.
- *RFC7946* by default GeoJSON 2008 will be used. If set to YES, the updated RFC 7946 standard will be used. Default is NO (thus GeoJSON 2008). See https://gdal.org/en/latest/drivers/vector/geojson.html#rfc-7946-write-support for the main differences, in short: only EPSG:4326 is allowed, other crs's will be transformed, polygons will be written such as to follow the right-hand rule for orientation, values of a "bbox" array are [west, south, east, north], not [minx, miny, maxx, maxy]. Some extension member names are forbidden in FeatureCollection, Feature and Geometry objects, the default coordinate precision is 7 decimal digits
- *WRITE_BBOX* set to YES to include the bounding box of the geometries at the feature and feature collection level

Besides GeoJSON there is also an option to export to "GeoJSON - Newline Delimited" (see https://gdal.org/en/latest/drivers/vector/geojsonseq.html). Instead of a FeatureCollection with Features, you can stream one type (probably only Features) sequentially separated with newlines.

GeoJSON - Newline Delimited has some specific Layer options availabe too:

- *COORDINATE_PRECISION* see above (same as for GeoJSON)
- *RS* whether to start records with the RS=0x1E character. The difference is how the features are separated: only by a newline (LF) character (Newline Delimited JSON, geojsonl) or by also prepending a record-separator (RS) character (giving GeoJSON Text Sequences, geojsons). Default to NO. Files are given the `.json` extension if extension is not provided.

SAP HANA Spatial Layers

This section contains some details on how QGIS accesses SAP HANA layers. Most of the time, QGIS should simply provide you with a list of database tables and views that can be loaded, and it will load them on request. However, if you have trouble loading an SAP HANA table or view into QGIS, the information below may help you understand the root cause and assist in resolving the issue.

Feature Identification

If you'd like to use all of QGIS' feature editing capabilities, QGIS must be able to unambiguously identify each feature in a layer. Internally, QGIS uses a 64-bit signed integer to identify features, whereas the negative range is reserved for special purposes.

Therefore, the SAP HANA provider requires a unique key that can be mapped to a positive 64-bit integer to fully support QGIS' feature editing capabilities. If it is not possible to create such a mapping, you might still view the features, but editing might not work.

Adicionando tabelas

When adding a table as a layer, the SAP HANA provider uses the table's primary key to map it to a unique feature id. Therefore, to have full feature editing support, you need to have a primary key to your table definition.

The SAP HANA provider supports multi-column primary keys, but if you'd like to get the best performance, your primary key should be a single column of type `INTEGER`.

Adding views

When adding a view as a layer, the SAP HANA provider cannot automatically identify columns that unambiguously identify a feature. Furthermore, some views are read-only and cannot be edited.

To have full feature editing support, the view must be updatable (check column `IS_READ_ONLY` in system view `SYS.VIEWS` for the view in question) and you must manually provide QGIS with one or more columns that identify a feature. The columns can be given by using *Layer* ► *Add Layer* ► *Add SAP HANA Spatial Layer* and then selecting the columns in the *Feature id* column. For best performance, the *Feature id* value should be a single `INTEGER` column.

11.4.3 Layers crossing 180° longitude

Many GIS packages don't wrap layers with a geographic reference system (lat/lon) crossing the 180 degrees longitude line. As result, if we open such a layer in QGIS, we could see two widely separated locations, that should appear near each other. In Fig.11.42, the tiny point on the far left of the map canvas (Chatham Islands) should be within the grid, to the right of the New Zealand main islands.

Figura11.42: Map in lat/lon crossing the 180° longitude line

Solving in PostgreSQL

A work-around is to transform the longitude values using PostgreSQL and the ST_ShiftLongitude function. This function reads every point/vertex in every component of every feature in a geometry, and shifts its longitude coordinate from -180..0° to 180..360° and vice versa if between these ranges. This function is symmetrical so the result is a 0..360° representation of a -180..180° data and a -180..180° representation of a 0..360° data.

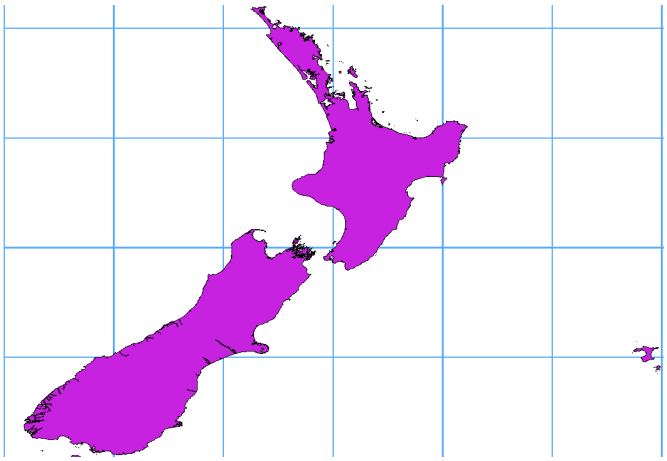


Figura11.43: Crossing 180° longitude applying the **ST_ShiftLongitude** function

1. Import data into PostgreSQL (*Importing Data into PostgreSQL*) using, for example, the DB Manager plugin.
2. Use the PostgreSQL command line interface to issue the following command:

```
-- In this example, "TABLE" is the actual name of your PostgreSQL table
update TABLE set geom=ST_ShiftLongitude(geom);
```

3. If everything went well, you should receive a confirmation about the number of features that were updated. Then you'll be able to load the map and see the difference (*Figure_vector_crossing_map*).

CAPÍTULO 12

Trabalhando com Dados Vetoriais

12.1 Janela de Propriedades de Vetor

A caixa de diálogo:guilabel:`Propriedades da camada ‘para uma camada vetorial fornece configurações gerais para gerenciar a aparência dos recursos da camada no mapa (simbologia, rotulagem, diagramas), interação com o mouse (ações, dicas de mapa, design de formulário). Ele também fornece informações sobre a camada.

Para acessar a caixa de diálogo:guilabel:*Propriedades da camada*:

- No painel:guilabel:*Camadas*, clique duas vezes na camada ou clique com o botão direito do mouse e selecione: guilabel:` Propriedades… `no menu pop-up;
- Vá para: seleção de menu: menu Camada -> Propriedades da camada … `quando a camada estiver selecionada.

O diálogo vetor:guilabel:*Propriedades da camada* fornece as seguintes seções:

metadados | *Informações*	sistema | *Fonte*	*Symbology* ([1])
Labels ([1])	*Mask* ([1])	*3D View* ([1])
diagrama | *Diagramas*	|**fonteCampos**| *Campos*	|**formulárioVisão**| Formulário de atributos
Joins	|**Armazenamento auxiliar**| *Armazenamento Auxiliar*	ação *Ações*
Display	*Rendering*	*Temporal*
|**expressão**| *Variáveis*	*Elevation*	|editMetadata | *Metadados*
Dependencies	*Legend*	|**sobreposição**| *Servidor QGIS*
|**digitalização**| *Digitalizando*	*External plugins* ([2]) tabs	

[1] Also available in the *Layer styling panel*
[2] *External plugins* you install can optionally add tabs to this dialog. Those are not presented in this document. Refer to their documentation.

Dica

** Compartilhe propriedades totais ou parciais dos estilos de camada **

A opção: seleção de menu: *Estilo* na parte inferior da caixa de diálogo permite importar ou exportar essas ou parte dessas propriedades de / para vários destinos (arquivo, área de transferência, banco de dados). Veja: ref: *manage_custom_style*.

Nota

Como as propriedades (simbologia, etiqueta, ações, valores padrão, formulários …) das camadas incorporadas (consulte:ref:` projetos de aninhamento`) são extraídas do arquivo original do projeto e, para evitar alterações que possam quebrar esse comportamento, o diálogo de propriedades da camada fica indisponível para essas camadas.

12.1.1 Propriedades da informação

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- general such as name in the project, source path, list of auxiliary files, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*, *legend widgets*, *layer variables*, *form properties*… More custom properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- based on the provider of the layer: format of storage, geometry type, data source encoding, extent, feature count…
- o Sistema de Referência de Coordenadas: nome, unidades, método, precisão, referência (ou seja, se é estático ou dinâmico)
- picked from the *filled metadata*: access, extents, links, contacts, history…
- e relacionados à sua geometria (extensão espacial, SRC…) ou seus atributos (número de campos, características de cada um…).

12.1.2 Propriedades de origem

|sistema| Use esta guia para definir configurações gerais para a camada vetorial.

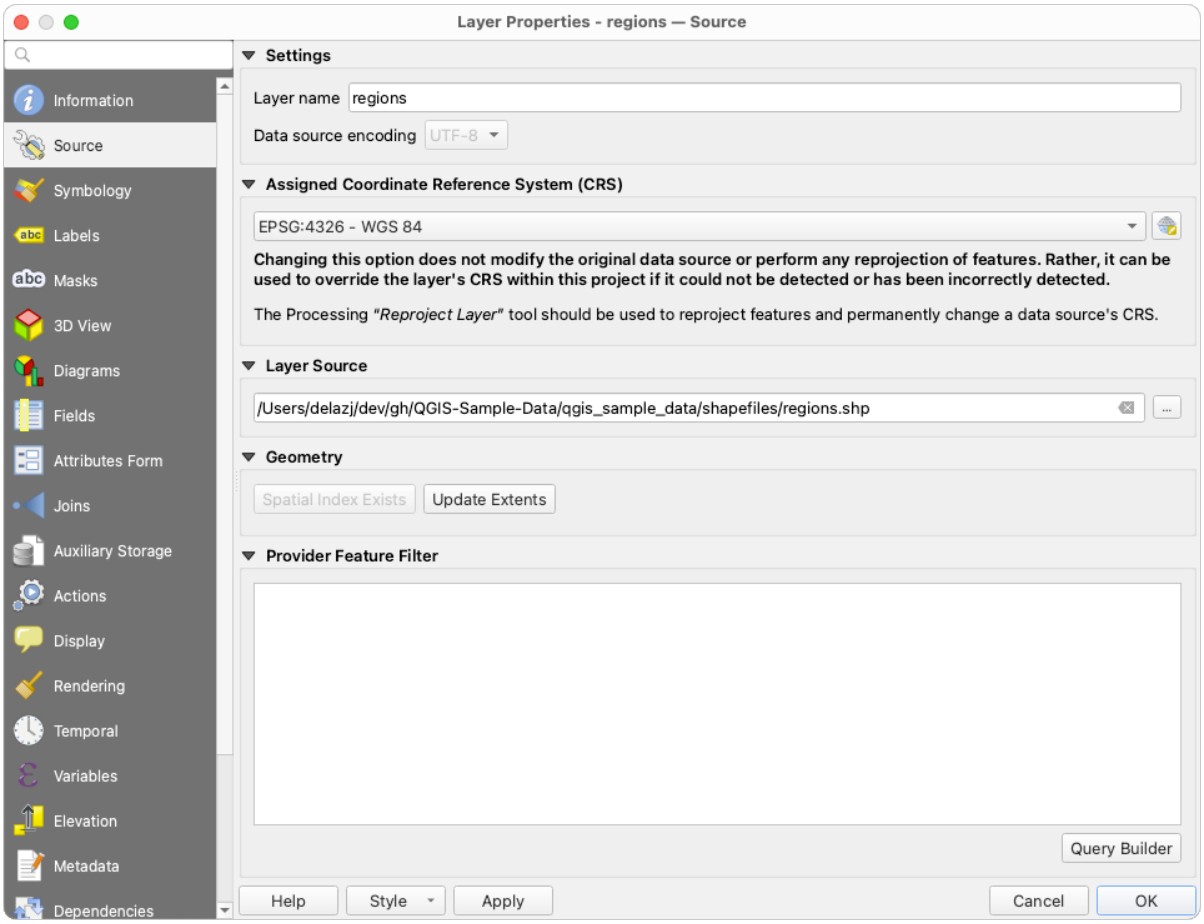


Figura12.1: Guia Origem na caixa de diálogo Propriedades da camada de vetor

Configurações

- Set a *Layer name* different from the layer filename that will be used to identify the layer in the project (in the *Layers Panel*, with expressions, in print layout legend, …)
- Depending on the data format, select the *Data source encoding* if not correctly detected by QGIS.

Coordinate Reference System

- Displays the layer's *Assigned Coordinate Reference System (CRS)*. You can change the layer's CRS, selecting a recently used one in the drop-down list or clicking on Select CRS button (see *Seletor do Sistemas de Referência de Coordenadas*). Use this process only if the CRS applied to the layer is a wrong one or if none was applied. If you wish to reproject your data into another CRS, rather use layer reprojection algorithms from Processing or *Save it into another layer*.

Layer source and Geometry

- Depending on the data provider, a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer:
  - When the layer is stored as file on disk, edit the path shown in the text box or press … Browse to select another file on the disk. Both layers do not need to share attribute fields, geometry type or file formats.
  - When the layer is provided by an ArcGIS Feature service, it is possible to modify its *authentication settings*, while keeping unchanged the details for connecting to the service.
- *Create spatial index* (only for OGR-supported formats): helps speeding layer rendering and features' geometry retrieval.
- :guilabel:informações da atualização de extensões para uma camada.

Criador de consultas

The Query Builder dialog is accessible through the *Query Builder* button at the bottom of the *Source* tab in the Layer Properties dialog, under the *Provider feature filter* group.

O Criador de consultas fornece uma interface que permite definir um subconjunto dos recursos na camada usando uma cláusula WHERE semelhante a SQL e exibir o resultado na janela principal. Enquanto a consulta estiver ativa, apenas os recursos correspondentes ao resultado estarão disponíveis no projeto.

You can use one or more layer attributes to define the filter in the `Query Builder`. The use of more than one attribute is shown in Fig.12.2. In the example, the filter combines the attributes

- Campo `` toa`` (`` Date Time``: `` cast ("toa" como personagem)>> 2017-05-17 ") e `` cast ("toa" como personagem) <'2019-12 -24T18: 00: 00 "),
- campo `` name`` (`` String``: `` "name"> 'S"```) e
- `` FID`` (campo `` Integer``: `` FID> 10``)

usando os operadores AND, OR e NOT e parênteses. Essa sintaxe (incluindo o formato DateTime para o campo `` toa``) funciona para conjuntos de dados Pacote geográfico.

The filter is made at the data provider (OGR, PostgreSQL, MS SQL Server…) level. So the syntax depends on the data provider (DateTime is for instance not supported for the ESRI Shapefile format). The complete expression:

```
cast("toa" as character) > '2017-05-17' AND
cast("toa" as character) < '2019-12-24T18:00:00' AND
NOT ("name" > 'S' OR FID > 10)
```

Você também pode abrir a caixa de diálogo:guilabel:*Criador de consultas* usando a opção: guilabel:` Filtro … *da opção: seleção de menus: menu* Camada` ou menu contextual da camada. As seções:guilabel:` Campos`,: guilabel:` Valores` e:guilabel:*Operadores* na caixa de diálogo ajudam a construir a consulta semelhante a SQL exposta na caixa: guilabel:` Expressão de filtro específica do provedor`.

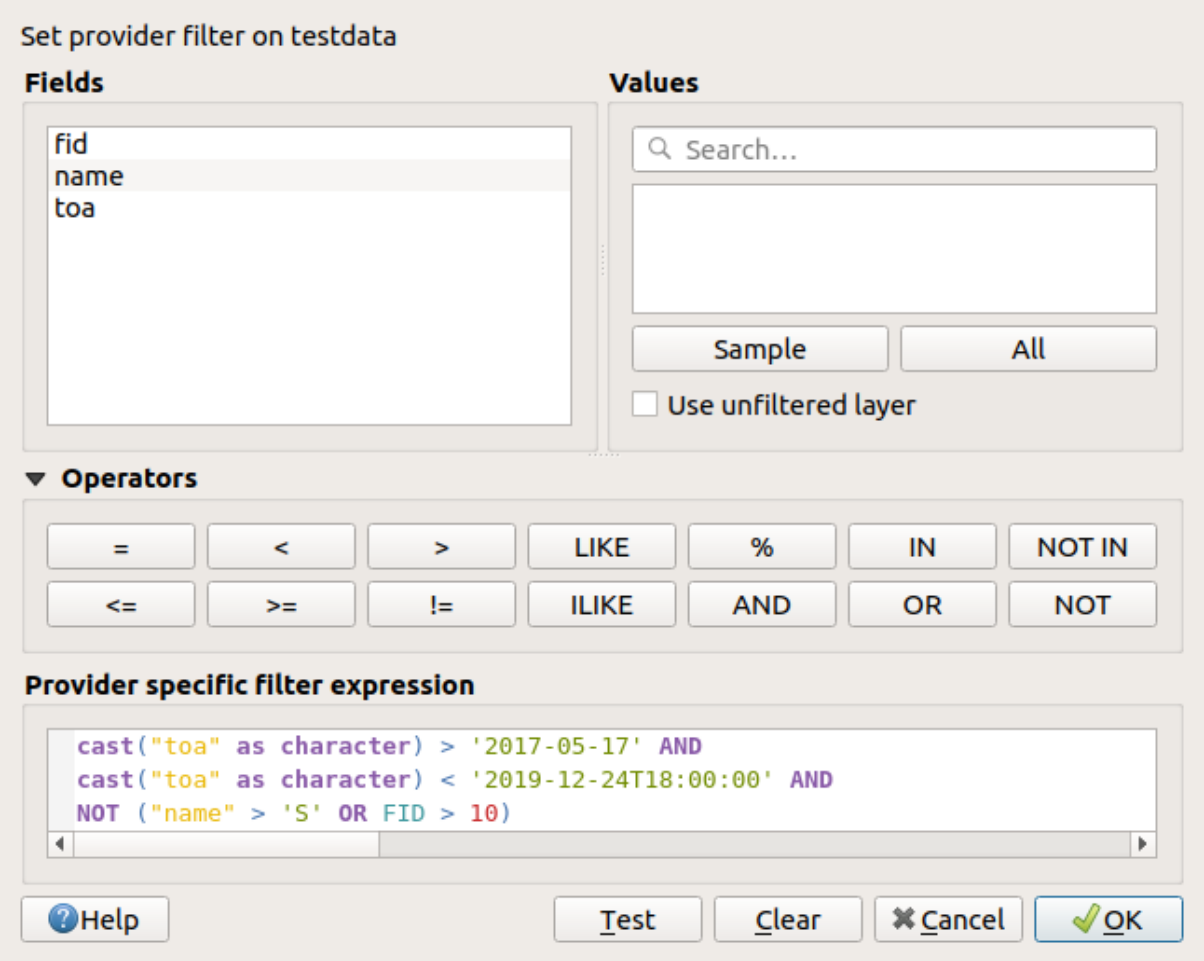


Figura12.2: Criador de consultas

A lista ** Campos ** contém todos os campos da camada. Para adicionar uma coluna de atributo ao campo da cláusula SQL WHERE, clique duas vezes no nome ou digite-o na caixa SQL.

O quadro ** Valores ** lista os valores do campo selecionado no momento. Para listar todos os valores exclusivos de um campo, clique no botão: guilabel: *Todos*. Para listar os 25 primeiros valores exclusivos da coluna, clique no botão:guilabel:*Amostra*. Para adicionar um valor ao campo da cláusula SQL WHERE, clique duas vezes em seu nome na lista Valores. Você pode usar a caixa de pesquisa na parte superior do quadro Valores para navegar e encontrar facilmente os valores dos atributos na lista.

A seção ** Operadores ** contém todos os operadores utilizáveis. Para adicionar um operador ao campo da cláusula SQL WHERE, clique no botão apropriado. Operadores relacionais (`` = , ``> ``, ...), operador de comparação de cadeias ( LIKE``) e operadores lógicos (`` AND``, `` OR``, ...) Estão disponíveis.

The *Test* button helps you check your query and displays a message box with the number of features satisfying the current query. Use the *Clear* button to wipe the SQL query and revert the layer to its original state (ie, fully load all the features). It is possible to *Save...* the query as a `.QQF` file, or *Load...* the query from a file into the dialog.

Quando um filtro é aplicado, o QGIS trata o subconjunto resultante como se fosse a camada inteira. Por exemplo, se você aplicou o filtro acima para 'Bairro' (`` "TYPE_2" = 'Borough') , não poderá exibir, consultar, salvar ou editar `` Ancoragem, porque é um' Município 'e portanto, não faz parte do subconjunto.

Dica

** As camadas filtradas são indicadas no painel Camadas **

In the *Layers* panel, filtered layer is listed with a Filter icon next to it indicating the query used when the mouse hovers over the button. Double-click the icon opens the *Query Builder* dialog for edit. This can also be achieved through the *Layer* ► *Filter…* menu.

12.1.3 Propriedades da simbologia

The *Symbology* tab provides you with a comprehensive tool for rendering and symbolizing your vector data. You can use tools that are common to all vector data, as well as special symbolizing tools that were designed for the different kinds of vector data. However all types share the following dialog structure: in the upper part, you have a widget that helps you prepare the classification and the symbol to use for features and at the bottom the *Renderização da camada* widget.

Dica

** Alterne rapidamente entre diferentes representações de camada **

Usando a: seleção de menu: *Estilos* -> *Add* no final da caixa de diálogo: guilabel:*Propriedades da camada*, você pode salvar quantos estilos forem necessários. Um estilo é a combinação de todas as propriedades de uma camada (como simbologia, rotulagem, diagrama, formulário de campos, ações …) como você deseja. Em seguida, basta alternar entre os estilos no menu de contexto da camada em:guilabel:`Painel de camadas 'para obter automaticamente diferentes representações dos seus dados.

Dica

Exportar simbologia vetorial

Você tem a opção de exportar a simbologia vetorial do QGIS para os arquivos das guias Google *. Kml, *. Dxf e MapInfo *. Basta abrir o menu direito do mouse da camada e clicar em: seleção de menu: *Save As …* para especificar o nome do arquivo de saída e seu formato. Na caixa de diálogo, use o menu: seleção de menu: *Exportação de simbologia* para salvar a simbologia como: seleção de menu:` Simbologia de recursos -> *ou como: seleção de menu:* Simbologia da camada de símbolo -> `. Se você usou camadas de símbolos, é recomendável usar a segunda configuração.

Apresenta renderização

The renderer is responsible for drawing a feature together with the correct symbol. Regardless layer geometry type, there are four common types of renderers: single symbol, categorized, graduated and rule-based. For point layers, there are point displacement, point cluster and heatmap renderers available while polygon layers can also be rendered with the merged features, inverted polygons and 2.5 D renderers.

Não há renderizador de cores contínuo, porque na verdade é apenas um caso especial do renderizador graduado. Os renderizadores classificados e graduados podem ser criados especificando um símbolo e uma rampa de cores - eles definirão as cores dos símbolos adequadamente. Para cada tipo de dados (pontos, linhas e polígonos), os tipos de camada de símbolo vetorial estão disponíveis. Dependendo do renderizador escolhido, a caixa de diálogo fornece diferentes seções adicionais.

Nota

Se você alterar o tipo de processador ao definir o estilo de uma camada de vetor as configurações feitas para o símbolo serão mantidas. Esteja ciente de que este procedimento só funciona para uma mudança. Se você repetir a alteração do tipo de renderizador as configurações para o símbolo irão se perder.

Renderizador de símbolo único

O | Símbolo único | :guilabel:o renderizador *Símbolo único* é usado para renderizar todos os recursos da camada usando um único símbolo definido pelo usuário. Veja:ref:`seletor de símbolos 'para mais informações sobre representação de símbolos.

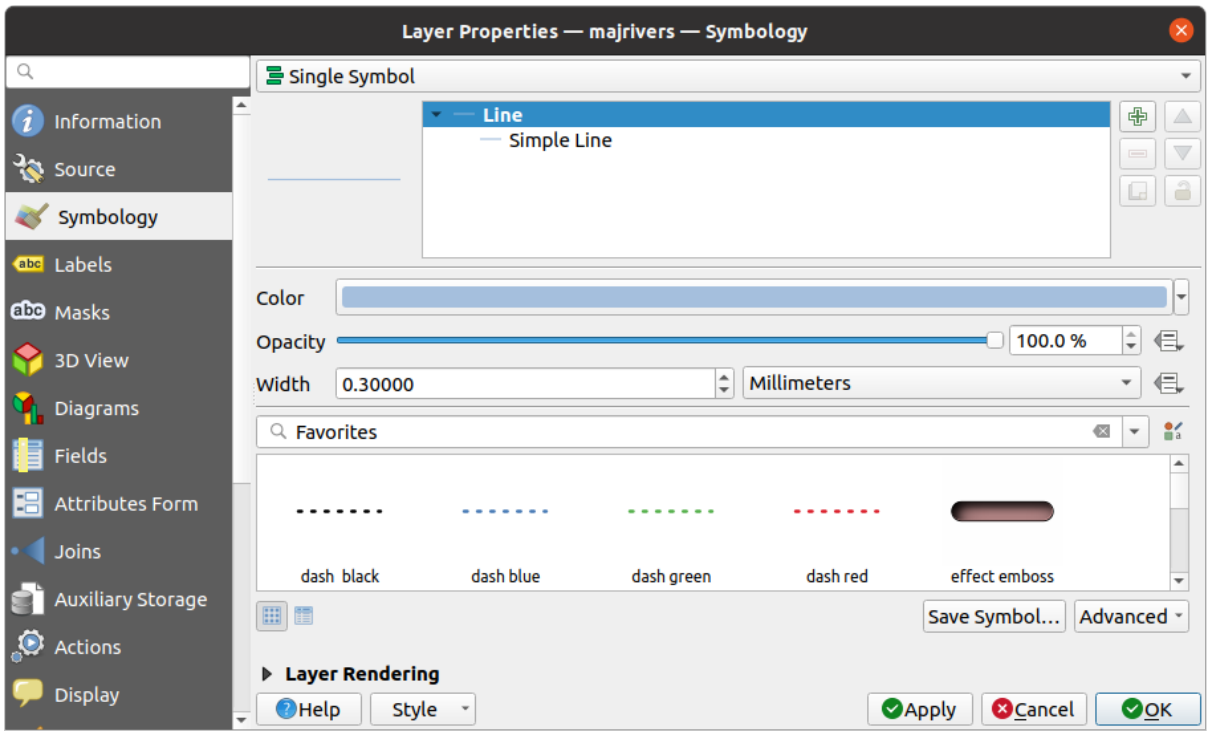


Figura12.3: Propriedades da linha de símbolo único

Renderizador sem símbolos

O | símbolo nulo | :guilabel:o renderizador *Sem símbolos* é um caso de uso especial do renderizador Símbolo único, pois aplica a mesma renderização a todos os recursos. Usando este renderizador, nenhum símbolo será desenhado para os recursos, mas ainda serão mostrados rótulos, diagramas e outras partes que não sejam de símbolos.

Ainda é possível fazer seleções na camada da tela e os recursos selecionados serão renderizados com um símbolo padrão. Os recursos editados também serão mostrados.

Este é um atalho útil para as camadas nas quais você deseja exibir apenas etiquetas ou diagramas e evita a necessidade de renderizar símbolos com preenchimento / borda totalmente transparente para conseguir isso.

Renderizador categorizado

O | símbolo categorizado | :guilabel:o renderizador categorizado é usado para renderizar os recursos de uma camada, usando um símbolo definido pelo usuário cujo aspecto reflete os valores discretos de um campo ou expressão.

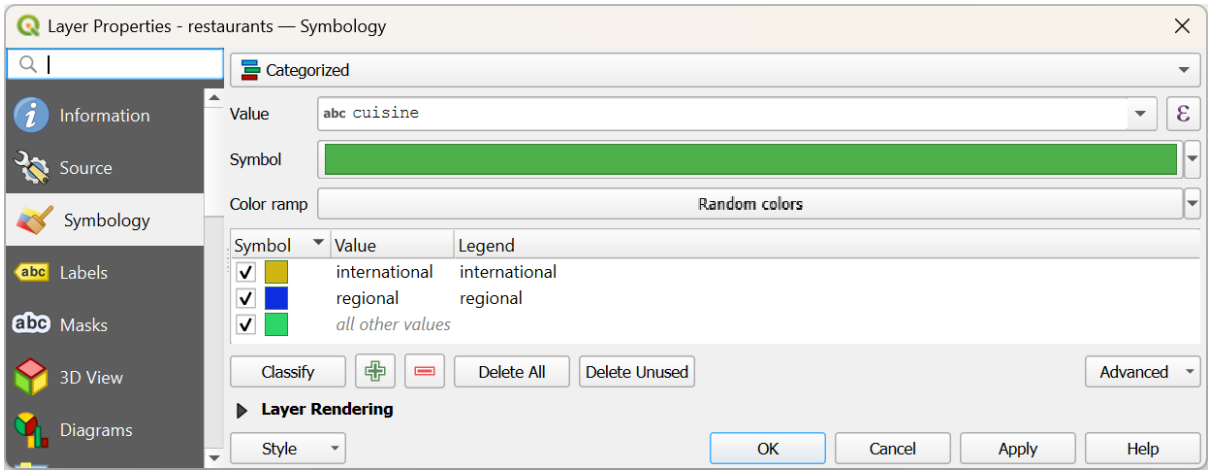


Figura12.4: Opções de simbolização categorizadas

Para usar a simbologia categorizada para uma camada:

1. Select the *Value* of classification: it can be an existing field or an *expression* you can type in the box or build using the associated ε button. Using expressions for categorizing avoids the need to create a field for symbology purposes only (eg, if your classification criteria are derived from one or more attributes).

   A expressão usada para classificar recursos pode ser de qualquer tipo; por exemplo, pode:

   - seja uma comparação. Nesse caso, o QGIS retorna os valores `` 1`` (** True) **e `` 0`` (False **). Alguns exemplos:

     ```
     myfield >= 100
     $id = @atlas_featureid
     myfield % 2 = 0
     within( $geometry, @atlas_geometry )
     ```

   - combine diferentes campos:

     ```
     concat( field_1, ' ', field_2 )
     ```

   - seja um cálculo nos campos:

     ```
     myfield % 2
     year( myfield )
     field_1 + field_2
     substr( field_1, -3 )
     ```

   - ser usado para transformar valores lineares em classes discretas, por exemplo:

     ```
     CASE WHEN x > 1000 THEN 'Big' ELSE 'Small' END
     ```

   - combine vários valores discretos em uma única categoria, por exemplo:

     ```
     CASE
     WHEN building IN ('residence', 'mobile home') THEN 'residential'
     WHEN building IN ('commercial', 'industrial') THEN 'Commercial and␣
     ↪Industrial'
     END
     ```

Dica

Embora você possa usar qualquer tipo de expressão para categorizar recursos, para algumas expressões complexas, pode ser mais simples usar:ref:*renderização baseada em regras<rule_based_rendering>*.

2. Configure o:ref:'Símbolol<symbol-selector>`, que será usado como símbolo base para todas as classes;
3. Indicate the *Color ramp*, i.e. the range of colors from which the color applied to each symbol is selected.

   Além das opções comuns do:ref:*ferramenta de rampa de cores<color_ramp_widget>*, você pode aplicar um | desmarcado | *Random Color Ramp* para as categorias. Você pode clicar na entrada: guilabel: *Aleatório cores aleatórias* para gerar novamente um novo conjunto de cores aleatórias, se você não estiver satisfeito.
4. Em seguida, clique no botão:guilabel:*Classificar* para criar classes a partir dos valores distintos do campo ou expressão fornecido.
5. For each class, you can edit the *Legend* column to a more meaningful label (used in the *Layers* panel and the print layout).

   Nota

   Keep in mind that some values may use widgets that do not display the actual value stored in the field. For example, a checkbox widget may store `1` and `0` for checked and unchecked states, while displaying `True` and `False` labels. In this case, to categorize features based on the checkbox state, you need to use the stored values (`1` and `0`) in the expression. QGIS will automatically use the display value for the legend column.

6. *Aplique as alterações se o: ref:* atualização ao vivo <layer_styling_panel>`não estiver em uso e cada recurso na tela do mapa for renderizado com o símbolo de sua classe.

   Por padrão, o QGIS anexa uma classe:guilabel:*todos os outros valores* à lista. Enquanto vazia no início, essa classe é usada como uma classe padrão para qualquer recurso que não se enquadre nas outras classes (por exemplo, quando você cria recursos com novos valores para o campo / expressão de classificação).

Outros ajustes podem ser feitos na classificação padrão:

- You can Add new categories, Remove selected categories, *Delete All* of them or *Delete Unused* categories.
- Uma classe pode ser desativada desmarcando a caixa de seleção à esquerda do nome da classe; os recursos correspondentes estão ocultos no mapa.
- Arraste e solte as linhas para reordenar as classes
- Para alterar o símbolo, o valor ou a legenda de uma classe, clique duas vezes no item.

Clicar com o botão direito do mouse sobre os item (s) selecionados mostra um menu contextual para:

- *Copiar símbolo* e: guilabel:` Colar símbolo`, uma maneira conveniente de aplicar a representação do item a outras pessoas
- *Alterar Color* … do símbolo (s) selecionado
- *Alterar opacidade* … do símbolo (s) selecionado
- *Alterar unidade de saída* … do símbolo (s) selecionado
- *Alterar largura* … do símbolo (s) de linha selecionado
- *Alterar tamanho* … do símbolo (s) de ponto selecionado
- *Alterar ângulo* … do símbolo (s) de ponto selecionado
- *Mesclar categorias*: agrupa várias categorias selecionadas em uma única. Isso permite um estilo mais simples de camadas com um grande número de categorias, onde pode ser possível agrupar várias categorias distintas em um conjunto menor e mais gerenciável de categorias que se aplicam a vários valores.

> **Dica**
>
> Como o símbolo mantido para as categorias mescladas é o da categoria selecionada mais no topo da lista, convém mover a categoria cujo símbolo você deseja reutilizar para o topo antes de mesclar.

- *Unmerge Categories* que foram mescladas anteriormente

The created classes also appear in a tree hierarchy in the *Layers* panel. Double-click an entry in the map legend to edit the assigned symbol. Right-click and you will get some *more options*.

O menu:guilabel:*Avançado* dá acesso a opções para acelerar a classificação ou ajustar a renderização de símbolos:

- *Corresponde aos símbolos salvos*: Usando a: ref:` biblioteca de símbolos <vector_style_manager>`, atribui a cada categoria um símbolo cujo nome representa o valor de classificação da categoria
- *Corresponde aos símbolos do arquivo …*: Fornece um arquivo com símbolos, atribui a cada categoria um símbolo cujo nome representa o valor de classificação da categoria
- *Níveis de símbolos …* para definir a ordem de renderização dos símbolos.

Renderer graduado

O símbolo | graduado | :guilabel:o renderizador *Graduado* é usado para renderizar todos os recursos de uma camada, usando um símbolo definido pelo usuário cuja cor ou tamanho reflete a atribuição do atributo de um recurso selecionado a uma classe.

Como o Renderizador categorizado, o Renderizador graduado permite definir a rotação e a escala de tamanho das colunas especificadas.

Além disso, de forma análoga ao Renderizador Categorizado, permite selecionar:

- The *Value* of classification: it can be an existing field or an *expression* you can type in the box or build using the associated ε button. Using expressions for graduating avoids the need to create a field for symbology purposes only (eg, if your classification criteria are derived from one or more attributes).
- O símbolo (usando a caixa de diálogo Seletor de símbolos)
- O formato da legenda e a precisão
- O método a ser usado para alterar o símbolo: cor ou tamanho
- As cores (usando a lista Rampa de cores) se o método de cores estiver selecionado
- O tamanho (usando o domínio de tamanho e sua unidade)

Em seguida, você pode usar a guia Histograma, que mostra um histograma interativo dos valores do campo ou expressão atribuído. As quebras de classe podem ser movidas ou adicionadas usando o widget de histograma.

> **Nota**
>
> Você pode usar o painel Resumo Estatístico para obter mais informações sobre sua camada vetorial. Veja:ref:"resumo estatístico".

De volta à guia Classes, você pode especificar o número de classes e também o modo para classificar os recursos dentro das classes (usando a lista Modo). Os modos disponíveis são:

- Equal Count (Quantile): each class will have the same number of elements (the idea of a boxplot).
- Intervalo igual: cada classe terá o mesmo tamanho (por exemplo, com os valores de 1 a 16 e quatro classes, cada classe terá um tamanho de quatro).
- Fixed Interval: each class will have a fixed range of values (e.g. with the values from 1 to 16 and an interval size of 4, the classes will be 1-4, 5-8, 9-12 and 13-16).

- Logarithmic scale: suitable for data with a wide range of values. Narrow classes for low values and wide classes for large values (e.g. for decimal numbers with range [0..100] and two classes, the first class will be from 0 to 10 and the second class from 10 to 100).
- Natural Breaks (Jenks): the variance within each class is minimized while the variance between classes is maximized.
- Pretty Breaks: computes a sequence of about n+1 equally spaced nice values which cover the range of the values in x. The values are chosen so that they are 1, 2 or 5 times a power of 10. (based on pretty from the R statistical environment https://www.rdocumentation.org/packages/base/topics/pretty).
- Desvio padrão: as classes são construídas dependendo do desvio padrão dos valores.

A caixa de listagem na parte central da guia:guilabel:*Simbologia* lista as classes, juntamente com seus intervalos, rótulos e símbolos que serão renderizados.

Clique no botão ** Classificar ** para criar classes usando o modo escolhido. Cada classe pode ser desativada desmarcando a caixa de seleção à esquerda do nome da classe.

Para alterar o símbolo, valor e/ou rótulo da classe, basta clicar duas vezes no item que você deseja alterar.

Clicar com o botão direito do mouse sobre os item (s) selecionados mostra um menu contextual para:

- *Copiar símbolo* e: guilabel:` Colar símbolo`, uma maneira conveniente de aplicar a representação do item a outras pessoas
- *Alterar Color …* do símbolo (s) selecionado
- *Alterar opacidade …* do símbolo (s) selecionado
- *Alterar unidade de saída …* do símbolo (s) selecionado
- *Alterar largura …* do símbolo (s) de linha selecionado
- *Alterar tamanho …* do símbolo (s) de ponto selecionado
- *Alterar ângulo …* do símbolo (s) de ponto selecionado

The example in Fig.12.5 shows the graduated rendering dialog for the major_rivers layer of the QGIS sample dataset.

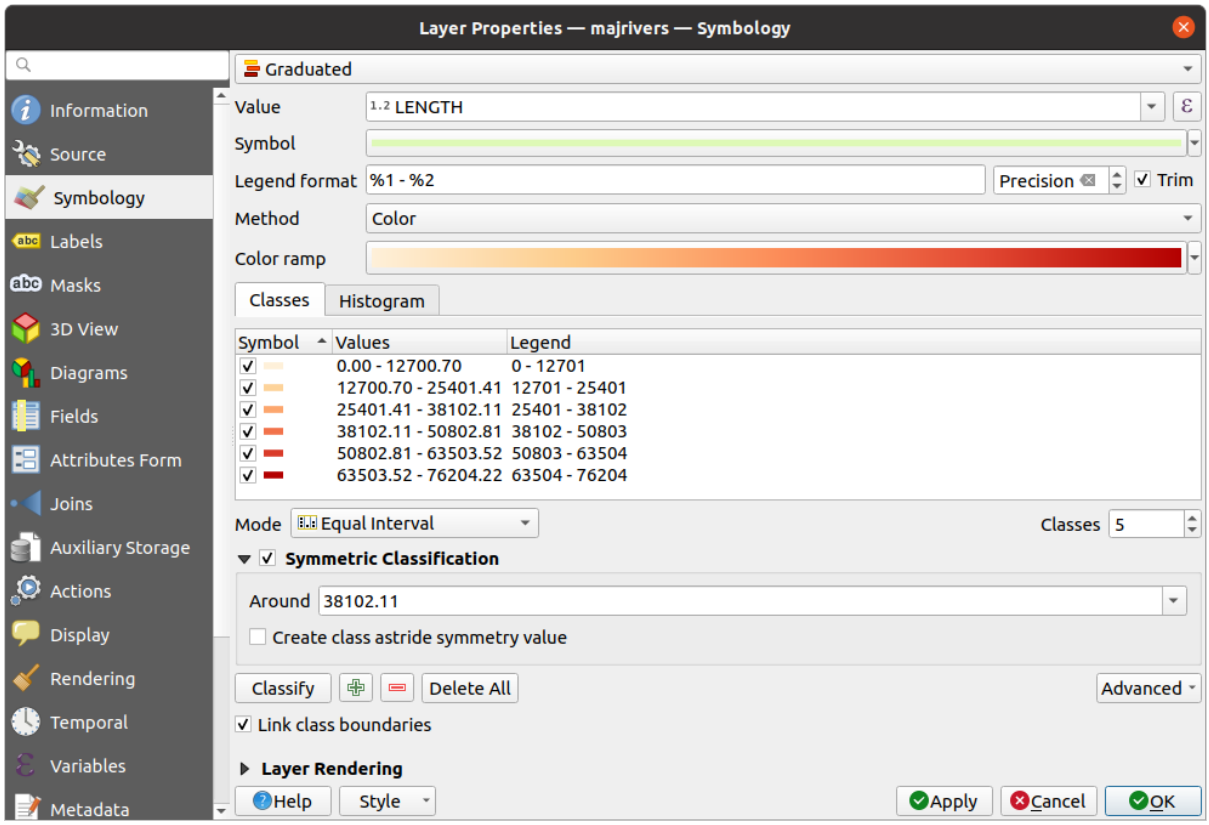


Figura12.5: Opções de simbolização graduadas

The created classes also appear in a tree hierarchy in the *Layers* panel. Double-click an entry in the map legend to edit the assigned symbol. Right-click and you will get some *more options*.

Símbolo Proporcional e Análise Multivariada

Símbolo proporcional e análise multivariada não são tipos de renderização disponíveis na lista suspensa Renderização de simbologia. No entanto, com as opções:ref:*substituição definida por dados<data_defined>* aplicada a qualquer uma das opções de renderização anteriores, o QGIS permite exibir seus dados de ponto e linha com essa representação.

** Criando símbolo proporcional **

Para aplicar uma renderização proporcional:

1. Primeiro aplique à camada o:ref:*renderizador de símbolo único<single_symbol_renderer>*.
2. Em seguida, defina o símbolo para aplicar aos recursos.
3. Selecione o item no nível superior da árvore de símbolos e use o | dadosDefinidos | Substituição definida por dados: ref:` botão<data_defined> *ao lado da opção: guilabel:`Tamanho* (para camada de pontos) ou:guilabel:*Largura* (para camada de linha).
4. Selecione um campo ou insira uma expressão e, para cada recurso, o QGIS aplicará o valor de saída à propriedade e redimensionará proporcionalmente o símbolo na tela do mapa.

   Se necessário, use a opção:guilabel:*Assistente de tamanho …* da | dadosDefinidos | menu para aplicar alguma transformação (exponencial, flannery …) ao redimensionamento do tamanho do símbolo (consulte:ref:*assistente_dados_definidos* para obter mais detalhes).

Você pode optar por exibir os símbolos proporcionais na:ref:*Camadas do painel* e na: ref:` item da legenda do layout de impressão *: desdobre a lista suspensa: guilabel:* Avançado` na parte inferior da caixa de diálogo principal do :guilabel:aba *Simbologia* e selecione ** Legenda do tamanho definido pelos dados … ** para configurar os itens da legenda (consulte: ref:` legenda_tamanho_multado _dados` para obter detalhes).

** Criando análise multivariada **

Uma renderização de análise multivariada ajuda a avaliar o relacionamento entre duas ou mais variáveis, por exemplo, uma pode ser representada por uma rampa de cores enquanto a outra é representada por um tamanho.

A maneira mais simples de criar análises multivariadas no QGIS é:

1. Primeiro, aplique uma renderização categorizada ou graduada em uma camada, usando o mesmo tipo de símbolo para todas as classes.
2. Em seguida, aplique uma simbologia proporcional nas classes:
    1. Clique no botão:guilabel:*Mudança* acima do quadro de classificação: você obtém a caixa de diálogo: ref:` seletor de símbolos`.
    2. Redimensione novamente o tamanho ou a largura da camada de símbolo usando o | dadosDefinidos | *dados definidos substituem o ferramenta* como visto acima.

Como o símbolo proporcional, a simbologia escalada pode ser adicionada à árvore de camadas, sobre os símbolos das classes categorizadas ou graduadas, usando o recurso:ref:*legenda do tamanho definido dos dados<data_defined_size_legend>*. E ambas as representações também estão disponíveis no item da legenda do layout de impressão.

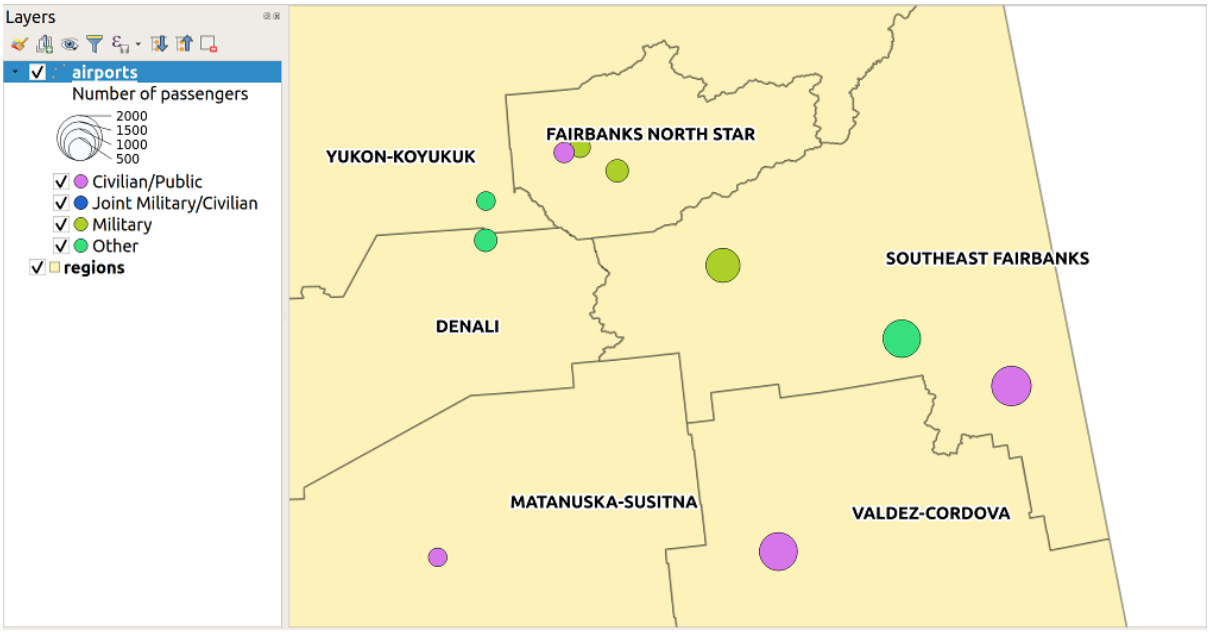


Figura12.6: Exemplo multivariado com legenda de tamanho dimensionado

Renderizador baseado em regras

Rules are QGIS *expressions* used to discriminate features according to their attributes or properties in order to apply specific rendering settings to them. Rules can be nested, and features belong to a class if they belong to all the upper nesting level(s).

The *Rule-based* renderer is thus designed to render all the features from a layer, using symbols whose aspect reflects the assignment of a selected feature to a fine-grained class.

Para criar uma regra:

1. Activate an existing row by double-clicking it (by default, QGIS adds a symbol without a rule when the rendering mode is enabled) or click the Edit rule or Add rule button.

2. Na caixa de diálogo:guilabel:*Editar regra* que é aberta, você pode definir um rótulo para ajudá-lo a identificar cada regra. Este é o rótulo que será exibido em:guilabel:`Painel de camadas 'e também na legenda do compositor de impressão.
3. Insira manualmente uma expressão na caixa de texto ao lado de | botão de opção Ativado | :guilabel:opção *Filtro* ou pressione a expressão | ao lado dele para abrir a caixa de diálogo do construtor de cadeias de expressão.
4. Use as funções fornecidas e os atributos da camada para criar uma expressão: ref:<vector_expressions> *para filtrar os recursos que você deseja recuperar. Pressione o botão:guilabel:`Test* para verificar o resultado da consulta.
5. Você pode inserir um rótulo mais longo para concluir a descrição da regra.
6. Você pode usar a caixa de seleção | :guilabel:opção ` Escala de escala` para definir escalas nas quais a regra deve ser aplicada.
7. Por fim, configure o símbolo:ref:*para usar<symbol-selector>* para esses recursos.
8. E pressione:guilabel:*OK*.

Uma nova linha que resume a regra é adicionada à caixa de diálogo Propriedades da camada. Você pode criar quantas regras forem necessárias, seguindo as etapas acima ou copiar colando uma regra existente. Arraste e solte as regras uma sobre a outra para aninhar e refinar os recursos da regra superior nas subclasses.

The rule-based renderer can be combined with categorized or graduated renderers. Selecting a rule, you can organize its features in subclasses using the *Refine selected rules* drop-down menu. Refined classes appear like sub-items of the rule, in a tree hierarchy and like their parent, you can set the symbology and the rule of each class. Automated rule refinement can be based on:

- **scales**: given a list of scales, this option creates a set of classes to which the different user-defined scale ranges apply. Each new scale-based class can have its own symbology and expression of definition. This can e.g. be a convenient way to display the same features with various symbols at different scales, or display only a set of features depending on the scale (e.g. local airports at large scale vs international airports at small scale).
- **categories**: applies a *categorized renderer* to the features falling in the selected rule.
- or **ranges**: applies a *graduated renderer* to the features falling in the selected rule.

Refined classes appear like sub-items of the rule, in a tree hierarchy and like above, you can set symbology of each class. Symbols of the nested rules are stacked on top of each other so be careful in choosing them. It is also possible to uncheck *Symbols* in the *Edit rule* dialog to avoid rendering a particular symbol in the stack.

Na caixa de diálogo:guilabel:*Editar regra*, você pode evitar escrever todas as regras e usar o | botão de rádio desativado | :guilabel:opção *Outra* para capturar todos os recursos que não correspondem a nenhuma das outras regras, no mesmo nível. Isso também pode ser conseguido escrevendo `Outra` na coluna * Rule * da seção: seleção de menus: *Propriedades da camada -> Simbologia-> Baseado em regras*.

Clicar com o botão direito do mouse sobre os item (s) selecionados mostra um menu contextual para:

- *Copiar* e: guilabel:` Colar`, uma maneira conveniente de criar novos item com base nos itens existentes
- *Copiar símbolo* e: guilabel:` Colar símbolo`, uma maneira conveniente de aplicar a representação do item a outras pessoas
- *Alterar Color* … do símbolo (s) selecionado
- *Alterar opacidade* … do símbolo (s) selecionado
- *Alterar unidade de saída* … do símbolo (s) selecionado
- *Alterar largura* … do símbolo (s) de linha selecionado
- *Alterar tamanho* … do símbolo (s) de ponto selecionado
- *Alterar ângulo* … do símbolo (s) de ponto selecionado
- *Refine Current Rule*: open a submenu that allows to refine the current rule with **scales**, **categories** or **Ranges**. Same as selecting the *corresponding menu* at the bottom of the dialog.

Unchecking a row in the rule-based renderer dialog hides in the map canvas the features of the specific rule and the nested ones.

The created rules also appear in a tree hierarchy in the map legend. Double-click an entry in the map legend to edit the assigned symbol. Right-click and you will get some *more options*.

The example in Fig.12.7 shows the rule-based rendering dialog for the rivers layer of the QGIS sample dataset.

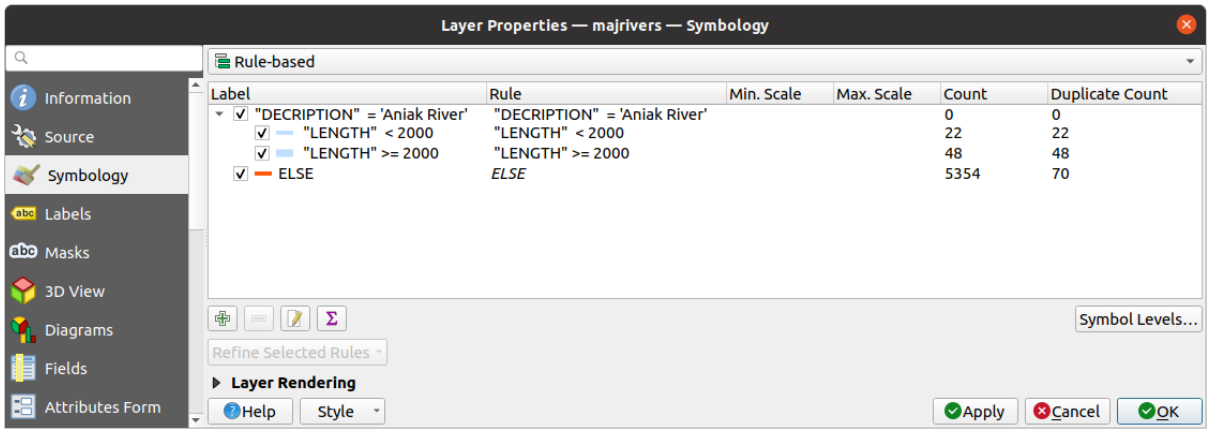


Figura12.7: Opções de simbolização baseadas em regras

Renderizador de deslocamento pontual

The *Point Displacement* renderer takes the point features falling in a given distance tolerance from each other and places their symbols around their barycenter, following different placement methods. This can be a convenient way to visualize all the features of a point layer, even if they have the same location (e.g. amenities in a building).

To configure a point displacement renderer, you have to:

1. Set the *Center symbol*: how the virtual point at the center will look like
2. Select the *Renderer* type: how you want to classify features in the layer (single, categorized, rule-based…)
3. Press the *Renderer Settings…* button to configure features' symbology according to the selected renderer
4. Indicate the *Distance* tolerance in which close features are considered overlapping and then displaced over the same virtual point. Supports common symbol units.
5. Configure the *Placement methods*:
   - ** Toque **: coloca todos os recursos em um círculo cujo raio depende do número de recursos a serem exibidos.
   - ** Anéis concêntricos **: usa um conjunto de círculos concêntricos para mostrar os recursos.
   - ** Grade **: gera uma grade regular com um símbolo de ponto em cada interseção.
6. Displaced symbols are placed on the *Displacement lines*. While the minimal spacing of the displacement lines depends on the point symbols renderer, you can still customize some of their settings such as the *Stroke width*, *Stroke color* and *Size adjustment* (e.g., to add more spacing between the rendered points).
7. Use the *Labels* group options to perform points labeling: the labels are placed near the displaced symbol, and not at the feature real position.
   1. Select the *Label attribute*: a field of the layer to use for labeling
   2. Indicate the *Label font* properties and size
   3. Pick a *Label color*

4. Set a *Label distance factor*: for each point feature, offsets the label from the symbol center proportionally to the symbol's diagonal size.

5. Turn on *Use scale dependent labeling* if you want to display labels only on scales larger than a given *Minimum map scale*.

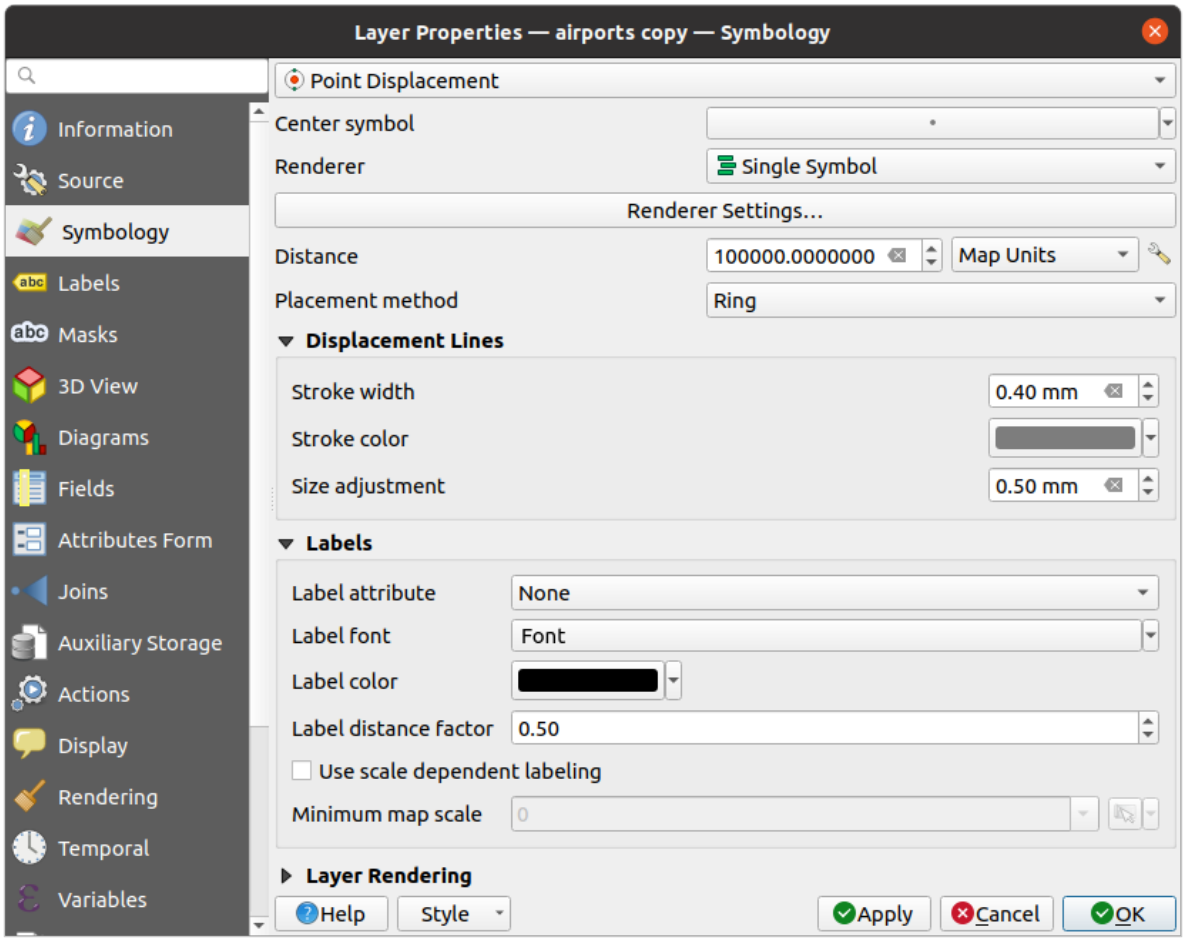


Figura12.8: Diálogo deslocamento de ponto

Nota

O renderizador de deslocamento de ponto não altera a geometria do recurso, o que significa que os pontos não são movidos de sua posição. Eles ainda estão localizados em seu local inicial. As alterações são apenas visuais, para fins de renderização. Em vez disso, use o algoritmo Em processamento: ref: *qgispointsdisplacement* se desejar criar recursos deslocados.

Processador de cluster de pontos

Unlike the *Point Displacement* renderer which blows up nearest or overlaid point features placement, the *Point Cluster* renderer groups nearby points into a single rendered marker symbol. Points that fall within a specified distance from each others are merged into a single symbol. Points aggregation is made based on the closest group being formed, rather than just assigning them the first group within the search distance.

Na caixa de diálogo principal, você pode:

1. Set the symbol to represent the point cluster in the *Cluster symbol*; the default rendering displays the number of aggregated features thanks to the `@cluster_size` *variable* on Font marker symbol layer.
2. Select the *Renderer* type, i.e. how you want to classify features in the layer (single, categorized, rule-based…)
3. Press the *Renderer Settings…* button to configure features' symbology as usual. Note that this symbology is only visible on features that are not clustered, the *Cluster symbol* being applied otherwise. Also, when all the point features in a cluster belong to the same rendering class, and thus would be applied the same color, that color represents the `@cluster_color` variable of the cluster.
4. Indicate the maximal *Distance* to consider for clustering features. Supports common symbol units.

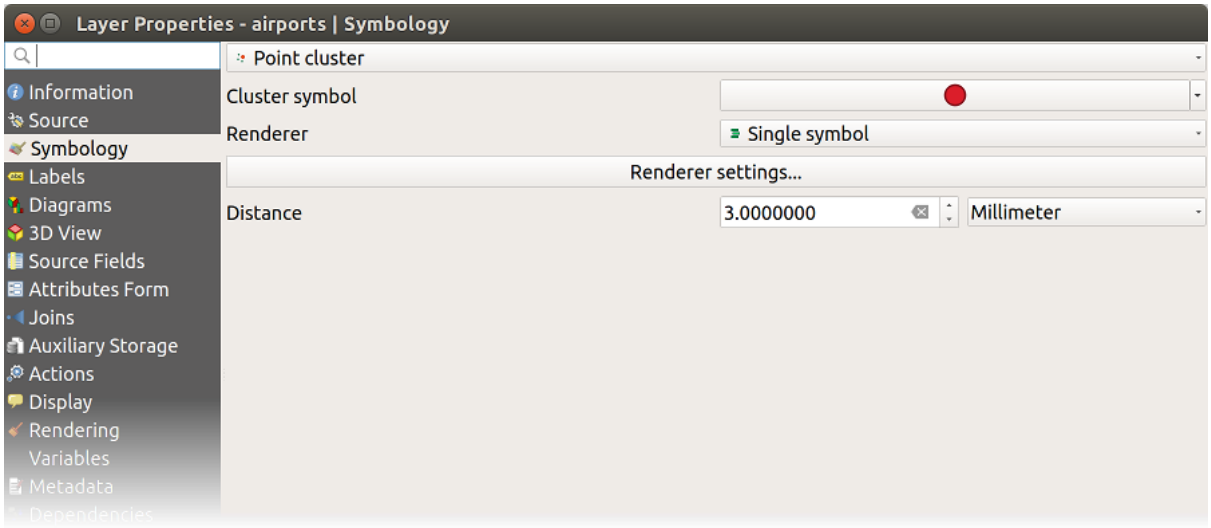


Figura12.9: Caixa de diálogo Cluster de pontos

Nota

O renderizador de cluster de pontos não altera a geometria do recurso, o que significa que os pontos não são movidos de sua posição. Eles ainda estão localizados em seu local inicial. As alterações são apenas visuais, para fins de renderização. Em vez disso, use o algoritmo em processamento:ref:*qgiskmeansclustering* ou: ref:` qgisdbscanclustering` se desejar criar recursos baseados em cluster.

Merged Features Renderer

The *Merged Features* renderer allows area and line features to be “dissolved” into a single object prior to rendering to ensure that complex symbols or overlapping features are represented by a uniform and contiguous cartographic symbol.

Renderizador de polígono invertido

O | símbolo invertido | :guilabel:O renderizador *Polígono invertido* permite ao usuário definir um símbolo a ser preenchido fora dos polígonos da camada. Como acima, você pode selecionar sub-remetentes, ou seja, renderizador de símbolo único, graduado, categorizado, baseado em regras ou 2.5D.

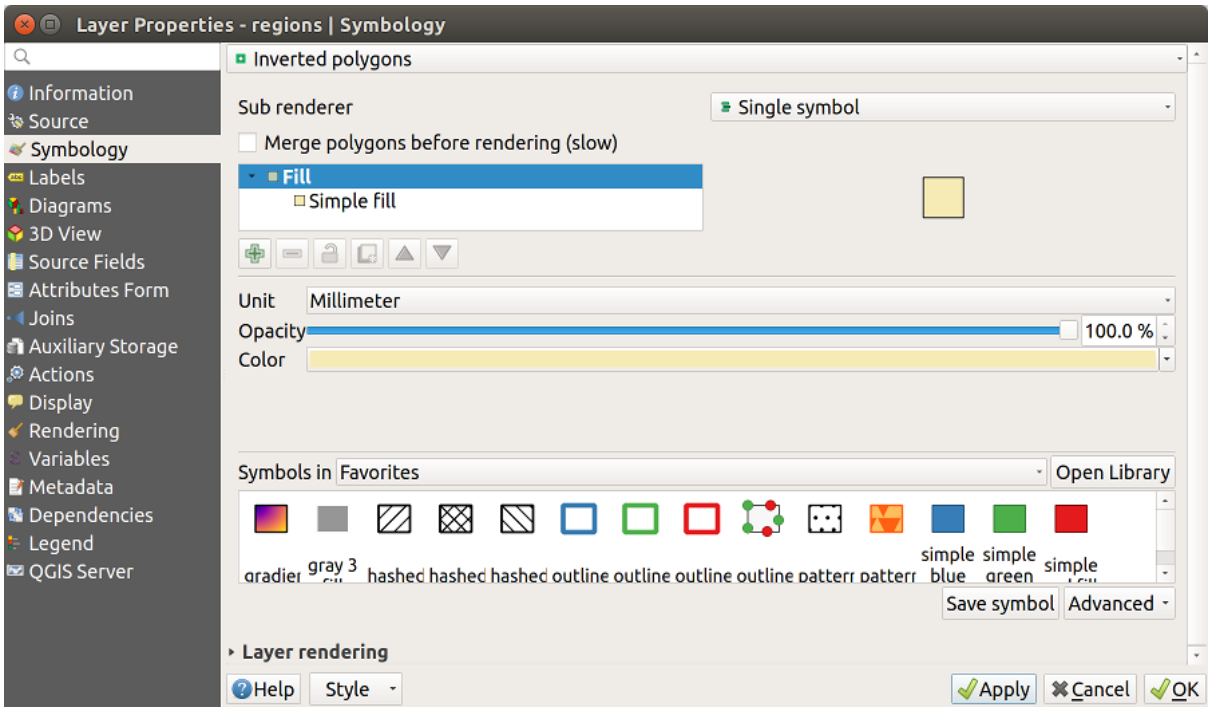


Figura12.10: Caixa de diálogo Polígono invertido

Renderizador de mapa de calor

With the *Heatmap* renderer you can create live dynamic heatmaps for (multi)point layers. You can specify the heatmap *Radius* in millimeters, points, pixels, map units or inches, choose and edit a *Color ramp* for the heatmap style and use a slider for selecting a trade-off between render speed and quality. You can also define a *Maximum value* limit and *Weight points by* using a field or an expression.

Use Data defined override to dynamically control *Radius* and *Maximum value* based on the attributes of your data. For example, the radius of a heatmap point could be determined by its population attribute, or the maximum value could be based on a temporal range.

When adding or removing a feature the heatmap renderer updates the heatmap style automatically. The *Color ramp* will be shown as a legend bar and in the *Legend settings* you can set the *Labels* for the *Maximum* and *Minimum* values. You can also change the orientation and direction of the legend in the *Layout*.

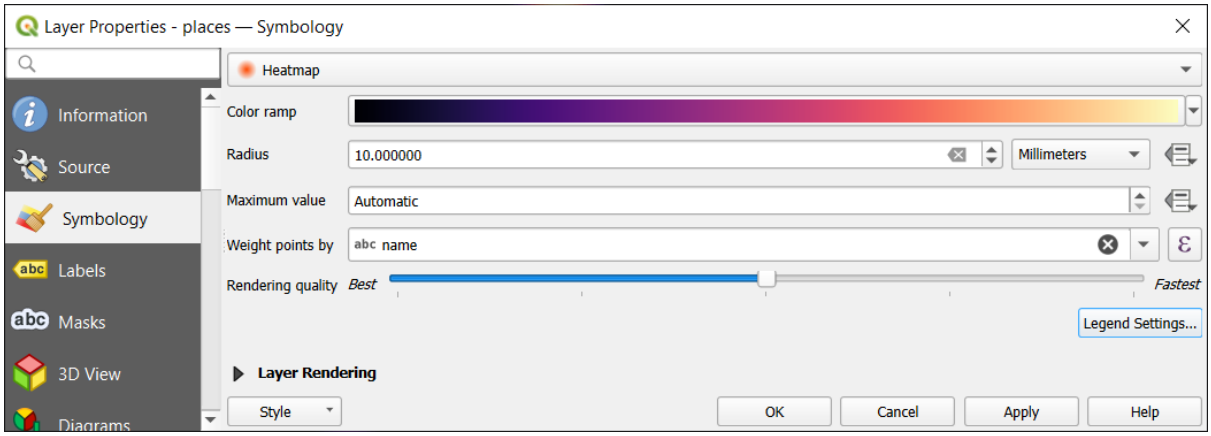


Figura12.11: Caixa de diálogo Mapa de calor

Renderizador 2.5D

Usando o | 25dSymbol | :guilabel:renderizador *2.5D* é possível criar um efeito 2.5D nos recursos da sua camada. Você começa escolhendo um valor: guilabel: *Altura* (em unidades do mapa). Para isso, você pode usar um valor fixo, um dos campos da sua camada ou uma expressão. Você também precisa escolher um:guilabel:*Ângulo* (em graus) para recriar a posição do visualizador (0 | graus | significa oeste, crescendo no sentido anti-horário). Use opções de configuração avançadas para definir:guilabel:*Cor do telhado* e: guilabel:` Cor da parede`. Se você deseja simular radiação solar nas paredes dos recursos, marque a caixa de seleção | :guilabel:opção Paredes sombreadas com base no aspecto. Você também pode simular uma sombra definindo a:guilabel:*Cor* e: guilabel:` Size` (em unidades do mapa).

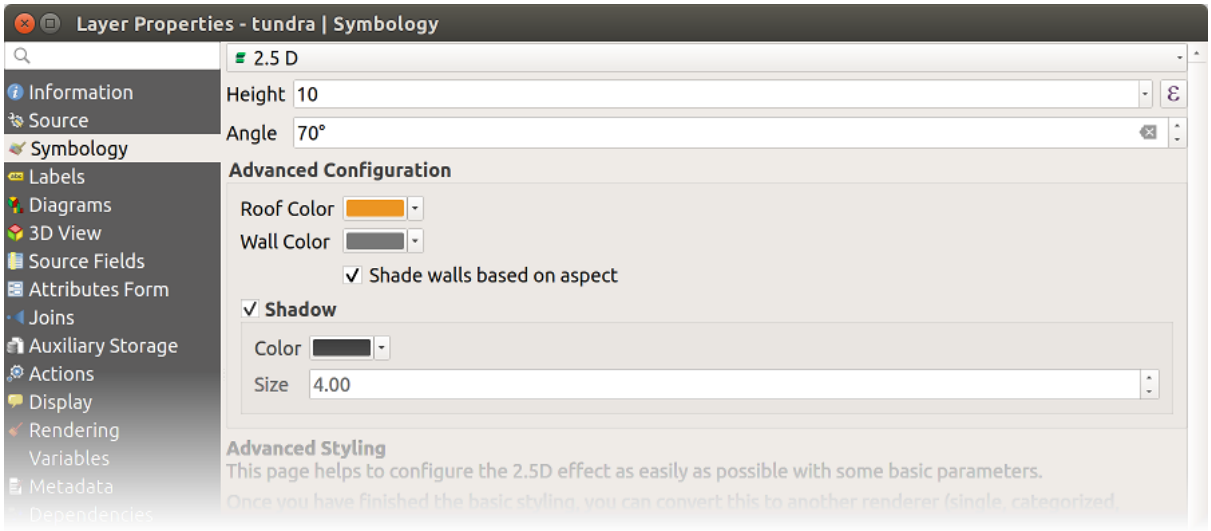


Figura12.12: Caixa de diálogo 2.5D

Dica

** Usando o efeito 2.5D com outros renderizadores **

Depois de concluir a configuração do estilo básico no renderizador 2.5D, você poderá convertê-lo em outro renderizador (único, categorizado e graduado). Os efeitos 2.5D serão mantidos e todas as outras opções específicas do renderizador estarão disponíveis para você ajustá-los (desta forma, você pode ter, por exemplo, símbolos categorizados com uma boa representação 2.5D ou adicionar um estilo extra aos seus símbolos 2.5D). Para garantir

que a sombra e o “edifício” em si não interfiram com outros recursos próximos, pode ser necessário ativar os Níveis de símbolos (: menuelection: *Advanced* -> *Symbol levels* …). Os valores de altura e ângulo de 2.5D são salvos nas variáveis da camada, para que você possa editá-lo posteriormente na guia variáveis da caixa de diálogo de propriedades da camada.

Embedded Renderer

The *Embedded Symbols* renderer allows to display the ‘native’ symbology of a provided datasource. This is mostly the case with `KML` and `TAB` datasets that have predefined symbology.

Renderização da camada

Na guia da Simbologia, você também pode definir algumas opções que atuam invariavelmente em todos os recursos da camada:

- *Opacity* : You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your vector layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Modo de mesclagem* nos níveis: guilabel:*Camada* e:guilabel:*Característica*: Você pode obter efeitos especiais de renderização com essas ferramentas que antes só conhecia em programas gráficos. Os pixels de suas camadas de sobreposição e de sobreposição são misturados através das configurações descritas em:ref:*modos de mesclagem*.
- Aplique:ref:*efeitos de pintura<draw_effects>* em todos os recursos da camada com o botão:guilabel: *Efeitos de desenho*.
- *Controlar a ordem de renderização do recurso ‘permite que você, usando atributos de recursos, defina a ordem z na qual eles serão renderizados. Ative a caixa de seleção e clique no | classificar | botão ao lado. Você então recebe a caixa de diálogo:guilabel:`Definir ordem* na qual você:

  1. Escolha um campo ou crie uma expressão para aplicar aos recursos da camada.
  2. Defina em que ordem os recursos buscados devem ser classificados, ou seja, se você escolher a ordem ** Ascendente **, os recursos com valor mais baixo serão renderizados naqueles com valor mais alto.
  3. Defina quando os recursos que retornam valor NULO devem ser renderizados: ** primeiro ** (inferior) ou ** último ** (superior).
  4. Repita as etapas acima quantas vezes você desejar.

  A primeira regra é aplicada a todos os recursos da camada, ordenando-os de acordo com o valor retornado. Em seguida, dentro de cada grupo de recursos com o mesmo valor (incluindo aqueles com valor NULO) e, portanto, o mesmo nível z, a próxima regra é aplicada para classificá-los. E assim por diante…

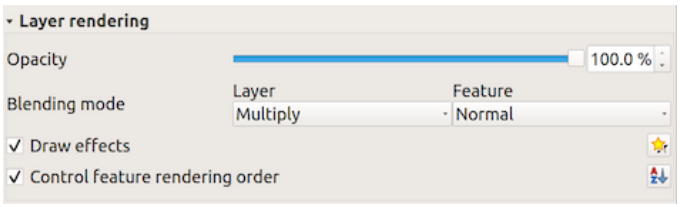


Figura12.13: Opções de renderização de camada

Outros ajustes

Níveis de símbolo

Para renderizadores que permitem camadas de símbolos empilhados (apenas o mapa de calor não), existe uma opção para controlar a ordem de renderização dos níveis de cada símbolo.

Para a maioria dos renderizadores, você pode acessar a opção Níveis de símbolos clicando no botão:guilabel:*Avançado* abaixo da lista de símbolos salvos e escolhendo: guilabel:` Níveis de símbolos`. Para o:ref:*renderização baseada em regras*, a opção está diretamente disponível através do botão:guilabel:*Símbolos Níveis … `, enquanto para o renderizador ref:`deslocamento de ponto* o mesmo botão está dentro da caixa de diálogo:guilabel:*Configurações de renderização*

Para ativar os níveis de símbolos, marque a caixa de seleção | *Ativar níveis de símbolo*. Cada linha exibirá uma pequena amostra do símbolo combinado, seu rótulo e a camada de símbolos individuais divididos em colunas com um número ao lado. Os números representam o nível da ordem de renderização no qual a camada de símbolo será desenhada. Os níveis de valores mais baixos são desenhados primeiro, permanecendo na parte inferior, enquanto os valores mais altos são desenhados por último, sobre os outros.

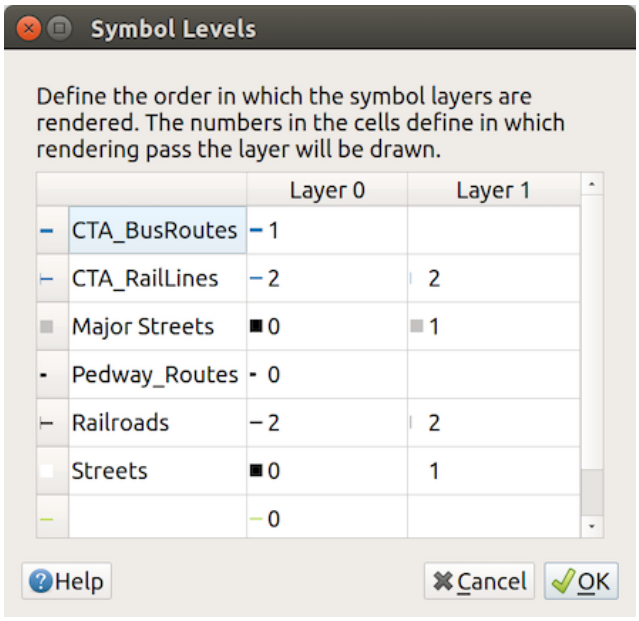


Figura12.14: Caixa de diálogo Níveis de símbolo

Nota

Se os níveis de símbolos estiverem desativados, os símbolos completos serão desenhados de acordo com a respectiva ordem de recursos. Símbolos sobrepostos simplesmente serão ofuscados para os outros abaixo. Além disso, símbolos semelhantes não “se fundem” entre si.

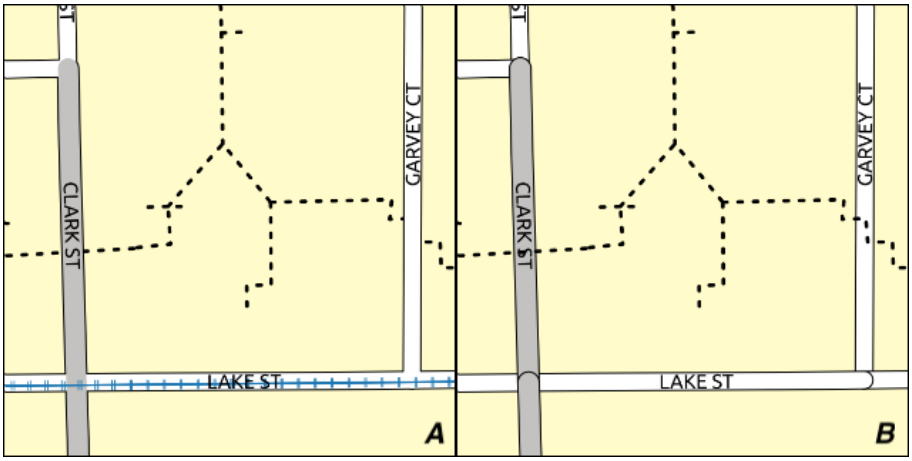


Figura12.15: Diferença nos níveis de símbolo ativado (A) e desativado (B)

Legenda de tamanho definido por dados

Quando uma camada é renderizada com o símbolo:ref:*proporcional ou multivariado<proportional_symbols>* ou quando um:ref:*diagrama de tamanho em escala<diagram_size>* é aplicado à camada, você pode permitir a exibição dos símbolos em escala em ambos:ref:*Camadas do painel<label_legend>* e: ref: *legenda do layout de impressão<layout_legend_item>*.

Para ativar a caixa de diálogo:guilabel:*Legenda do tamanho definido por dados* para renderizar simbologia, selecione a opção de epônimo no botão: guilabel:` Avançado` abaixo da lista de símbolos salvos. Para diagramas, a opção está disponível na guia:guilabel:guia Legenda. A caixa de diálogo fornece as seguintes opções para:

- selecione o tipo de legenda: | botão de opção Ativado | *Legenda não ativada*, | botão de opção Desativado | *Itens de legenda separados* e |botão de opção Desativado | *legenda recolhida*. Para a última opção, você pode selecionar se os itens da legenda estão alinhados na ** Parte inferior ** ou no ** Centro **;
- preview the *symbol to use* for legend representation;
- insira o título na legenda;
- redimensionar as classes a serem usadas: por padrão, o QGIS fornece uma legenda de cinco classes (com base em pausas bonitas naturais), mas você pode aplicar sua própria classificação usando a caixa de seleção | : guilabel: opção *Classes de tamanho manual*. Use o | assinarPlus | e | assinar menos | para definir seus valores e rótulos de classes personalizadas.
- For collapsed legend, it's possible to:
  - *Align symbols* in the center or the bottom
  - configure the horizontal leader *Line symbol* from the symbol to the corresponding legend text.

A preview of the legend is displayed in the right panel of the dialog and updated as you set the parameters.

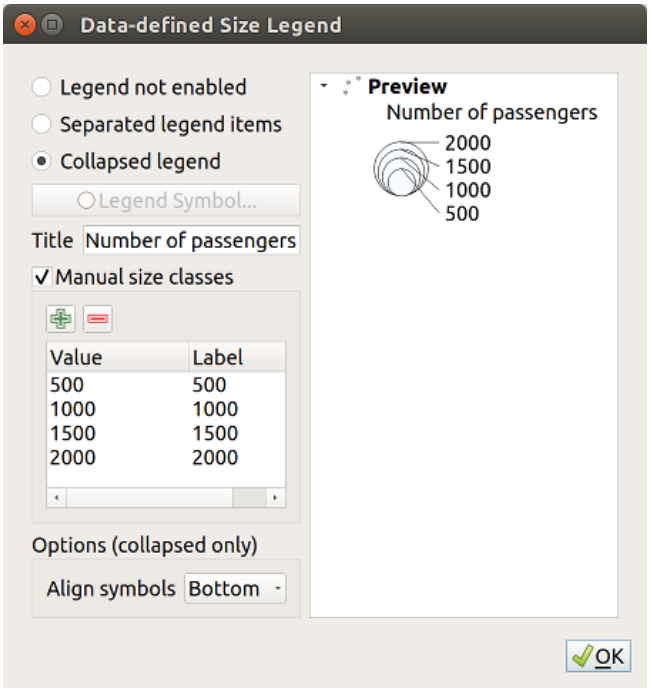


Figura12.16: Configurando legenda em tamanho dimensionado

Nota

Atualmente, a legenda do tamanho definido pelos dados para a simbologia de camada só pode ser aplicada à camada de ponto usando uma simbologia única, categorizada ou graduada.

Animation settings

To allow any symbol to become an *animated symbol*, you can utilize *Animation settings* panel. In this panel, you can enable animation for the symbol and set a specific frame rate for the symbol's redrawing.

1. Start by going to the top symbol level and select *Advanced* menu in the bottom right of the dialog
2. Find *Animation settings* option
3. Check *Is Animated* to enable animation for the symbol
4. Configure the *Frame rate*, i.e. how fast the animation would be played
5. You can now use `@symbol_frame` variable in any sub-symbol data defined property in order to animate that property.

For example, setting the symbol's rotation to data defined expression `@symbol_frame % 360` will cause the symbol to rotate over time, with rotation speed dictated by the symbol's frame rate:

Extent buffer

You may set an extent buffer for a *symbol*. This means that a buffer is applied to the current map extent so that if a feature is outside of the actual map extent but inside the buffered extent it will still be rendered. This is useful for example with symbols which use the geometry generator where you would like to still see the generated geometries even if the actual feature is outside of the map extent.

To edit the extent buffer you can utilize the *Extent buffer* panel.

1. Start by going to the top symbol level and select *Advanced* menu in the bottom right of the dialog
2. Find *Extent buffer* option
3. In the new panel you can set the buffer distance

The buffer distance units can be changed. You can also control the distance value by using the data defined override widget. For example you can change the value based on the current map scale `if(@map_scale > 50000, 5000, 0)`:

Desenhar efeitos

Para melhorar a renderização da camada e evitar (ou pelo menos reduzir) o recurso a outro software para a renderização final dos mapas, o QGIS fornece outra funcionalidade poderosa: os | Efeitos de pintura | :guilabel:opções *Efeitos de desenho*, que adicionam efeitos de pintura para personalizar a visualização de camadas vetoriais.

A opção está disponível na caixa de diálogo: seleção de menus: *Propriedades da camada-> Simbologia*, no grupo: ref:` Renderização de camada <layer_rendering>`(aplicável a toda a camada) ou em: ref:` propriedades da camada de símbolo<symbol-selector> `(aplicável às correspondentes recursos). Você pode combinar os dois usos.

Paint effects can be activated by checking the *Draw effects* option and clicking the Customize effects button. That will open the *Effect Properties* Dialog (see Fig.12.17). The following effect types, with custom options are available:

- ** Origem **: desenha o estilo original do recurso de acordo com a configuração das propriedades da camada. O:guilabel:*Opacidade* do seu estilo pode ser ajustado assim como: ref:` Modo de mistura` e:ref:*Modo Draw*. Essas são propriedades comuns para todos os tipos de efeitos.

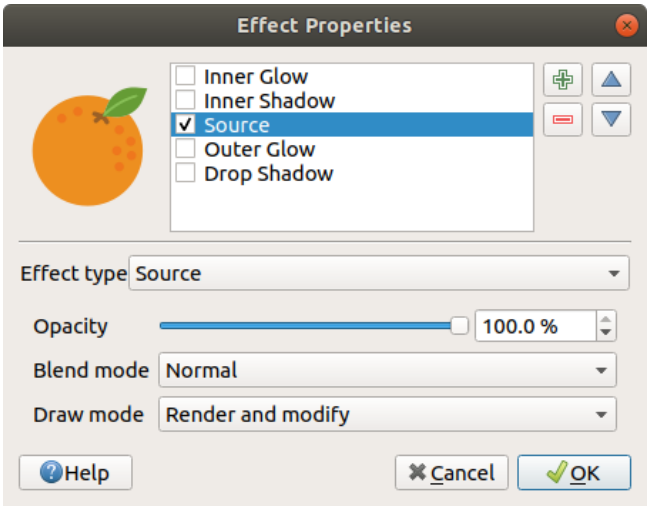


Figura12.17: Efeitos de desenho: caixa de diálogo Origem

- ** Desfoque **: adiciona um efeito de desfoque na camada vetorial. As opções personalizadas que você pode alterar são:guilabel:` Tipo de desfoque` (: guilabel:` Desfoque de pilha (fast) *ou: guilabel:* Desfoque Gaussiano (qualidade) *) e: guilabel:`Força do borrão*.

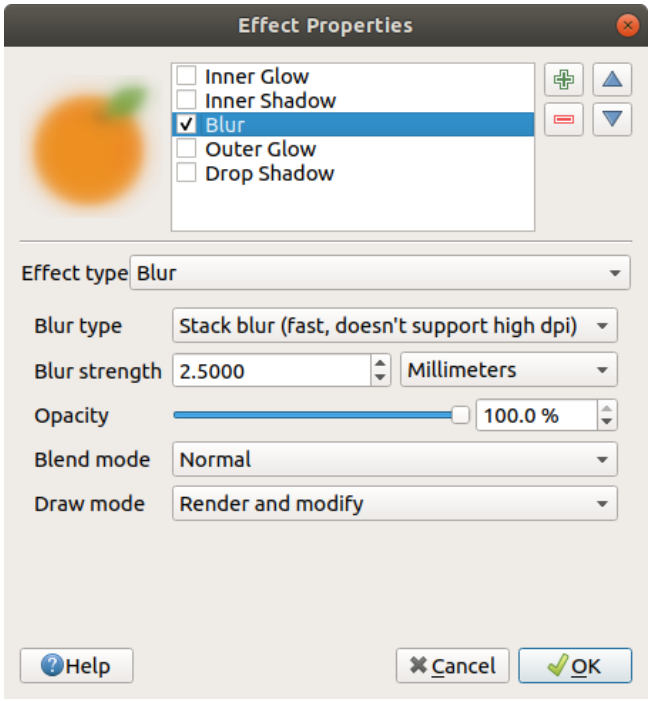


Figura12.18: Efeitos de desenho: caixa de diálogo Blur

- ** Colorizar **: esse efeito pode ser usado para criar uma versão do estilo usando uma única tonalidade. A base sempre será uma versão em escala de cinza do símbolo e você pode:
  - Use o | selecione String | *Escala de cinza* para selecionar como criá-la: as opções são 'Por luminosidade', 'Por luminosidade', 'Por média' e 'Desligado'.
  - Se | caixa de seleção | *Colorise* está selecionado, será possível misturar outra cor e escolher quão forte ela deve ser.
  - Controle os níveis:guilabel:*Brilho*,: guilabel:` Contraste` e:guilabel:*Saturaturação* do símbolo resultante.

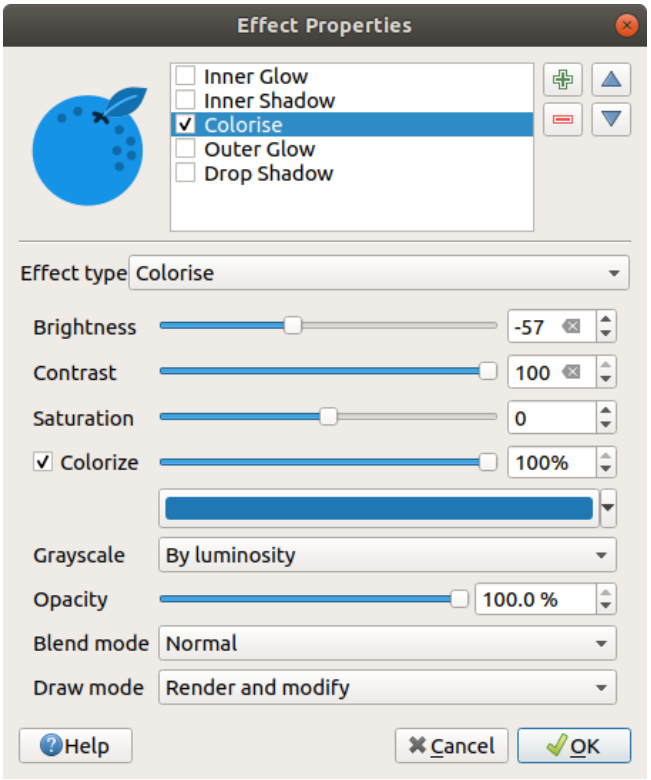


Figura12.19: Efeitos de desenho: caixa de diálogo Colorir

- ** Sombra projetada **: o uso desse efeito adiciona uma sombra ao recurso, que parece adicionar uma dimensão extra. Este efeito pode ser personalizado alterando o:guilabel:ângulo e distância do ` Deslocamento ‘, determinando para onde a sombra se desloca e a proximidade do objeto de origem. : menuselection: *Drop Shadow* também tem a opção de alterar: guilabel:` Raio de desfoque` e:guilabel:*Corr* do efeito.

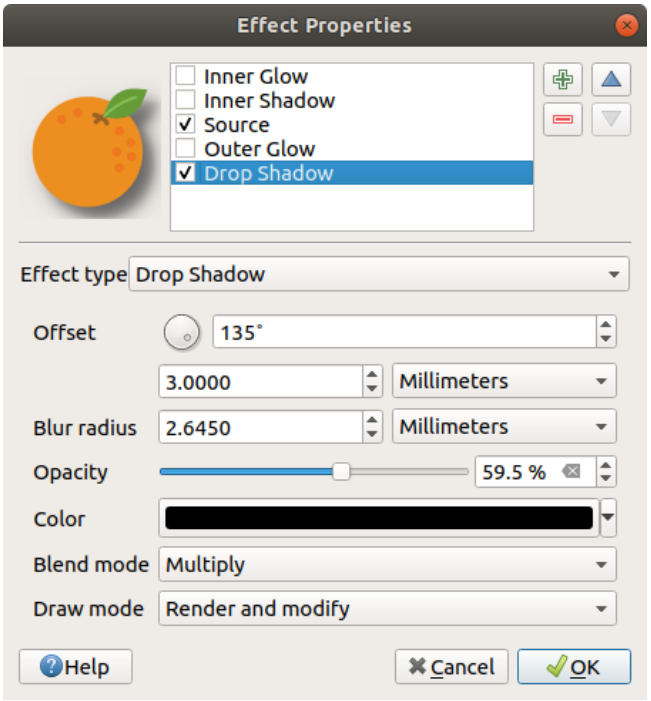


Figura12.20: Caixa de diálogo Efeitos de desenho: Sombra projetada

- ** Sombra interior**: Este efeito é semelhante ao efeito:guilabel:*Drop Shadow*, mas adiciona o efeito de sombra na parte interna das bordas do recurso. As opções disponíveis para personalização são as mesmas do efeito: guilabel: *Sombra projetada*.

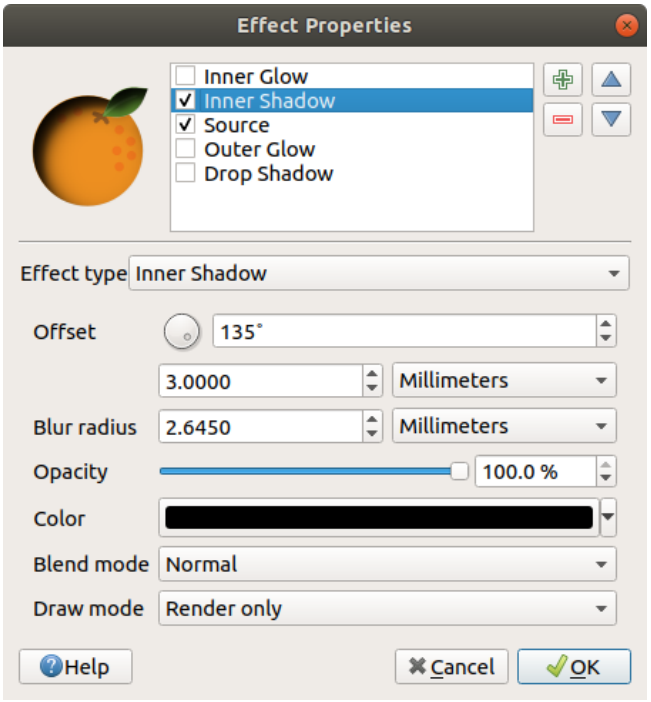


Figura12.21: Caixa de diálogo Efeitos de desenho: Sombra interna

- ** Brilho interno **: adiciona um efeito de brilho dentro do recurso. Este efeito pode ser personalizado ajustando:guilabel:*Espalhe* (largura) do brilho ou: guilabel:` Raio de desfoque`. Este último especifica a proximidade da borda do recurso em que você deseja que ocorra qualquer desfoque. Além disso, existem opções para personalizar a cor do brilho usando a:guilabel:*Única cor* ou a: guilabel:` Rampa de cores`.

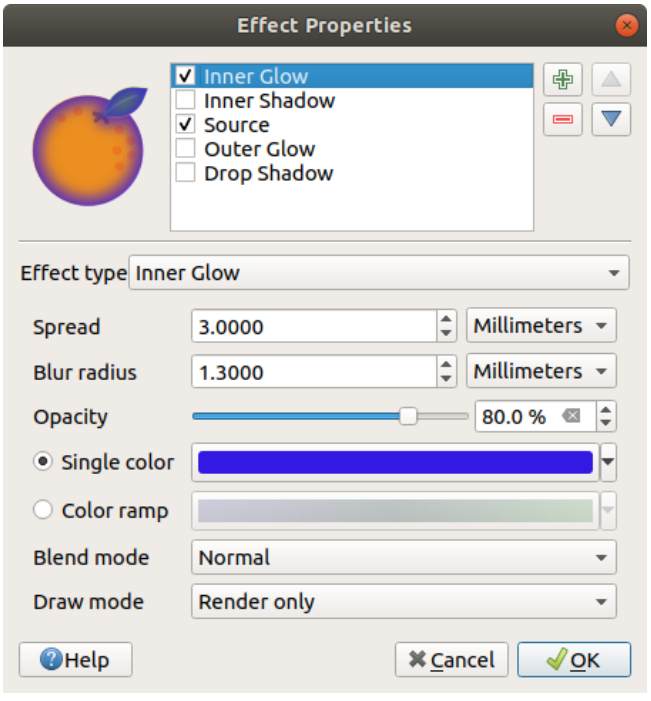


Figura12.22: Caixa de diálogo Efeitos de desenho: Brilho interno

- ** Brilho externo **: Este efeito é semelhante ao efeito:guilabel:*Brilho interno*, mas adiciona o efeito de brilho na parte externa das bordas do recurso. As opções disponíveis para personalização são as mesmas do efeito: guilabel: *Brilho interior*.

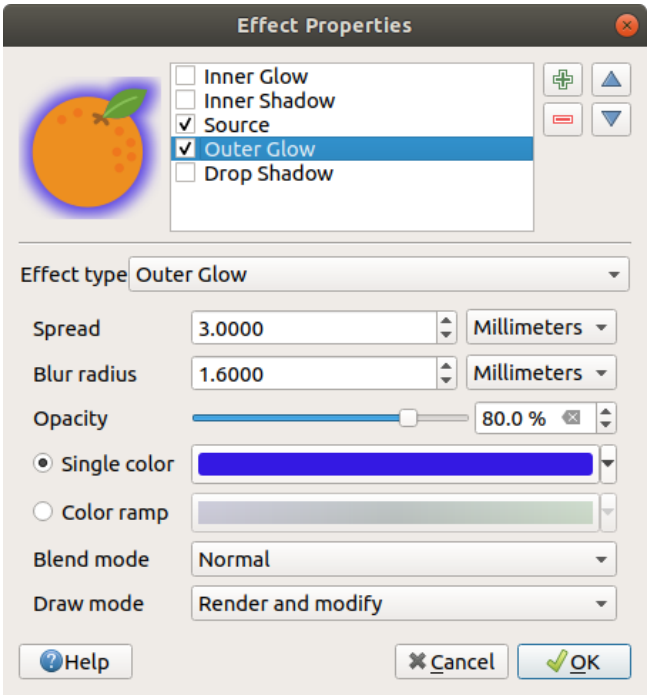


Figura12.23: Caixa de diálogo Draw Effects: Brilho externo

- **Transform**: Adds the possibility of transforming the shape of the symbol. The first options available for customization are the *Reflect horizontal* and *Reflect vertical*, which actually create a reflection on the horizontal and/or vertical axes. The other options are:
  - *Shear X, Y*: inclina o recurso ao longo do eixo X e / ou Y.
  - *Escala X, Y*: amplia ou minimiza o recurso ao longo do eixo X e / ou Y pela porcentagem especificada.
  - *Rotação*: gira o recurso em torno de seu ponto central.
  - e:guilabel:"Traduzir X, Y" altera a posição do item com base na distância indicada no eixo X e / ou Y.

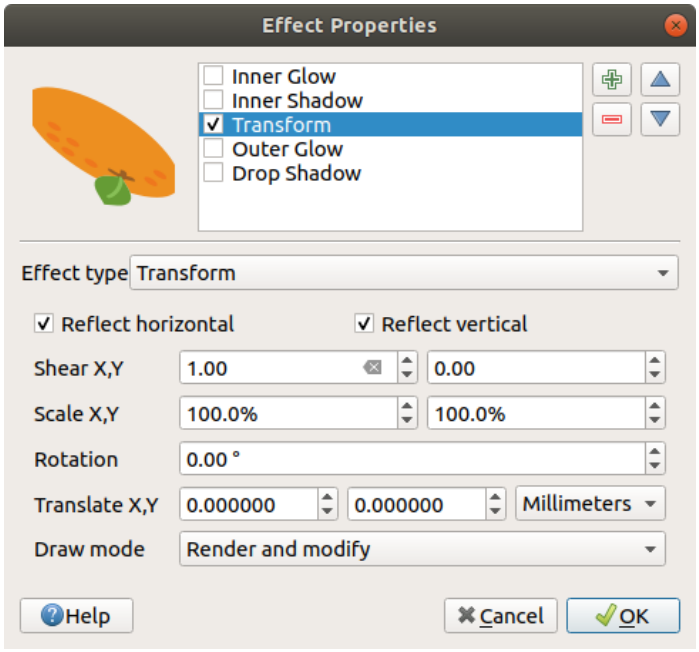


Figura12.24: Caixa de diálogo Efeitos de desenho: transformação

Um ou mais tipos de efeito podem ser usados ao mesmo tempo. Você (des) ativa um efeito usando sua caixa de seleção na lista de efeitos. Você pode alterar o tipo de efeito selecionado usando o | selecione String | : guilabel: opção *Tipo de efeito*. Você pode reordenar os efeitos usando | seta para cima | Mover para cima e | seta para baixo | :sup:botões *Mover para baixo* e também adicionar / remover efeitos usando o | assinar mai | Adicionar novo efeito e | assinar menos | :sup:botões *Remover efeito*.

Existem algumas opções comuns disponíveis para todos os tipos de efeito de desenho. As opções:guilabel:*Opacidade* e: guilabel:` Modo de mistura` funcionam de maneira semelhante às descritas em:ref:*renderização de camada* e podem ser usadas em todos os efeitos de desenho, exceto o de transformação.

Há também uma |selecione String | :guilabel:opção `Draw mode 'disponível para todos os efeitos, e você pode optar por renderizar e / ou modificar o símbolo, seguindo algumas regras:

- Os efeitos são renderizados de cima para baixo.
- :guilabel:o modo *Renderizar apenas* significa que o efeito será visível.
- :guilabel:modo "Somente modificador" significa que o efeito não será visível, mas as alterações que ele aplicar serão passadas para o próximo efeito (o imediatamente abaixo).
- O modo:guilabel:*Renda e modifique* tornará o efeito visível e passará as alterações para o próximo efeito. Se o efeito estiver no topo da lista de efeitos ou se o efeito imediatamente acima não estiver no modo de modificação, ele usará o símbolo de origem original nas propriedades das camadas (semelhante à origem).

12.1.4 Propriedades das etiquetas

A rotulagem | :guilabel:As propriedades *Etiquetas* fornecem todos os recursos necessários e adequados para configurar a etiquetagem inteligente em camadas vetoriais. Essa caixa de diálogo também pode ser acessada no painel:guilabel:*Estilo da camada* ou usando o rótulo | :sup:botão *Opções de rotulagem de camada* da barra de ferramentas ** Etiquetas **.

At the top of the dialog, you have:

- a combobox for selecting the appropriate labeling method for the active layer
- the Configure project labeling rules button: helps you control interactions between labels and features across the layers in the project. More details at *Configuring project labeling rules*.

- the Automated placement settings (applies to all layers) button: configure general properties on label placement and conflicts resolution. More details at *Configurando o mecanismo de posicionamento automatizado*.

O primeiro passo é escolher o método de rotulagem na lista suspensa. Os métodos disponíveis são:

- *No labels*: the default value, showing no labels from the layer
- *Single labels*: Show labels on the map using a single attribute or an expression
- **|com base em regras|** *Rotulagem baseada em regras*
- e | rotulagem Obstáculo | *Bloqueio*: permite definir uma camada apenas como um obstáculo para os rótulos de outras camadas sem renderizar nenhum rótulo próprio.

As próximas etapas pressupõem que você selecione a | etiqueta | :guilabel:opção *Etiquetas únicas*, abrindo a seguinte caixa de diálogo.

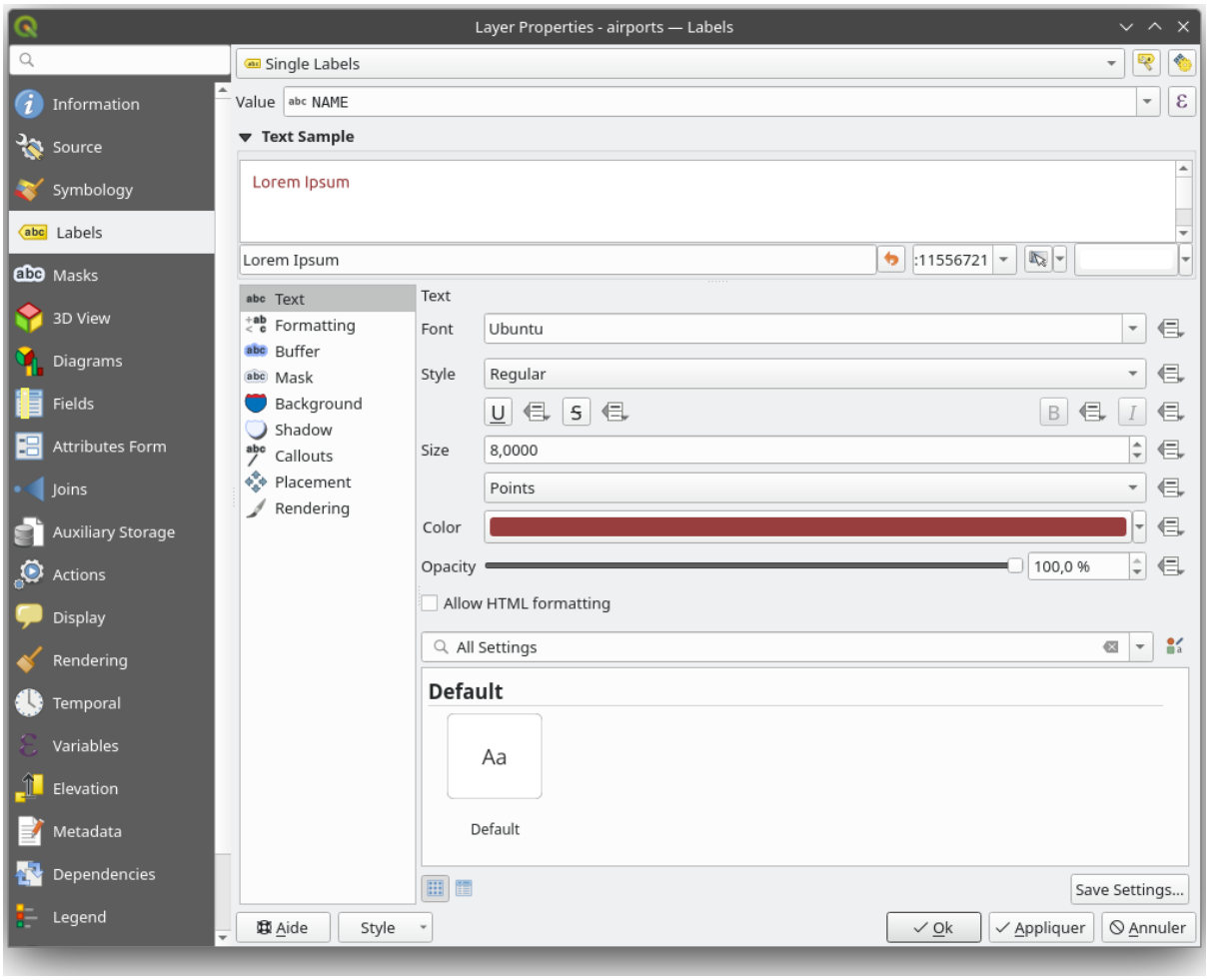


Figura12.25: Configurações de rotulagem de camada - Etiquetas únicas

Na parte superior da caixa de diálogo, uma lista suspensa:guilabel:*Valor* está ativada. Você pode selecionar uma coluna de atributo a ser usada para rotular. Por padrão, o:ref:*campo de exibição<maptips>* é usado. Clique em | expressão | se você deseja definir rótulos com base em expressões - Veja: ref: *rotulando com expressões*.

> **Nota**
>
> Labels with their formatting can be displayed as entries in the legends, if enabled in the *Legend* tab.

Abaixo estão as opções exibidas para personalizar os rótulos, em várias guias:

- texto | *Texto*
- formatação de etiquetas | *Formatação*
- *Buffer*
- *Mask*
- *Background*
- *Shadow*
- *Callouts*
- *Placement*
- *Rendering*

A descrição de como definir cada propriedade é exposta em:ref:*showlabels*.

Configuring project labeling rules

Pressing the Configure project labeling rules button next to the labeling method drop-down selector, you can create rules that controls how labels from a layer can interact with labels or features from another layer.

1. Press the Add rule button and in the drop-down menu, select one of the rule types:
   - *Prevent labels overlapping features*: prevents labels being placed overlapping features from a different layer.
   - *Pull labels towards features*: prevents labels being placed too far from features from a different layer. The maximum distance can be set in the unit of your choice.
   - *Push labels away from features*: prevents labels being placed too close to features from a different layer. The minimum distance can be set in the unit of your choice, as well as the *rule's priority* (The highest-priority rules are more important to respect in the event of a label placement conflict).
   - *Push labels away from other labels*: prevents labels being placed too close to labels from a different layer.

   Atenção

   The last three options are only available on QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).

2. Fill the properties at your will; you can provide a more meaningful name to the rule.
3. Pressione *OK*.
4. Add as many rules as necessary.
5. If necessary, press Edit rule to modify the selected rule or Remove rule to delete it from the project.

The set rules are available from any layer *Labels* properties tab, pressing the Configure project labeling rules button. You can temporary enable or disable any of them, using the checkbox next to the name. Hover over a rule to preview its details.

Configurando o mecanismo de posicionamento automatizado

You can use the automated placement settings to configure a project-level automated behavior of the labels. In the top right corner of the *Labels* tab, click the Automated placement settings (applies to all layers) button, opening a dialog with the following options:

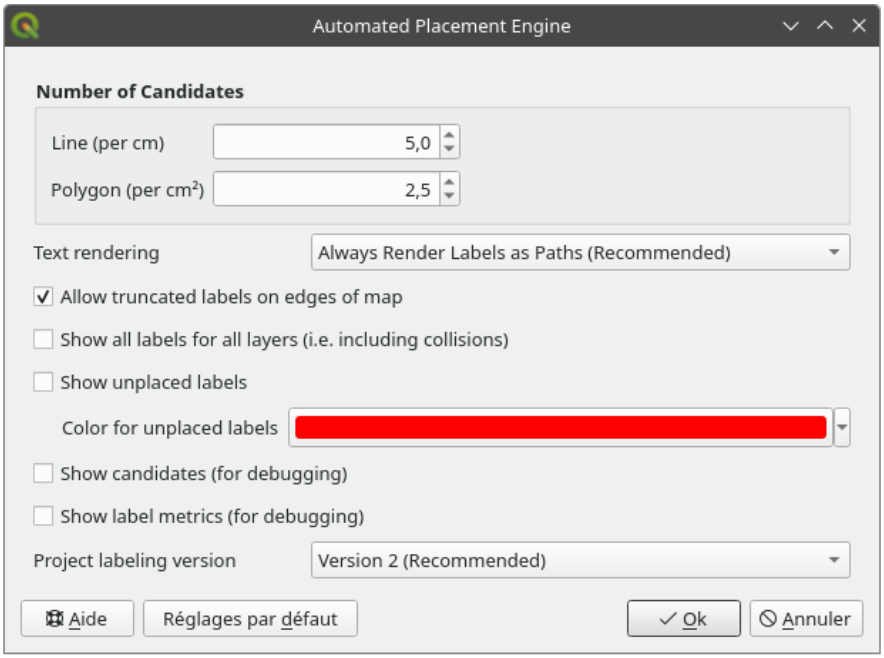


Figura12.26: O mecanismo de posicionamento automatizado de etiquetas

- *Number of candidates*: calculates and assigns to line and polygon features the number of possible labels placement based on their size. The longer or wider a feature is, the more candidates it has, and its labels can be better placed with less risk of collision.
- *Text rendering*: sets the default value for label rendering widgets when *exporting a map canvas* or *a layout* to PDF or SVG. If *Always render labels as text* is selected then labels can be edited in external applications (e.g. Inkscape) as normal text. BUT the side effect is that the rendering quality is decreased, and there are issues with rendering when certain text settings like buffers are in place. That's why *Always render labels as paths (recommended)* which exports labels as outlines but guarantees complete compatibility with the full range of formatting options available, is recommended. With *Prefer rendering labels as text* option, labels are rendered as text objects, unless doing so results in rendering artifacts or poor quality rendering (depending on text format settings).

> **Nota**
>
> When rendering labels as text to a vector based device (e.g. PDF or SVG), care must be taken to ensure that the required fonts are available to users when opening the created files, or default fallback fonts will be used to display the output instead. (Although PDF exports MAY automatically embed some fonts when possible, depending on the user's platform).

- *Allow truncated labels on edges of map*: controls whether labels which fall partially outside of the map extent should be rendered. If checked, these labels will be shown (when there's no way to place them fully within the visible area). If unchecked then partially visible labels will be skipped. Note that this setting has no effects on labels' display in the *layout map item*.
- |desmarcado | *Mostrar todos os rótulos para todas as camadas (isto é, incluindo objetos em colisão)*. Observe que esta opção também pode ser definida por camada (consulte:ref:*etiquetas que rendem*)

- *Show unplaced labels*: allows to determine whether any important labels are missing from the maps (e.g. due to overlaps or other constraints). They are displayed using a customizable color.
- *Show candidates (for debugging)*: controls whether boxes should be drawn on the map showing all the candidates generated for label placement. Like the label says, it's useful only for debugging and testing the effect different labeling settings have. This could be handy for a better manual placement with tools from the *label toolbar*.
- *Show label metrics (for debugging)*: displays the text bounds of the label in red and baselines in blue
- *Project labeling version*: QGIS supports two different versions of label automatic placement:
  - *Version 1*: the old system (used by QGIS versions 3.10 and earlier, and when opening projects created in these versions in QGIS 3.12 or later). Version 1 treats label and obstacle priorities as "rough guides" only, and it's possible that a low-priority label will be placed over a high-priority obstacle in this version. Accordingly, it can be difficult to obtain the desired labeling results when using this version and it is thus recommended only for compatibility with older projects.
  - *Version 2 (recommended)*: this is the default system in new projects created in QGIS 3.12 or later. In version 2, the logic dictating when labels are allowed to overlap *obstacles* has been reworked. The newer logic forbids any labels from overlapping any obstacles with a greater obstacle weight compared to the label's priority. As a result, this version results in much more predictable and easier to understand labeling results.

Rotulagem baseada em regras

Com a rotulagem baseada em regras, várias configurações de rótulos podem ser definidas e aplicadas seletivamente na base dos filtros de expressão e no intervalo de escala, como em:ref:*Renderização baseada em regras<rule_based_rendering>*.

Para criar uma regra:

1. Select the **Rule-based labeling** option in the main drop-down list from the *Labels* tab
2. Clique no botão Adicionar regra na parte inferior da caixa de diálogo.
3. Preencher a nova caixa de diálogo com:
   - *Description*: a text used to identify the rule in the *Labels* tab and as a *label legend entry* in the print layout legend
   - guilabel:*Filtro*: uma expressão para selecionar as características para aplicar as configurações do rótulo a
   - Se já houver regras definidas, a opção *Senão* pode ser utilizada para selecionar todas as características que não correspondem a nenhum filtro das regras no mesmo grupo.
4. Você pode definir um intervalo de escala de ref:`referência' na qual a regra de rótulo deve ser aplicada.
5. The options available under the *Labels* group box are the usual *label settings*. Configure them and press *OK*.

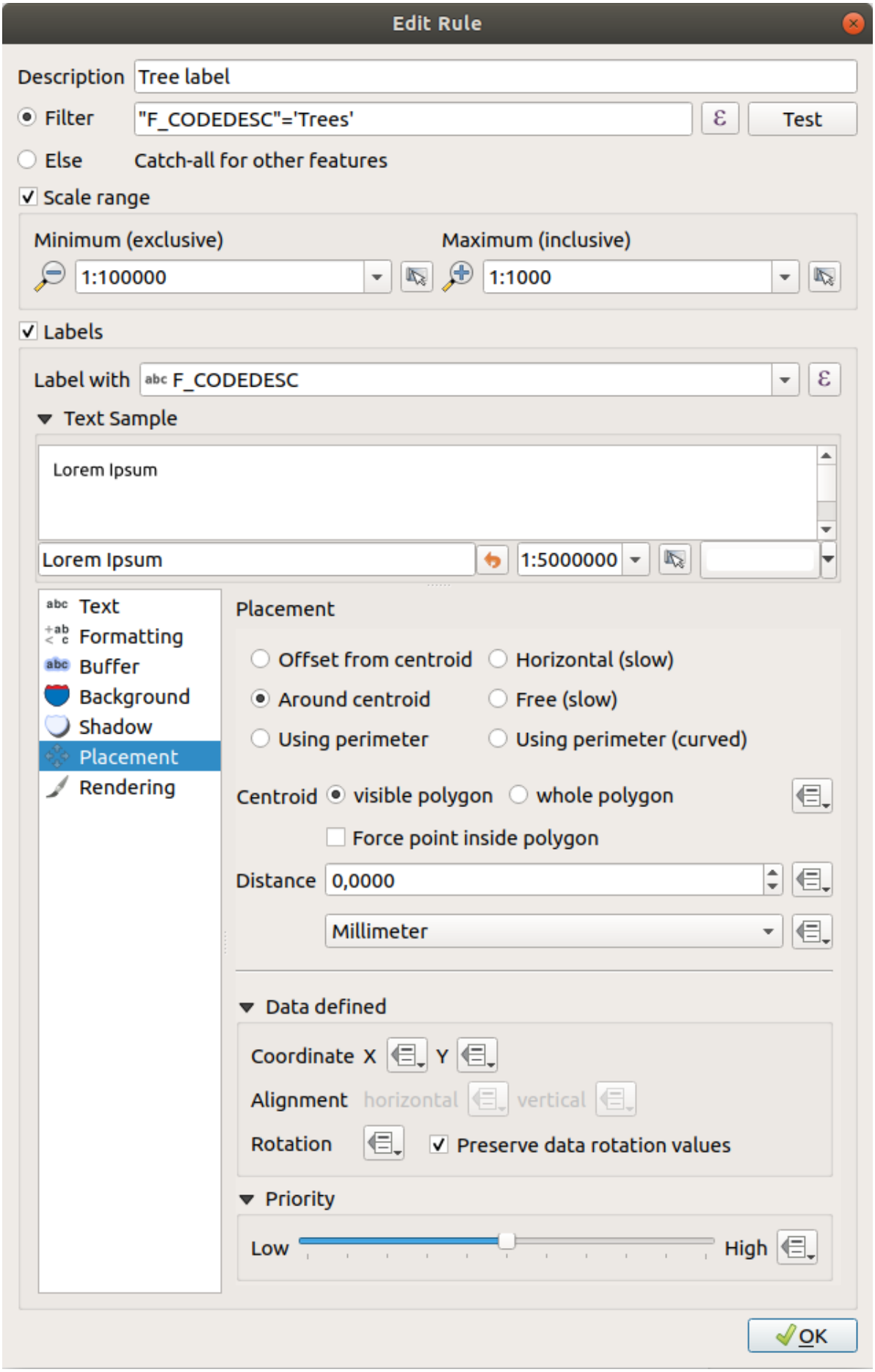


Figura12.27: Configurações de regra

A summary of existing rules is shown in the main dialog (see Fig.12.28). You can add multiple rules, reorder or imbricate them with a drag-and-drop. You can as well remove them with the button or edit them with button or a double-click.

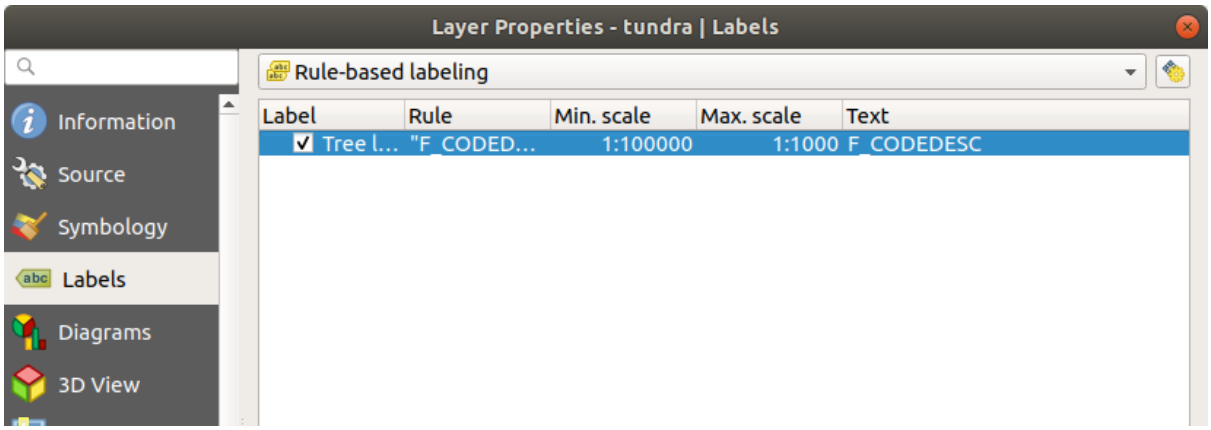


Figura12.28: Painel de rotulagem baseado em regras

Definir rótulos com base em expressões

Se você escolhe o tipo de rotulagem única ou baseada em regras, o QGIS permite o uso de expressões para rotular os recursos.

Supondo que você esteja usando o método:guilabel:*Etiquetas individuais*, clique no | expressão | próximo à lista suspensa:guilabel:*Valor* na | etiqueta | :guilabel:guia *Etiquetas* da caixa de diálogo de propriedades.

In Fig.12.29, you see a sample expression to label the alaska trees layer with tree type and area, based on the field 'VEGDESC', some descriptive text, and the function `$area` in combination with `format_number()` to make it look nicer.

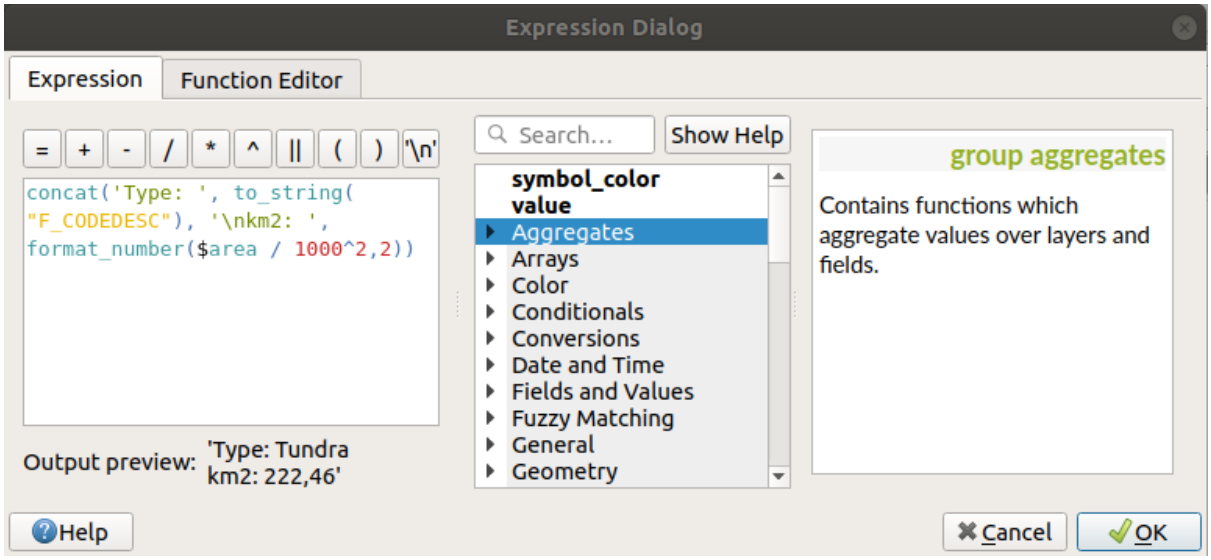


Figura12.29: Usando expressões para rotular

É fácil trabalhar com etiquetas baseadas em expressões. Tudo o que você precisa cuidar é o seguinte:

- Pode ser necessário combinar todos os elementos (strings, campos e funções) com uma função de concatenação de strings, como `` concat``, `` + `` ou `` || ``. Esteja ciente de que, em algumas situações (quando houver valor nulo ou numérico), nem todas essas ferramentas atenderão à sua necessidade.

- As strings são escritas em 'aspas simples'.
- Os campos são escritos em "aspas duplas" ou sem aspas.

Vamos dar uma olhada em alguns exemplos:

1. Rótulo com base em dois campos 'nome' e 'local' com uma vírgula como separador:

```
"name" || ', ' || "place"
```

   Retorna :

```
John Smith, Paris
```

2. Rótulo com base em dois campos 'nome' e 'local' com outros textos:

```
'My name is ' + "name" + 'and I live in ' + "place"
'My name is ' || "name" || 'and I live in ' || "place"
concat('My name is ', name, ' and I live in ', "place")
```

   Retorna :

```
My name is John Smith and I live in Paris
```

3. Rótulo com base em dois campos 'nome' e 'local' com outros textos que combinam diferentes funções de concatenação:

```
concat('My name is ', name, ' and I live in ' || place)
```

   Retorna :

```
My name is John Smith and I live in Paris
```

   Ou, se o campo 'local' for NULO, retornará:

```
My name is John Smith
```

4. Rótulo de várias linhas com base em dois campos 'nome' e 'local' com um texto descritivo:

```
concat('My name is ', "name", '\n' , 'I live in ' , "place")
```

   Retorna :

```
My name is John Smith
I live in Paris
```

5. Rótulo com base em um campo e na função $ area para mostrar o nome do local e seu tamanho da área arredondada em uma unidade convertida:

```
'The area of ' || "place" || ' has a size of '
|| round($area/10000) || ' ha'
```

   Retorna :

```
The area of Paris has a size of 10500 ha
```

6. Crie uma condição CASE ELSE. Se o valor da população no campo "população" for <= 50000, é uma cidade, caso contrário, é uma cidade:

```
concat('This place is a ',
CASE WHEN "population" <= 50000 THEN 'town' ELSE 'city' END)
```

   Retorna :

```
This place is a town
```

7. Nome para exibição das cidades e nenhum rótulo para os outros recursos (para o contexto “cidade”, veja o exemplo acima):

   ```
   CASE WHEN "population" > 50000 THEN "NAME" END
   ```

   Retorna :

   ```
   Paris
   ```

Como você pode ver no construtor de expressões, você tem centenas de funções disponíveis para criar expressões simples e muito complexas para rotular seus dados no QGIS. Veja o capítulo ref: *vector_expressions* para mais informações e exemplos sobre expressões.

Usando substituição definida por dados para rotular

Com o | dadosDefinidos | :sup:função “Substituição definida dos dados”, as configurações para a rotulagem são substituídas pelas entradas na tabela de atributos ou pelas expressões baseadas nelas. Esse recurso pode ser usado para definir valores para a maioria das opções de rotulagem descritas acima.

Por exemplo, usando o conjunto de dados de amostra QGIS do Alasca, vamos rotular a camada: file: *aeroportos* com seu nome, com base em seu militar` *USE*`, ou seja, se o aeroporto está acessível para:

- militares, em seguida, exibi-lo na cor cinza, tamanho 8;
- outros, em seguida, mostrar na cor azul, tamanho 10.

Para fazer isso, depois de habilitar a identificação no campo `` NAME`` da camada (consulte:ref:*showlabels*):

1. Ative a guia:guilabel:*Texto*.
2. Clique no | dadosDefinidos | ícone ao lado da propriedade:guilabel:*Tamanho*.
3. Selecione:guilabel:*Edit* … e digite

   ```
   CASE
     WHEN "USE" like '%Military%' THEN 8 -- because compatible values are
   ↪'Military'
                                       -- and 'Joint Military/Civilian'
     ELSE 10
   END
   ```

4. Pressione:guilabel:*OK* para validar. A caixa de diálogo é fechada e o | dadosDefinidos | o botão se torna | data Definir expressão ativada | significando que uma regra está sendo executada.
5. Em seguida, clique no botão ao lado da propriedade color, digite a expressão abaixo e valide:

   ```
   CASE
     WHEN "USE" like '%Military%' THEN '150, 150, 150'
     ELSE '0, 0, 255'
   END
   ```

Da mesma forma, você pode personalizar qualquer outra propriedade do rótulo da maneira que desejar. Veja mais detalhes sobre o | dadosDefinidos | : sup: descrição e manipulação do ferramenta *Substituição de definição de dados* na seção: ref:` data_defined`.

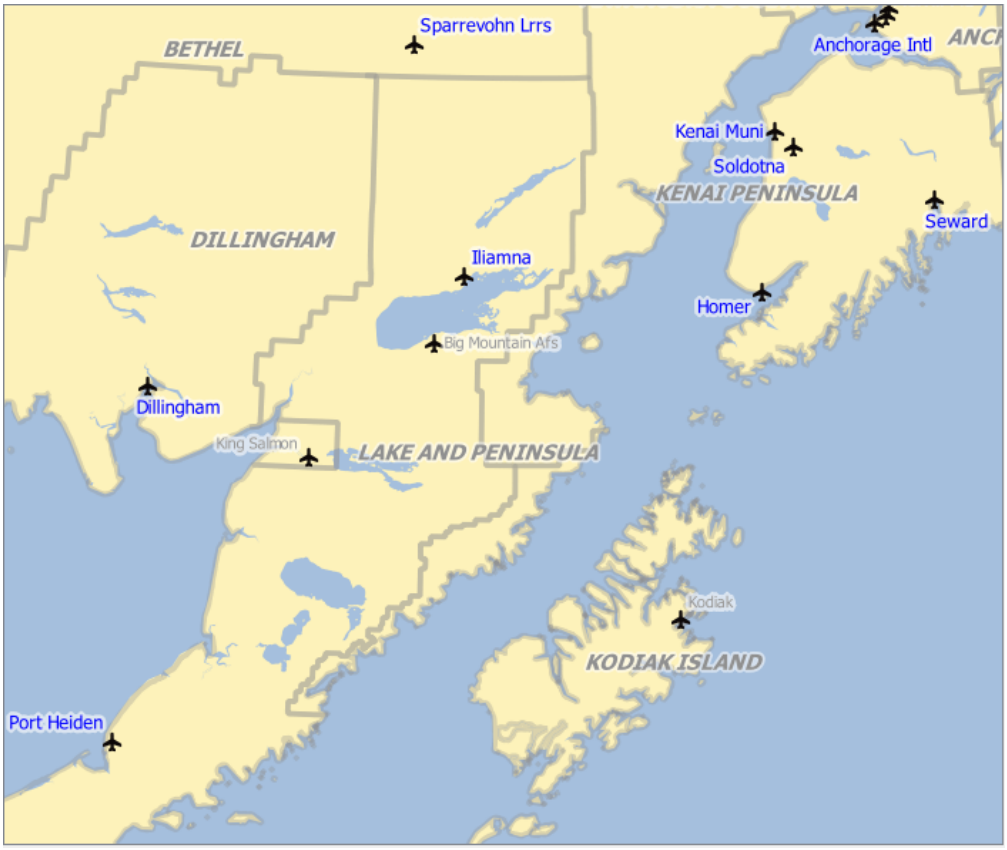


Figura12.30: Os rótulos dos aeroportos são formatados com base em seus atributos

Dica

** Use a substituição definida por dados para rotular todas as partes dos recursos de várias partes **

Há uma opção para definir a rotulagem para recursos de várias partes independentemente das propriedades da sua etiqueta. Escolha o | render | : ref: *Renderização<labels_rendering>*, `` Opções de recursos``, vá para | dados-Definidos | :sup:botão *Substituição de definição de dados* ao lado da caixa de seleção | desmarcado | *Rotule todas as partes dos recursos de várias partes* e defina os rótulos como descrito em: ref:` data_defined`.

A barra de ferramentas Rótulo

The *Label Toolbar* provides some tools to manipulate *label* (including their *callout*) or *diagram* properties:

Figura12.31: A barra de ferramentas Rótulo

- Highlight Pinned Labels, Diagrams and Callouts. If the vector layer of the item is editable, then the highlighting is green, otherwise it's blue.
- Toggle Display of Unplaced Labels: Allows to determine whether any important labels are missing from the maps (e.g. due to overlaps or other constraints). They are displayed with a customizable color (see *Configurando o mecanismo de posicionamento automatizado*).

- Pin/Unpin Labels and Diagrams. By clicking or draging an area, you pin overlaid items. If you click or drag an area holding `Shift`, the items are unpinned. Finally, you can also click or drag an area holding `Ctrl` to toggle their pin status.
- Show/Hide Labels and Diagrams. If you click on the items, or click and drag an area holding `Shift`, they are hidden. When an item is hidden, you just have to click on the feature to restore its visibility. If you drag an area, all the items in the area will be restored.
- Move a Label, Diagram or Callout: click to select the item and click to move it to the desired place. The new coordinates are stored in *auxiliary fields*. Selecting the item with this tool and hitting the `Delete` key will delete the stored position value.
- Rotate a Label. Click to select the label and click again to apply the desired rotation. Likewise, the new angle is stored in an auxiliary field. Selecting a label with this tool and hitting the `Delete` key will delete the rotation value of this label.
- Change Label Properties. It opens a dialog to change the clicked label properties; it can be the label itself, its coordinates, angle, font, size, multiline alignment … as long as this property has been mapped to a field. Here you can set the option to *Label every part of a feature*.

Aviso

** As ferramentas de etiqueta substituem os valores atuais do campo **

O uso da barra de ferramentas:guilabel:*Rótulo* para personalizar a rotulagem realmente grava o novo valor da propriedade no campo mapeado. Portanto, tenha cuidado para não substituir inadvertidamente os dados necessários posteriormente!

Nota

O mecanismo:ref:*vector_auxiliary_storage* pode ser usado para personalizar a identificação (posição e assim por diante) sem modificar a fonte de dados subjacente.

Personalizar os rótulos da tela do mapa

Combined with the *Label Toolbar*, the data defined override setting helps you manipulate labels in the map canvas (move, edit, rotate). We now describe an example using the data-defined override function for the Move Label, Diagram or Callout function (see Fig.12.32).

1. Importe: file: *lakes.shp* do conjunto de dados de amostra do QGIS.
2. Clique duas vezes na camada para abrir as propriedades da camada. Clique em: guilabel: *Etiquetas* e: guilabel:*Canal*. Selecione **|botão de opção Ativado|** *Deslocamento do centróide*.
3. Procure as entradas:guilabel:*Dados definidos*. Clique no | Dadosdefinidos | ícone para definir o tipo de campo para:guilabel:*Coordenada*. Escolha `` xlabel`` para X e `` ylabel`` para Y. Os ícones agora estão destacados em amarelo.

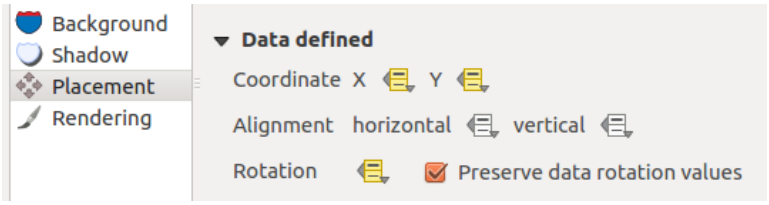


Figura12.32: Rotulagem de camadas de polígono de vetor com substituição definida por dados

4. Zoom em um lago
5. Defina editável a camada usando o | Editar edição | :sup:botão *Alternar edição*.
6. Go to the Label toolbar and click the icon. Now you can shift the label manually to another position (see Fig.12.33). The new position of the label is saved in the `xlabel` and `ylabel` columns of the attribute table.
7. Também é possível adicionar uma linha conectando cada lago a sua etiqueta movida usando:
   - the label's *callout property*
   - or the *geometry generator symbol layer* with the expression below:

```
make_line( centroid( $geometry ), make_point( "xlabel", "ylabel" ) )
```

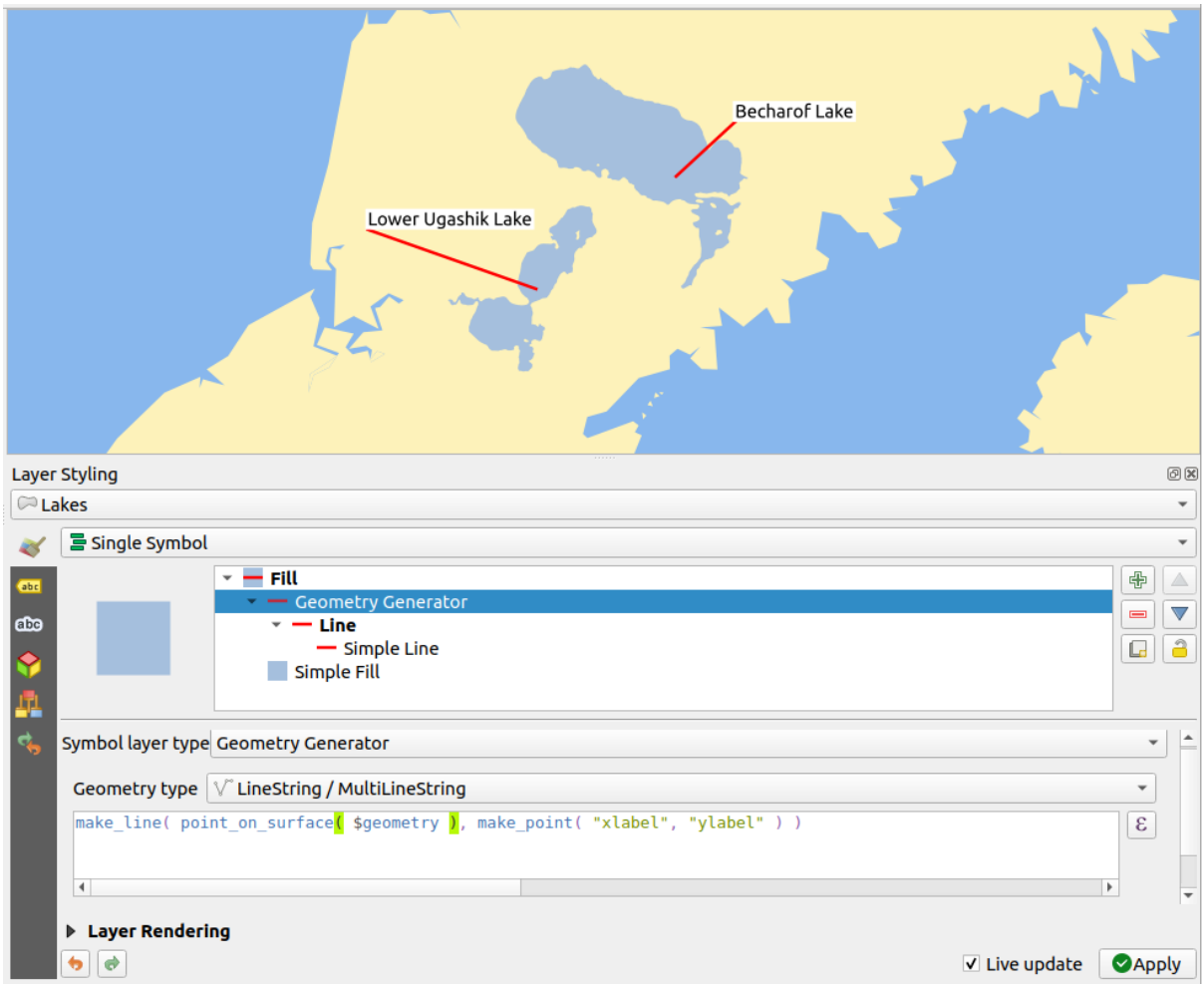


Figura12.33: Etiquetas movidas

Nota

O mecanismo:ref:*vector_auxiliary_storage* pode ser usado com propriedades definidas por dados sem ter uma fonte de dados editável.

12.1.5 Propriedades dos diagramas

The *Diagrams* tab allows you to add a graphic overlay to a vector layer (see Fig.12.34). This dialog can also be accessed from the *Layer Styling* panel, or using the Layer Diagram Options button of the **Labels toolbar**.

A implementação principal atual de diagramas fornece suporte para:

- *No diagrams*: the default value with no diagram displayed over the features;
- *Pie chart*, a circular statistical graphic divided into slices to illustrate numerical proportion. The arc length of each slice is proportional to the quantity it represents;
- abc *Text diagram*, a horizontally divided circle showing statistic values inside;
- *Histogram*, bars of varying colors for each attribute aligned next to each other;
- *Stacked bars*, stacks bars of varying colors for each attribute on top of each other vertically or horizontally;
- *Stacked diagram*, stacks diagrams of equal or varying types, next to each other, vertically or horizontally. More details at *Stacked Diagrams*.

No canto superior direito da guia:guilabel:*Diagramas*, o | posicionamento automático | O botão:sup:*Configurações automáticas de posicionamento (aplica-se a todas as camadas)* fornece meios para controlar o diagrama: ref: *etiqueta posicionamento<automated_placement>* na tela do mapa.

Dica

** Alterne rapidamente entre tipos de diagramas **

Como as configurações são quase comuns aos diferentes tipos de diagrama, ao projetar seu diagrama, você pode alterar facilmente o tipo de diagrama e verificar qual é o mais apropriado para seus dados sem perda.

Para cada tipo de diagrama, as propriedades são divididas em várias guias:

- *Atributos*
- *Renderização*
- *Tamanho*
- *Posicionamento*
- *Opções*
- *Legenda*

Atributos

Atributos define quais variáveis exibir no diagrama. Use | assinarPlus | :sup:botão *adicionar item* para selecionar os campos desejados no painel 'Atributos atribuídos'. Atributos gerados com:ref:*vector_expressions* também podem ser usados.

Você pode mover para cima e para baixo qualquer linha com clique e arraste, classificando como os atributos são exibidos. Você também pode alterar o rótulo na coluna 'Legenda' ou a cor do atributo clicando duas vezes no item.

Esse rótulo é o texto padrão exibido na legenda do layout de impressão ou da árvore de camadas.

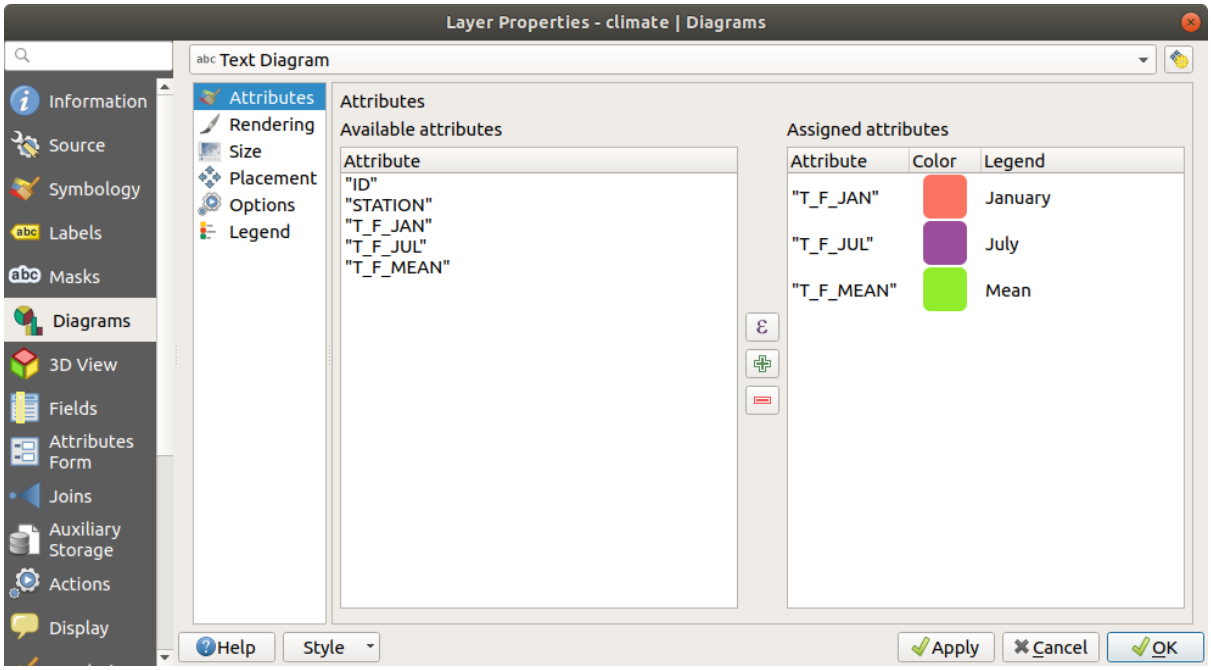


Figura12.34: Diagram properties - Attributes tab

Renderização

Renderização define como o diagrama se parece. Ele fornece configurações gerais que não interferem nos valores estatísticos, como:

- a opacidade do gráfico, sua largura e cor do contorno;
- dependendo do tipo de diagrama:
  - for histogram and stacked bars, the width of the bar and the spacing between the bars. You may want to set the spacing to `0` for stacked bars. Moreover, the *Axis line symbol* can be made visible on the map canvas and customized using *line symbol properties*.
  - for text diagram, the circle background color and the *font* used for texts;
  - for pie charts, the *Start angle* of the first slice and their *Direction* (clockwise or not).
- the use of *paint effects* on the graphics.

Nesta guia, você também pode gerenciar e ajustar a visibilidade do diagrama com diferentes opções:

- *Diagram z-index*: controls how diagrams are drawn on top of each other and on top of labels. A diagram with a high index is drawn over other diagrams and labels;
- *Show all diagrams*: shows all the diagrams even if they overlap each other;
- *Mostrar diagrama*: permite que somente diagramas específicos sejam renderizados;

- *Sempre mostrar*: seleciona diagramas específicos para sempre renderizar, mesmo quando eles se sobrepõem a outros diagramas ou etiquetas de mapas;
- configurando:ref:*Visibilidade dependente da escala<label_scaledepend>*;

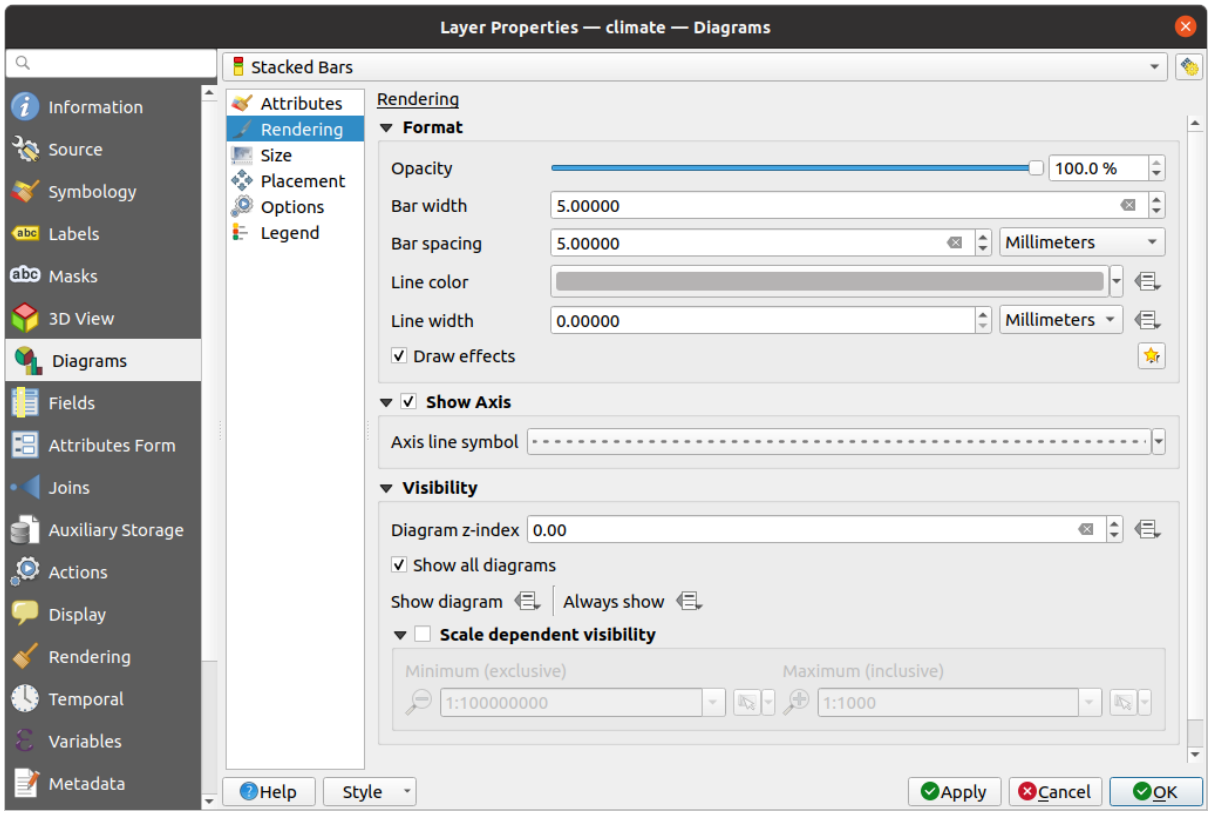


Figura12.35: Propriedades do diagrama - guia Renderização

Tamanho

Size is the main tab to set how the selected statistics are represented. The diagram size *units* can be 'Millimeters', 'Points', 'Pixels', 'Map Units' or 'Inches'. You can use:

- *Fixed size*, a unique size to represent the graphic of all the features (not available for histograms)
- or *Scaled size*, based on an expression using layer attributes:
  1. In *Attribute*, select a field or build an expression
  2. Press *Find* to return the *Maximum value* of the attribute or enter a custom value in the widget.
  3. For histogram and stacked bars, enter a *Bar length* value, used to represent the *Maximum value* of the attributes. For each feature, the bar length will then be scaled linearly to keep this matching.
  4. For pie chart and text diagram, enter a *Size* value, used to represent the *Maximum value* of the attributes. For each feature, the circle area or diameter will then be scaled linearly to keep this matching (from `0`). A *Minimum size* can however be set for small diagrams.

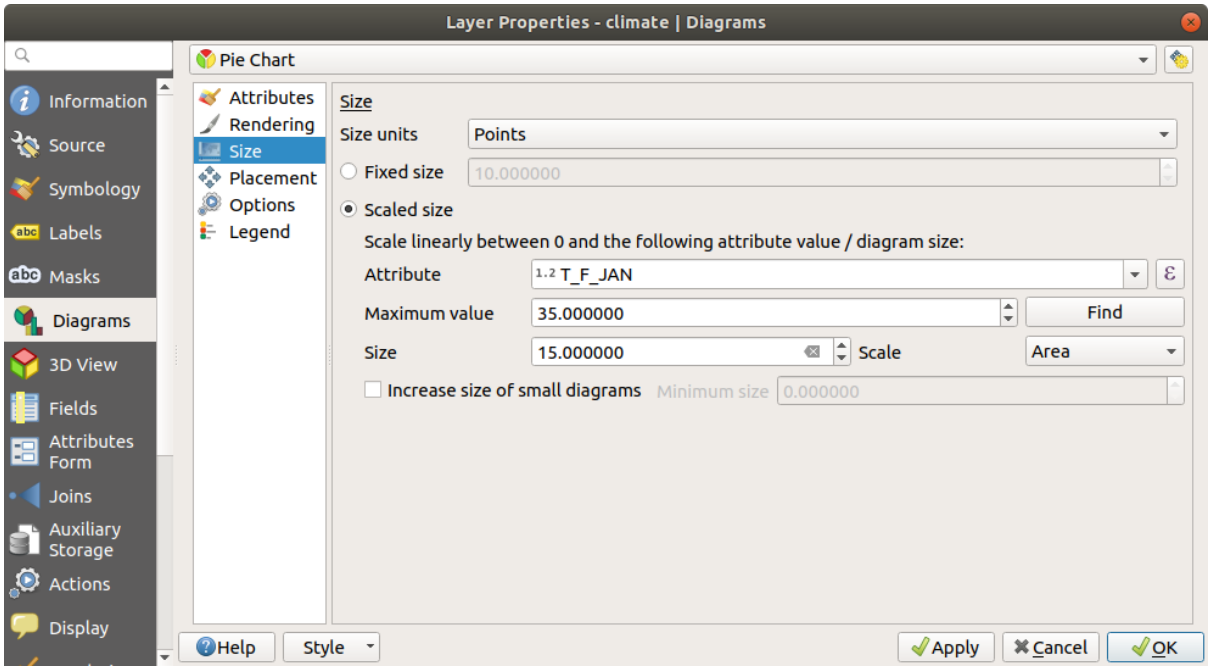


Figura12.36: Propriedades do diagrama - guia Tamanho

Localização

Placement defines the diagram position. Depending on the layer geometry type, it offers different options for the placement (more details at *Placement*):

- *Around point* or *Over point* for point geometry. The former variable requires a radius to follow.
- *Around line* or *Over line* for line geometry. Like point feature, the first variable requires a distance to respect and you can specify the diagram placement relative to the feature ('above', 'on' and/or 'below' the line) It's possible to select several options at once. In that case, QGIS will look for the optimal position of the diagram. Remember that you can also use the line orientation for the position of the diagram.
- *Around centroid* (at a set *Distance*), *Over centroid*, *Using perimeter* and *Inside polygon* are the options for polygon features.

The *Coordinate* group provides direct control on diagram placement, on a feature-by-feature basis, using their attributes or an expression to set the *X* and *Y* coordinate. The information can also be filled using the *Move labels and diagrams* tool.

In the *Priority* section, you can define the placement priority rank of each diagram, i.e. if there are different diagrams or labels candidates for the same location, the item with the higher priority will be displayed and the others could be left out.

Discourage diagrams and labels from covering features defines features to use as *obstacles*, i.e. QGIS will try to not place diagrams nor labels over these features. The priority rank is then used to evaluate whether a diagram could be omitted due to a greater weighted obstacle feature.

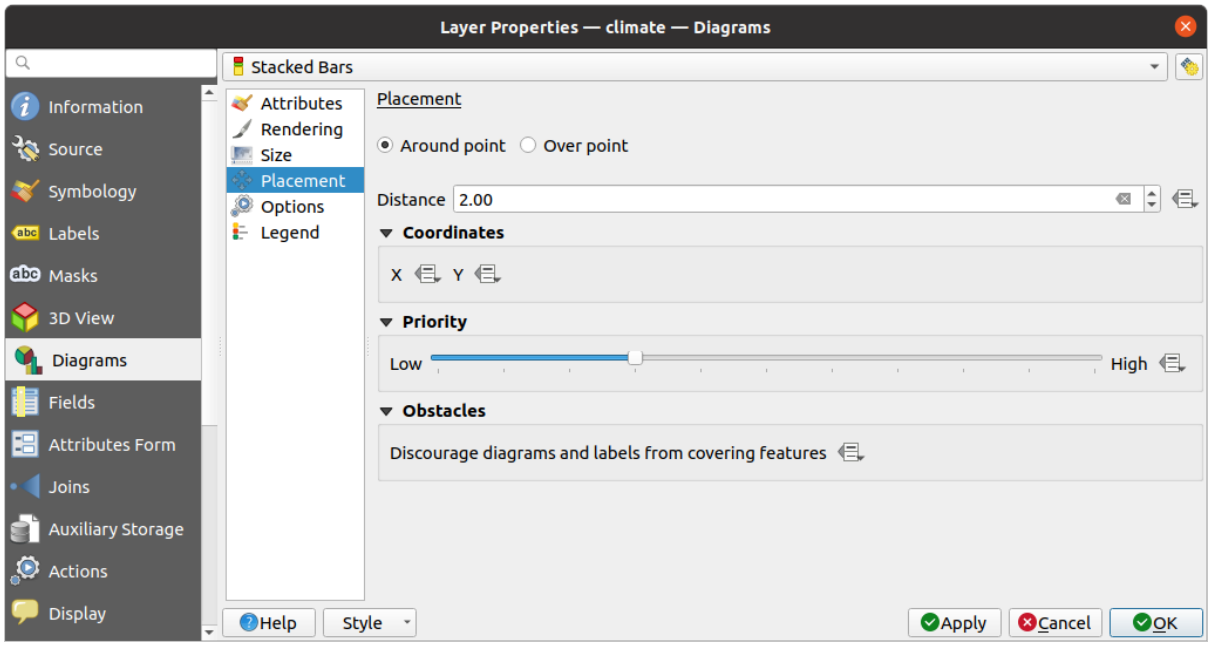


Figura12.37: Caixa de diálogo Propriedades do vetor com propriedades do diagrama, guia Posicionamento

Opções

The *Options* tab has settings for histograms and stacked bars. You can choose whether the *Bar orientation* should be *Up*, *Down*, *Right* or *Left*, for horizontal and vertical diagrams.

Legenda

From the *Legend* tab, you can choose to display items of the diagram in the *Layers panel*, and in the *print layout legend*, next to the layer symbology:

- Verifica:guilabel:*Mostrar entradas de legenda para os atributos do diagrama* para exibir nas legendas as propriedades` *Cor*` e` *Legenda*`, conforme previamente atribuído na guia: guilabel:` Atributos`;
- e, quando um:ref:*tamanho dimensionado<diagram_size>* estiver sendo usado para os diagramas, pressione o botão:guilabel:*Entradas de legenda para o tamanho do diagrama …* para configurar o aspecto do símbolo do diagrama nas legendas. Isso abre a caixa de diálogo:guilabel:*Legenda do tamanho definido por dados* cujas opções são descritas em: ref:` data_defined_size_legend`.

Quando definidos, os itens da legenda do diagrama (atributos com cor e tamanho do diagrama) também são exibidos na legenda do layout de impressão, ao lado da simbologia da camada.

Stacked Diagrams

Stacked diagrams allow users to create complex diagrams like population pyramids, where two subdiagrams, namely histograms, are located side by side and displayed horizontally.

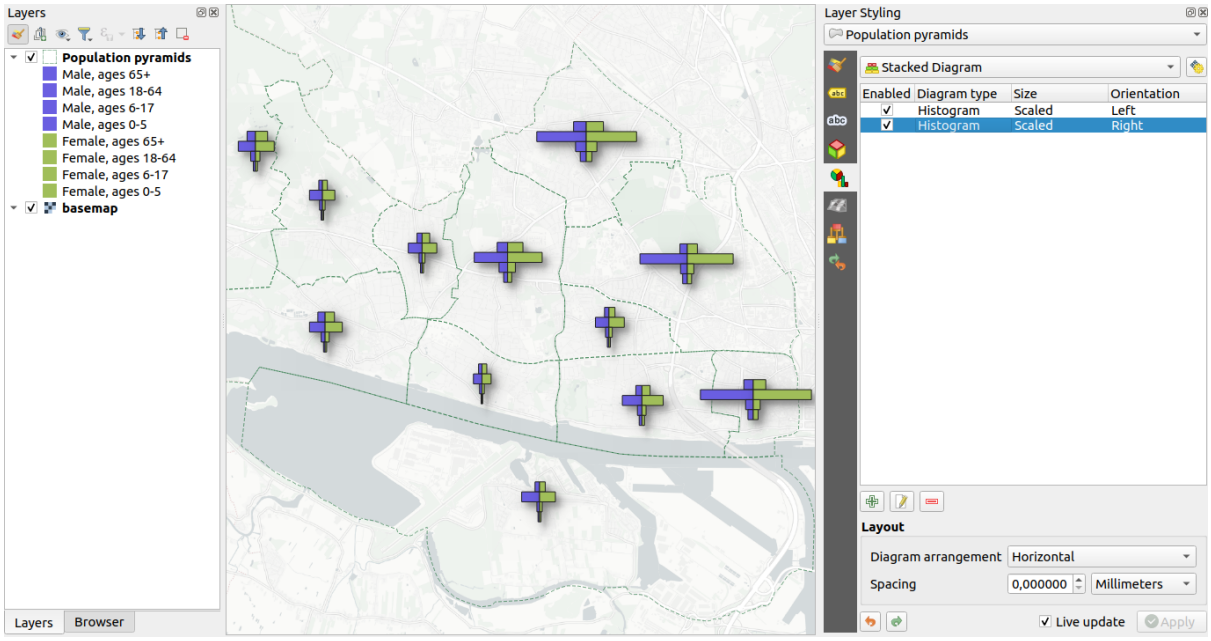


Figura12.38: Population pyramids built for each layer feature

Multi-temporal diagrams can also be constructed as stacked diagrams. The number of subdiagrams, as well as the spacing between them can be configured.

Moreover, subdiagrams can have different types (e.g., a pie chart alongside a histogram) and have their own independent settings like *Attributes*, *Rendering*, *Size*, *Options* and *Legend*.

Placement settings in a stacked diagram, as well as some visibility settings (located in the *Rendering* tab), are determined by the placement and visibility settings of the first subdiagram in the stack.

Finally, subdiagram ordering is given by the item ordering in the Stacked Diagram's list. The first subdiagram appears to the left in a horizontal stacked diagram, or in the upper part of a vertical one.

12.1.6 Propriedades das Máscaras

The *Masks* tab helps you configure the current layer symbols overlay with other symbol layers or labels, from any layer. This is meant to improve the readability of symbols and labels whose colors are close and can be hard to decipher when overlapping; it adds a custom and transparent mask around the items to "hide" parts of the symbol layers of the current layer.

To apply masks on the active layer, you first need to enable in the project either *mask symbol layers* or *mask labels*. Then, from the *Masks* tab, check:

- the *Masked symbol layers*: lists in a tree structure all the symbol layers of the current layer. There you can select the symbol layer item you would like to transparently "cut out" when they overlap the selected mask sources
- the *Mask sources* tab: list all the mask labels and mask symbol layers defined in the project. Select the items that would generate the mask over the selected masked symbol layers

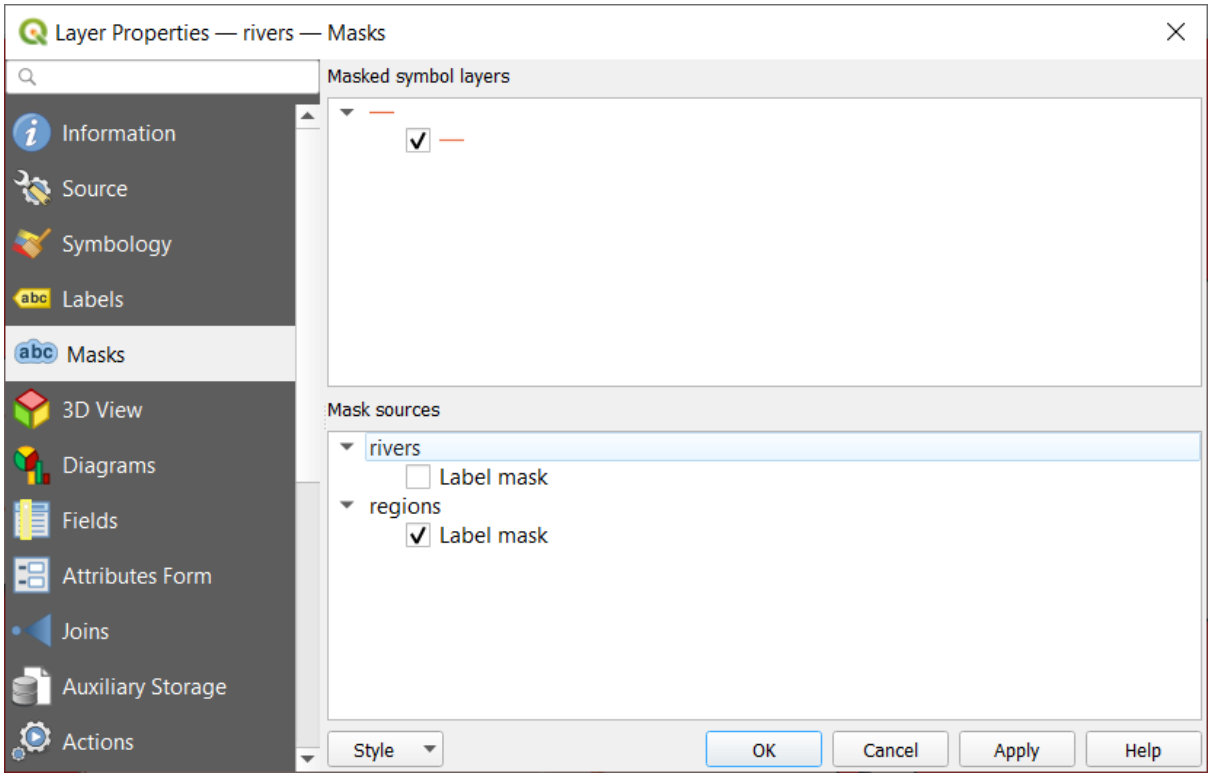


Figura12.39: Layer properties - Masks tab

12.1.7 Propriedades da vista 3D

The *3D View* tab provides settings for vector layers that should be depicted in the *3D Map view* tool.

Para exibir uma camada em 3D, selecione na caixa de combinação na parte superior da guia:

- *Single symbol*: features are rendered using a common 3D symbol whose properties can be *data-defined* or not. Read details on *setting a 3D symbol* for each layer geometry type.
- *Baseado em regras*: várias configurações de símbolos podem ser definidas e aplicadas seletivamente com base em filtros de expressão e escala de escala. Mais detalhes sobre como fazer em:ref:*Renderização baseada em regras<rule_based_rendering>*.

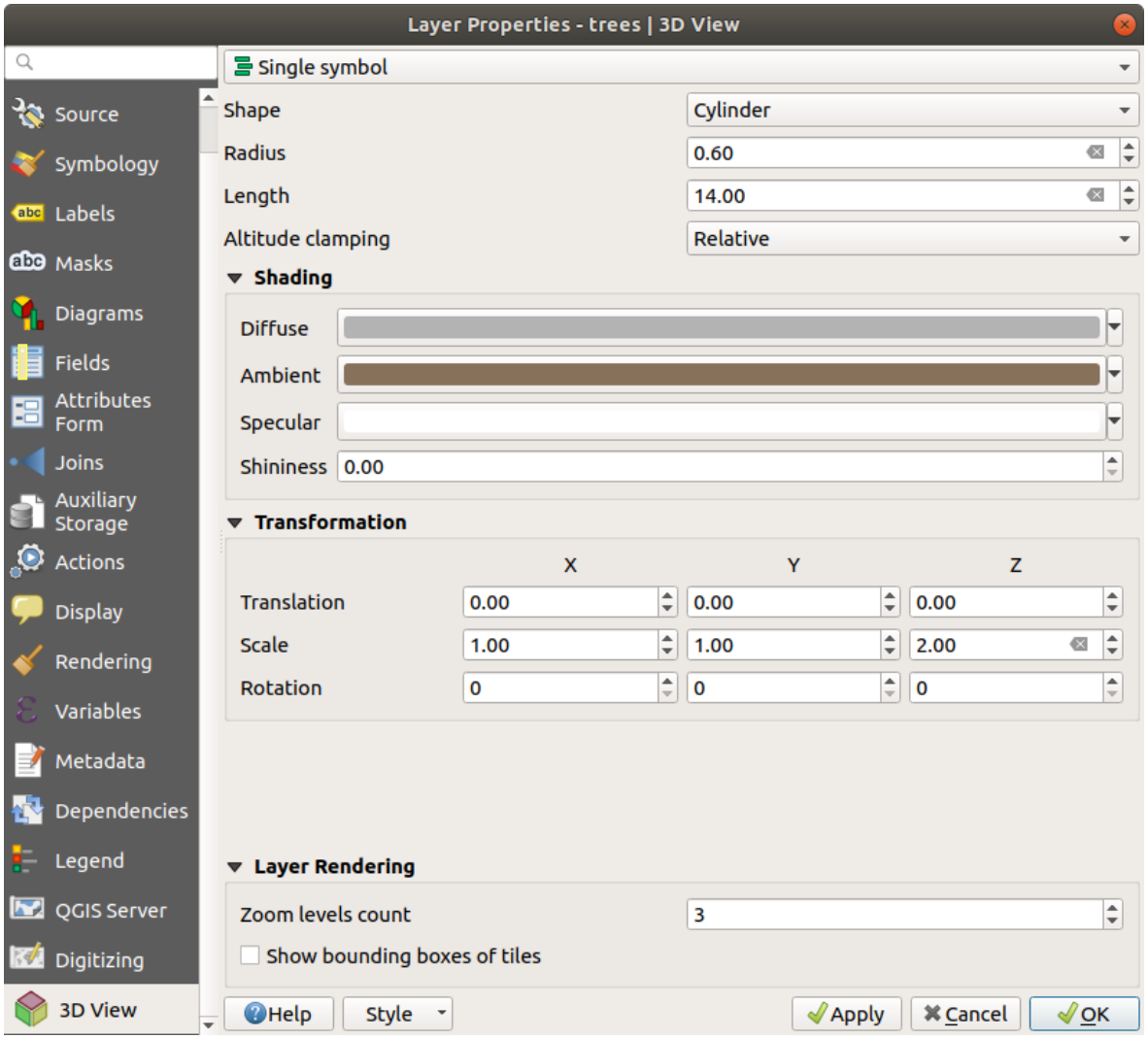


Figura12.40: Propriedades 3D de uma camada de ponto

Atenção

Prefer the *Elevation* tab for symbol elevation and terrain settings

Features' elevation and altitude related properties (*Altitude clamping*, *Altitude binding*, *Extrusion* or *Height*) in the *3D View* tab inherit their default values from the layer's *Elevation properties* and should preferably be set from within the *Elevation* tab.

For better performance, data from vector layers are loaded in the background, using multithreading, and rendered in tiles whose size can be controlled from the *Layer rendering* section of the tab:

- *Zoom levels count*: determines how deep the quadtree will be. For example, one zoom level means there will be a single tile for the whole layer. Three zoom levels means there will be 16 tiles at the leaf level (every extra zoom level multiplies that by 4). The default is `3` and the maximum is `8`.
- *Show bounding boxes of tiles*: especially useful if there are issues with tiles not showing up when they should.

12.1.8 Propriedades dos campos

The *Fields* tab provides information on fields related to the layer and helps you organize them.

A camada pode ser feita:ref:*editável<editingvector>* usando o botão de |alternância Edição | Alternar modo de edição. Neste momento, você pode modificar sua estrutura usando o | novo Atributo | Novo campo e | deletar Atributo | :sup:Botões *Excluir campo*.

When creating New field, the *Comment* option is available only for data sources that allow editing comments (See *Entradas no banco de dados* for more details). You can also set aliases within *Add Field* dialog, for supported OGR formats (GeoPackage and ESRI File Geodatabase).

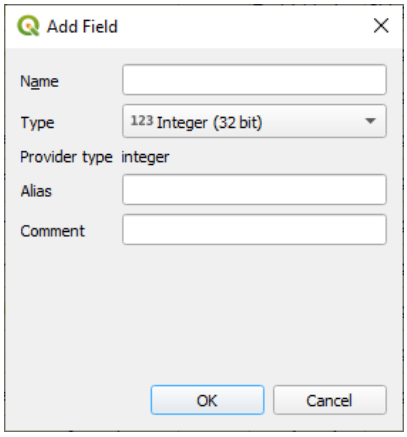


Figura12.41: Add Field Dialog

You can also rename fields by double-clicking its name. This is only supported for data providers like PostgreSQL, Oracle, Memory layer and some GDAL layers depending on the GDAL version.

Se definido na fonte de dados subjacente ou nas:ref:*propriedades dos formulários<configure_field>*, o alias do campo também é exibido. Um alias é um nome de campo legível por humanos que você pode usar no formulário de recurso ou na tabela de atributos. Os aliases são salvos no arquivo de projeto.

Other than the fields contained in the dataset, *virtual fields* and *Auxiliary Storage* included, the *Fields* tab also lists fields from any *joined layers*. Depending on the origin of the field, a different background color is applied to it.

For each listed field, the dialog also lists read-only characteristics such as its *Type*, *Type name*, *Length* and *Precision*`.

Dependendo do provedor de dados, você pode associar um comentário a um campo, por exemplo, na sua criação. Essas informações são recuperadas e mostradas na coluna:guilabel:*Comente* e mais tarde são exibidas ao passar o mouse sobre o rótulo do campo em um formulário de recurso.

Under the *Configuration* column, you can set how the field should behave in certain circumstances:

- `Not searchable`: check this option if you do not want this field to be queried by the *search locator bar*
- `Do not expose via WMS`: check this option if you do not want to display this field if the layer is served as WMS from QGIS server
- `Do not expose via WFS`: check this option if you do not want to display this field if the layer is served as WFS from QGIS server

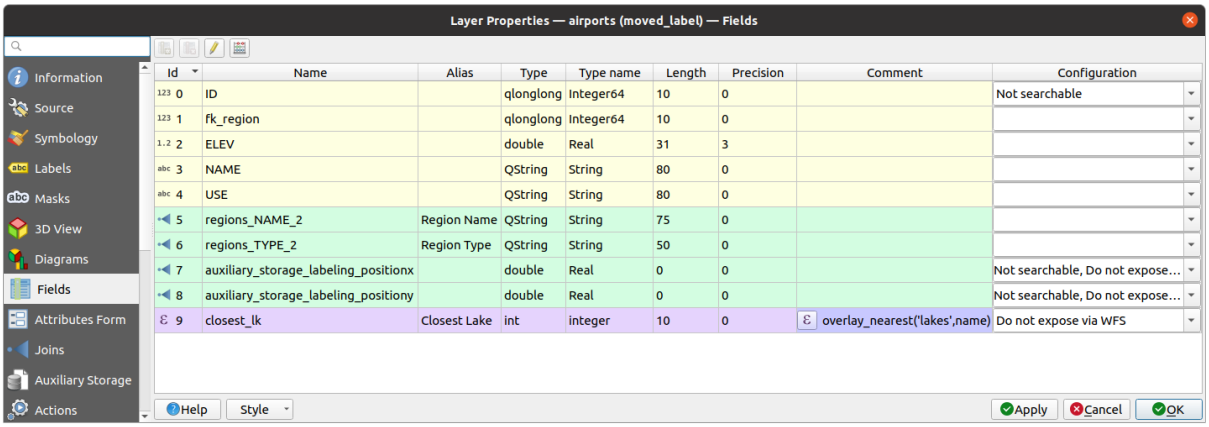


Figura12.42: Guia de propriedades dos campos

12.1.9 Propriedades do formulário de atributos

The *Attributes Form* tab helps you set up the form to display when creating new features, editing or querying existing one. This affects them in both *table and form views*. You can define:

- the look and the behavior (label, visibility, widget, constraints…) of each field in the current layer, the *joined layers* and the *relational ones*;
- the placement of enabled *actions*;
- the form's structure (custom or autogenerated);
- extra HTML, QML, text widgets to enhance the form;
- lógica extra no Python para lidar com a interação com os widgets de formulário ou campo.

No canto superior direito da caixa de diálogo, você pode definir se o formulário é aberto por padrão ao criar novos recursos. Isso pode ser configurado por camada ou globalmente com a opção:guilabel:*Suprimir pop-up do formulário de atributo após a criação do recurso* no menu: menuselection:` Settings -> Options -> Digitizing`.

Personalizando um formulário para seus dados

Por padrão, quando você clica em um recurso com a tecla | identificar | : sup: Ferramenta 'Identificar recursos` ou alternar a tabela de atributos para o modo * modo de exibição *, QGIS exibe um formulário básico com widgets predefinidos (geralmente caixas de rotação e caixas de texto — cada campo é representado em uma linha dedicada por seu rótulo próximo a o widget). Se:ref:*relações<vector_relations*> são definidas na camada, os campos das camadas de referência são mostrados em um quadro embutido na parte inferior do formulário, seguindo a mesma estrutura básica.

Esta renderização é o resultado do valor padrão *'Autogenerate`* da configuração:guilabel:'Atributo editor layout` na: seleção de menus:` Propriedades da camada -> Formulário de atributos. Esta propriedade possui três valores diferentes:

- *'Autogenerate`*: mantém a estrutura básica de" uma linha - um campo "para o formulário, mas permite personalizar cada widget correspondente.
- `Drag-and-drop designer`: other than widget customization, the form structure can be made more complex eg, with widgets embedded in groups and tabs.
- `` Fornecer arquivo ui``: permite usar um arquivo Qt designer, portanto, um template potencialmente mais complexo e completo, como forma de recurso.

Just below the *Attribute editor layout* setting, a search box allows you to filter fields, relations, containers and other available widgets.

For items that support aliases, like fields or relations, the filter targets both names and aliases.

The Show field aliases instead of names button allows you to toggle field aliases and field names.

When showing aliases, fields without an alias set will have their names displayed in light gray to help identify them easily.

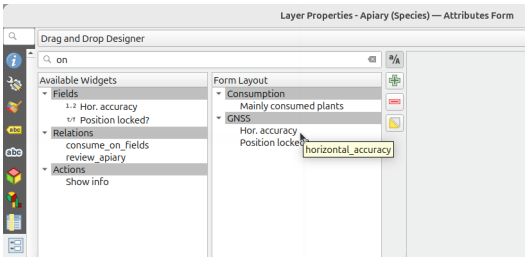


Figura12.43: Search fields using aliases and names in Attributes Form page

A forma autogerada

When the `Autogenerate` option is on, the *Available widgets* panel displays lists of fields (from the current layer, the join and relation layers) and the enabled actions that would be shown in the form. Select a field and you can configure its appearance and behavior in the right panel:

- adicionar:ref:*etiqueta personalizada e verificações automatizadas<configure_field>* ao campo;
- configurando um:ref:*widget particular<edit_widgets>* para usar.

In this editor mode, only the properties can be viewed for the actions. Moreover, they will be displayed through an *Actions* drop-down menu in the feature form, and following the *Show in attribute table* property in other contexts.

O designer de arrastar e soltar

The drag and drop designer allows you to create a form with several containers (tabs or groups) to present the attribute fields or other widgets that are not directly linked to a particular field (like the HTML/QML widgets or the *actions* defined for the layer), as shown for example in Fig.12.44.

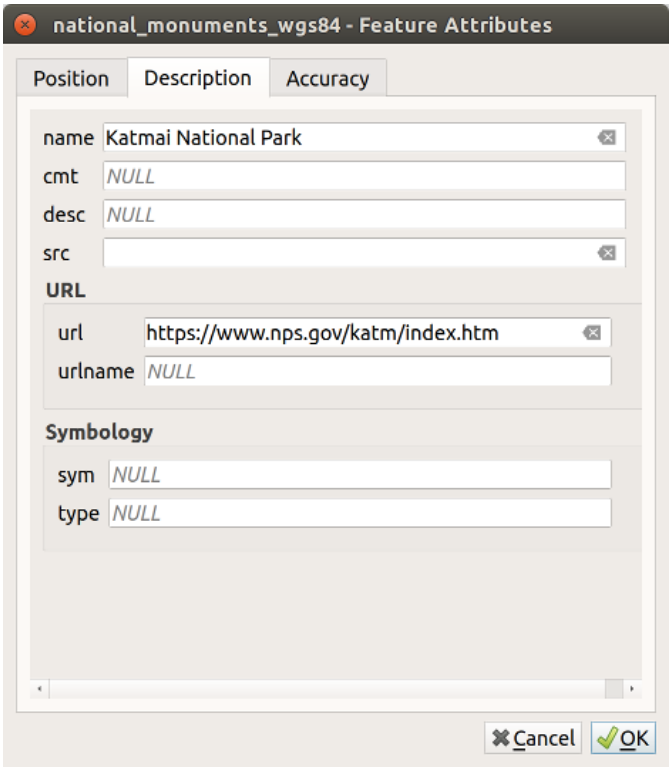


Figura12.44: Formulário integrado resultante com guias e grupos nomeados

1. Choose `Drag and drop designer` from the *Select attribute layout editor* combobox. This enables the *Form Layout* panel next to the *Available widgets* panel, filled with existing fields. The selected field displays its *properties* in a third panel.
2. Select fields you do not want to use in your *Form Layout* panel and hit the button to remove them. You can also toggle the selection with the Invert selection button.
3. Drag and drop fields from the first panel to the *Form Layout* one to re-add them. The same field can be added multiple times.
4. Arraste e solte os campos no painel:guilabel:*Layout de formulário* para reordenar suas posições.
5. Add containers to associate fields that belong to the same category and better structure the form.
    1. The first step is to use the Add a new tab or group to the form layout icon. Fields and other groups will be displayed in it.
    2. Then set the properties of the container, i.e.:
        - the *Label*: the title that will be used for the container
        - the *Container Type*: it can be a *Tab*, *Group box in container* (a collapsible group box inside a tab or another group) or a *Row* (a container type that allows you to arrange your widgets in a horizontal row, automatically determining the number of columns based on the number of widgets),
        - the *Within*: this optional feature allows you to select an existing container in which the new container (*Group box in container* or *Row*) will be embedded.
        - and the *Number of columns* the embedded fields should be distributed over

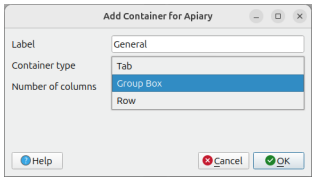


Figura12.45: Caixa de diálogo para criar contêineres com o ** Layout do editor de atributos **

Estas e outras propriedades podem ser atualizadas posteriormente selecionando o item e, no terceiro painel:

- ocultar ou mostrar o rótulo do contêiner
- renomear o contêiner
- definir o número de colunas
- insira uma expressão para controlar a visibilidade do contêiner. A expressão será reavaliada sempre que os valores no formulário mudarem, e a guia ou caixa de grupo mostrada / oculta de acordo
- *Show as Group Box*: converts a tab into a top-level group box and vice versa
- in case of a group box, set whether at the form opening, it should show as *Collapsed* for every features, or only for those matching an expression (*Control collapsed by expression*).
- configure the *Style* look of the container, with custom *Background color*, label color and font properties

3. You can create and embed as many containers as you want by pressing again the Add a new tab or group to the form layout icon.

6. The next step is to assign the relevant fields to each container, by simple drag and drop. Group boxes and tabs can also be moved the same way.
7. *Customize the widget* of the fields in use
8. In case the layer is involved in a *one or many to many relation*, drag-and-drop the relation name from the *Available Widgets* panel to the *Form Layout* panel. The associated layer attribute form will be embedded at the chosen place in the current layer's form. As for the other items, select the relation label to configure some properties:
   - ocultar ou mostrar o rótulo de relação
   - mostre o botão do link
   - mostre o botão de desvincular
9. In case the layer has one or more *actions* enabled for `Layer` or `Feature` scope, the actions will be listed under *Actions* and you can drag and drop them as with the other fields. The associated action will be embedded at the chosen place in the current layer's form.
10. Further customize the form by adding one or more widgets from *Other Widgets* (see *Other Widgets*)
11. mostre o botão de desvincular
12. Abra um formulário de atributo de feição (por exemplo, usando a ferramenta | identificar |:sup:'Identificar características`) e deve exibir o novo formulário.

Other Widgets

The drag and drop designer offers a number of widgets that are not connected to a particular field. They can be used to enhance the appearance of the form or to display dynamically calculated values.

- *HTML Widget*: embeds an HTML page, the HTML source may contain the result of dynamically calculated expressions.

  HTML widgets can be used for example to display images stored as BLOB in a field (let's call it `photo`):

  1. In the `Drag-and-drop designer` mode, add a *HTML Widget* to your *Form Layout*.
  2. Double-click on the *HTML Widget* to configure it.
  3. Change the default *Title* or hide it.
  4. Press the button and enter the following QGIS expression:

     ```
     '<img src= "' || to_base64("photo") || '">'
     ```

     Ensure that you replace *photo* with your own BLOB field name. The above expression creates a string with HTML image tag in which the BLOB file is encoded.
  5. Apply the dialog and then press the button.
  6. QGIS automatically applies HTML formatting and functions to evaluate your expression, resulting in following code:

     ```
     <script>document.write(expression.evaluate("'<img src=\"data:image/png;
     ↪base64,' || to_base64(\"photo\") || '\">'"));</script>
     ```

     A preview of your image is displayed on the right.

- *QML Widget*: embeds a Qt QML document, displaying graphical elements in the attribute form. Beside the custom *Title* of the added widget and whether it should be shown or not, you can select from predefined *QML code* elements:
  - *Free text...*: allows you to write from scratch or paste an existing code
  - *Rectangle*: provides a minimal code for displaying a rectangle
  - *Bar chart*: provides a minimal code for displaying a bar chart
  - *Pie chart*: provides a minimal code for displaying a pie chart

  You can extend the code with QML syntax, use layer fields or QGIS expressions that are dynamically calculated.

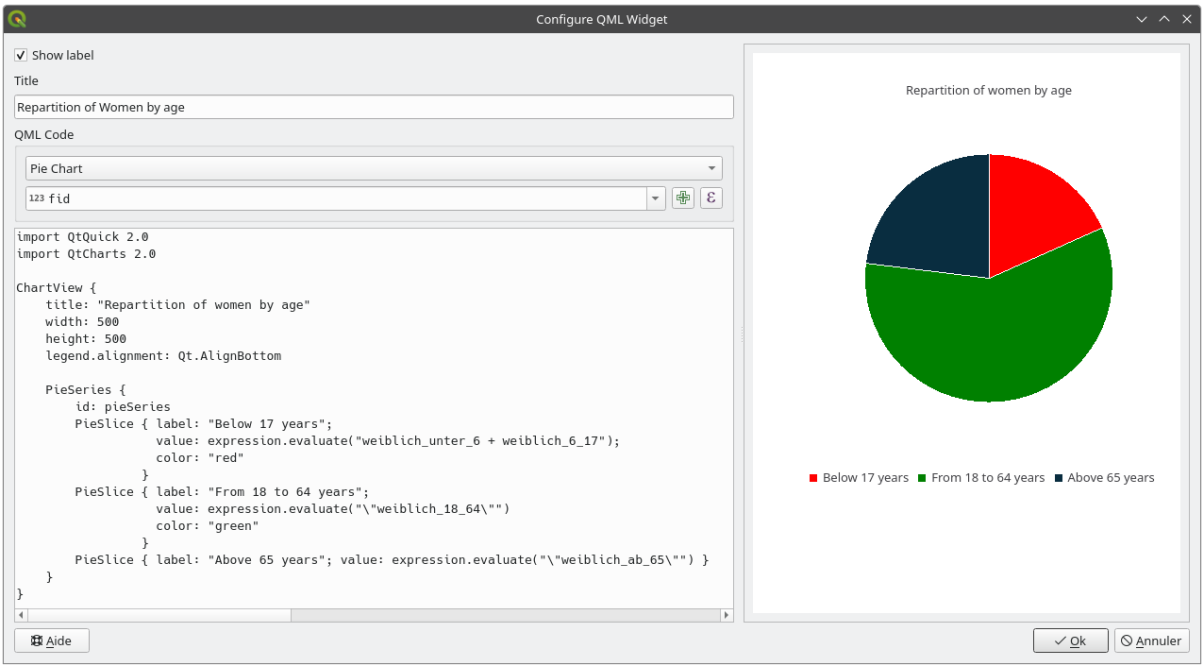


Figura12.46: Setting a QML graph to display in attribute form

- *Text Widget*: displays a text widget which supports basic HTML markup and may contain the result of dynamically calculated expressions.
- *Spacer Widget*: inserts an empty transparent rectangle, increasing the vertical distance between two widgets.

Dica

Display Dynamic Content

The widgets mentioned above (except the *Spacer Widget*) support expressions that can be used to display content that dynamically changes whenever another field in the form changes. This can be achieved using the `current_value('field_name')` function in the expression to examine the value of another field.

Usando arquivo ui personalizado

A opção `` Provide ui-file`` permite que você use diálogos complexos feitos com o Qt-Designer. Usar um arquivo de interface do usuário permite uma grande liberdade na criação de um diálogo. Observe que, para vincular os objetos gráficos (caixa de texto, caixa de combinação …) aos campos da camada, você precisa dar a eles o mesmo nome.

Use:guilabel:‘Editar IU` para definir o caminho para o arquivo a ser usado.

Os arquivos da IU também podem ser hospedados em um servidor remoto. Neste caso, você fornece a URL do formulário em vez do caminho do arquivo em: guilabel: ‘Editar UI`.

Você encontrará alguns exemplos na:ref:*Criando um novo formulário<creating-new-form>* lição da:ref:*QGIS-training-manual-index-reference*. Para obter informações mais avançadas, consulte https://woostuff.wordpress.com/2011/09/05/qgis-tips-custom-feature-forms-with-python-logic/.

Aprimore seu formulário com funções personalizadas

Os formulários QGIS podem ter uma função Python que é chamada quando a caixa de diálogo é aberta. Use esta função para adicionar lógica extra aos seus diálogos. O código do formulário pode ser especificado de três maneiras diferentes:

- `` carregar do ambiente``: use uma função, por exemplo em: file: *startup.py* ou de um plugin instalado
- `` carregar de um arquivo externo``: um seletor de arquivos permitirá que você selecione um arquivo Python de seu sistema de arquivos ou insira uma URL para um arquivo remoto.
- `` fornecer código nesta caixa de diálogo``: um editor Python aparecerá onde você pode digitar diretamente a função a ser usada.

Em todos os casos, você deve inserir o nome da função que será chamada (*'open`* no exemplo abaixo).

Um exemplo é (no módulo MyForms.py):

```
def open(dialog,layer,feature):
    geom = feature.geometry()
    control = dialog.findChild(QWidget,"My line edit")
```

Referência na função Python Init como: *'open`*

Configure o comportamento do campo

A parte principal da aba:guilabel:'Formulário de Atributos` ajuda a definir o tipo de widget usado para preencher ou exibir os valores do campo, na tabela de atributos ou no formulário de características: você pode definir como o usuário interage com cada campo e o valores ou intervalo de valores que podem ser adicionados a cada um.

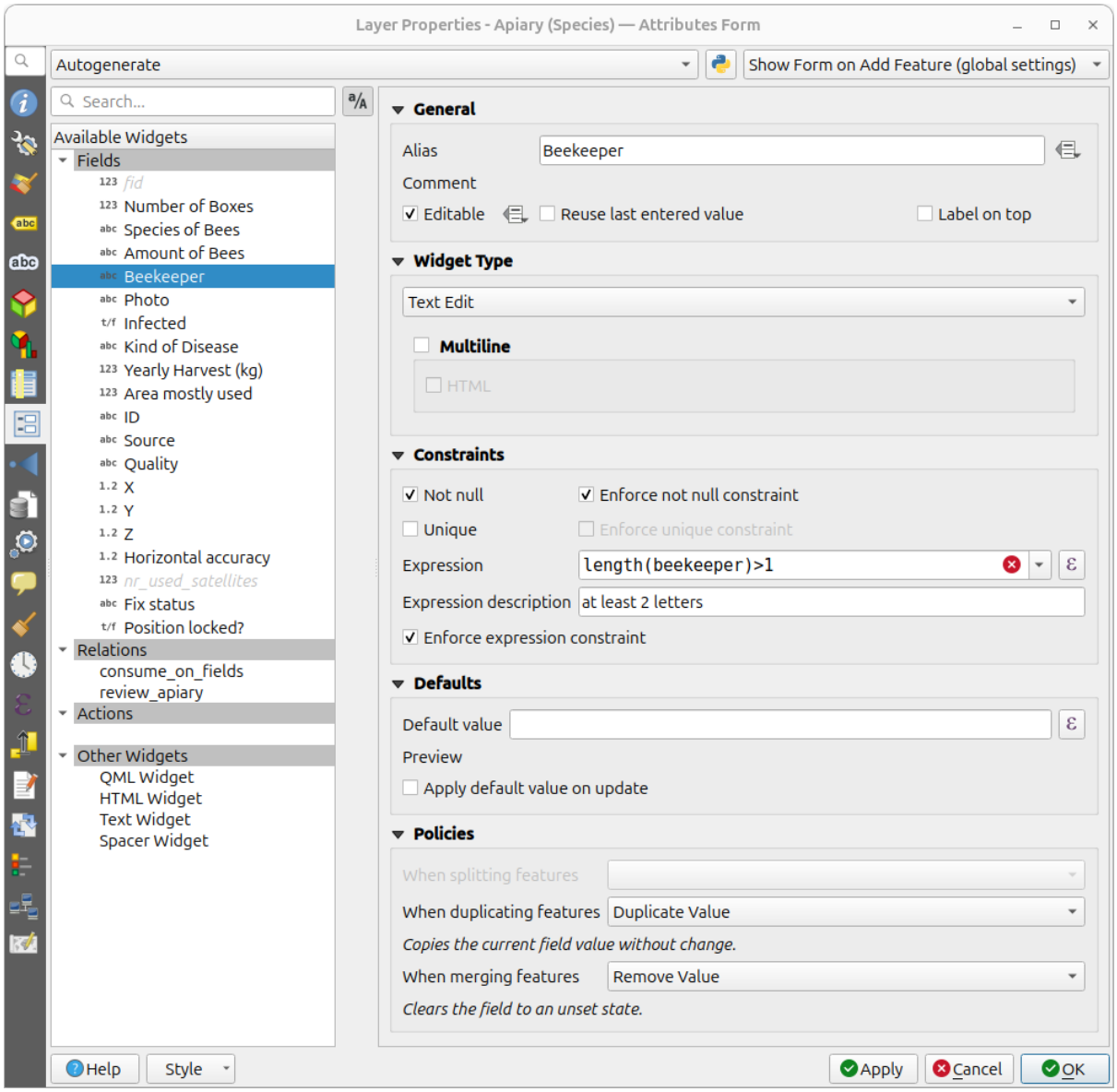


Figura12.47: Caixa de diálogo para selecionar um widget de edição para uma coluna de atributo

Dica

Copy and paste widget configuration

The field configuration can be copied to the clipboard and then pasted to another field via right click on a field in the *Available widgets* panel.

This comes handy to save time and to make sure that common settings are correctly applied to several related fields, e.g., when configuring latitude and longitude field widgets.

Copy and paste of widget configuration can be done between fields of the same layer, between fields from different layers in the QGIS project, or between fields from layers in different QGIS instances.

Note that neither aliases nor comments are copied, since those are specific to each particular field.

Configurações padrão

Independentemente do tipo de widget aplicado ao campo, existem algumas propriedades comuns que você pode definir para controlar se e como um campo pode ser editado.

Tela de widget

Only available for the *Drag and drop* designer mode, this group helps you configure the look of the widget assigned to the field:

- *Show label*: indicates whether the field name should be displayed in the form
- *Override label color*: applies specific color to the field's label
- *Override label font*: applies specific font properties (bold, italic, underline, strikethrough, or font family) to the field's label
- *Size*: allows to control how widgets will relatively resize when resizing an attribute form.
  - *Horizontal stretch*: sets a higher horizontal value for widgets that need more horizontal space.
  - *Vertical stretch*: determines how widgets resize vertically when the form is resized.

Opções gerais

- *Alias*: a human readable name to use for fields. The alias will be displayed in the feature form, the attribute table, or in the *Identify results* panel. It can also be used as field name replacement in the *expression builder*, easing expressions understanding and reviews. Aliases are saved in project file.
- *Comentário*: exibe o comentário do campo como mostrado na aba: guilabel:` Campos`, em um estado somente leitura. Essas informações são mostradas como dica de ferramenta ao passar o mouse sobre o rótulo do campo em um formulário de recurso.
- ☑ *Editable*: uncheck this option to set the field read-only (not manually modifiable) even when the layer is in edit mode. Note that checking this setting doesn't override any edit limitation from the provider. This option can be controlled by a data-defined property thanks to the Data-defined override button.
- ☑ *Reuse last entered value*: remembers the last value entered in this field and uses it as default for the next feature being edited in the layer.
- ☑ *Label on top*: places the field name above or beside the widget in the feature form.

Restrições

Você pode restringir o valor a ser inserido no campo. Essa restrição pode ser:

- ☑ *Not null*: requires the user to provide a value;
- ☑ *Unique*: guarantee the inserted value to be unique throughout the field;
- based on a custom *expression*: e.g. `not regexp_match(col0,'[^A-Za-z]')` will ensure that the value of the field *col0* has only alphabet letters. A short description can be added to help you remember the constraint.

Sempre que um valor é adicionado ou editado em um campo, ele é submetido às restrições existentes e:

- se atender a todos os requisitos, uma marca de seleção verde é exibida ao lado do campo do formulário;
- if it does not meet all the requirements, then the field is colored in yellow or orange and a corresponding cross is displayed next to the widget. You can hover over the cross to remind which constraints are applied to the field and fix the value:

  - A yellow cross appears when the unmet constraint is an unenforced one (`soft constraint`) and it does not prevent you to save the changes with the "wrong" values;
  - An orange cross can not be ignored and does not allow you to save your modifications until they meet the constraints. It appears when the *Enforce constraint* option is checked (`hard constraint`).

Valores padrão

- *Valor padrão*: para novos recursos, preenche automaticamente por padrão o campo com um valor pré-definido ou um: ref:` baseado em expressão<vector_expressions> `. Por exemplo, você pode:
  - use `` $ x``, `` $ length``, `` $ area`` para preencher automaticamente um campo com a coordenada X do elemento, comprimento, área ou qualquer informação geométrica em sua criação;
  - incrementa um campo em 1 para cada novo recurso usando `` maximum ("field") + 1``;
  - salve a data e hora de criação do recurso usando `` now () ``;
  - use:ref:*variáveis<general_tools_variables>* em expressões, tornando mais fácil, por exemplo, insira o nome do operador (`` @ user_full_name``), o caminho do arquivo do projeto (`` @ project_path``), …

  Uma visualização do valor padrão resultante é exibida na parte inferior do widget.

  Nota

  A opção `` Valor padrão`` não tem conhecimento dos valores em qualquer outro campo do recurso que está sendo criado, então não será possível usar uma expressão combinando qualquer um desses valores, ou seja, usando uma expressão como `` concat (campo1, field2) `` pode não funcionar.

- *Apply default value on update*: whenever the feature attribute or geometry is changed, the default value is recalculated. This could be handy to save values like last user that modifies data, last time it was changed…

Policies

Field policies determine how values are assigned to fields during various editing operations:

Split and Duplicate Policies

These policies apply when *Splitting features* or *Duplicating features*:

- *Duplicate Values*: Keeps the existing value of the field for the new features.
- *Use Default Value*: Resets the field by recalculating its *default value*. If no default value clause exists, the existing value is kept for the new features.
- *Remove Value*: Clears the field to an unset state.
- *Use Ratio Geometries*: Recalculates the field value for all split portions by multiplying the existing value by ratio of the split parts lengths or areas.

Merge Policies

These policies determine initial values when *Merging features*:

- *Remove Value*: Clears the field to an unset state (data provider may populate default value).
- *Use Default Value*: Uses the default field value set in QGIS. (expressions use a dummy feature with the merged geometry; references to other fields won't work).
- *Use Sum* (numeric fields only): Sums values from all merged features.
- *Use Minimum Value* (numeric fields only): Uses smallest value from the selected features.
- *Use Maximum Value* (numeric fields only): Uses largest value from the selected features.

- *Average Weighted by Geometry* (numeric fields only): New values are computed as the weighted average of the source values.
- *Use Largest Feature*: Uses value from feature with the largest geometry.
- *Set to NULL*: Sets attribute to NULL.

Editar widgets

Com base no tipo de campo, o QGIS determina automaticamente e atribui um tipo de widget padrão a ele. Você pode então substituir o widget por qualquer outro compatível com o tipo de campo. Os widgets disponíveis são:

- **Binary (BLOB)**: Available only for binary fields, it displays by default a label with the size of the embedded data, if not empty. A drop-down button next to the label allows to:
  - *Embed file*, replacing or filling the field
  - *Clear contents*, removing any data in the field
  - *Save contents to file*, exporting the data as a file on disk

  It is also possible to preview the embedded binary file in the field, if combined in a drag-and-drop form with e.g. a *QML or HTML widget*.
- **Checkbox**: Displays a checkbox whose state defines the value to insert. Internally, the widget stores a value (e.g. 1 for checked, 0 for unchecked) while representing it visually as a toggled box. You can configure the stored values for the checked and unchecked states in the representation. It is also possible to display the checkbox state as a plain text using "True"/"False" labels. These settings influence how the values are shown in classification renderers.
- ** Classificação **: Disponível apenas quando uma:ref:*simbologia categorizada<categorized_renderer>* é aplicada à camada, exibe uma caixa de combinação com os valores das classes.
- ** Cor **: Exibe um:ref:*widget de cor<color-selector>* permitindo selecionar uma cor; o valor da cor é armazenado como uma notação html na tabela de atributos.
- ** Data / Hora **: Exibe um campo de linha que pode abrir um widget de calendário para inserir uma data, hora ou ambos. O tipo de coluna deve ser texto. Você pode selecionar um formato personalizado, abrir um calendário, etc.
- ** Enumeração **: Abre uma caixa de combinação com valores predefinidos buscados no banco de dados. Isso é atualmente suportado apenas pelo provedor PostgreSQL, para campos do tipo *'enum`*.
- **Attachment**: Uses a "Open file" dialog to store file path in a relative or absolute mode. It can be used to display the document path as a hyperlink or render the document within a dedicated widget in the form. Supported document types are image, web page, audio and video, and the supported file formats depend on the Operating System. You can also configure an *external storage system* to fetch/store resources.

  > **Dica**
  >
  > ** Caminho relativo no widget de anexo **
  >
  > Se o caminho selecionado com o navegador de arquivos estiver localizado no mesmo diretório do arquivo de projeto: file: *.qgs* ou abaixo, os caminhos são convertidos em caminhos relativos. Isso aumenta a portabilidade de um projeto: file: *.qgs* com informações multimídia anexadas.

- **Geometry**: Compatible with the *Geometry* field type. For example, PostgreSQL and Memory layers can contain multiple geometry fields. QGIS uses the first geometry column as the layer geometry. Applying the Geometry widget to additional geometry fields allows lossless use of the data in the attribute form:
  - Copy in the formats WKT (Well-known text representation, e.g. `POINT (30 10)`) or GeoJSON
  - Paste
  - Clear

- ** Oculto **: Uma coluna de atributo oculto é invisível. O usuário não é capaz de ver o seu conteúdo.
- ** Chave / valor **: exibe uma tabela de duas colunas para armazenar conjuntos de pares de chave / valor em um único campo. Isso é atualmente suportado pelo provedor PostgreSQL, para campos do tipo `` hstore``.
- **JSON View**: Displays JSON data in a syntax highlighted text edit or in tree view. This widget is currently read only. Several options are available to change how the data is displayed. 'Default view' specify if the widget should appear in Text or Tree mode. 'Format JSON' has three options which are related to the tree view only:
  - Indented: Display data in a human readable form with newlines and four space characters for indentation.
  - Compact: Display data in a one-line size optimized string without newlines or spaces.
  - Disabled: Display data as it comes from the provider.
- ** Lista **: exibe uma única tabela de coluna para adicionar diferentes valores em um único campo. Isso é atualmente suportado pelo provedor PostgreSQL, para campos do tipo *'array`*.
- **Tamanho**: Permite definir valores numéricos de um tamanho específico. A edição do Widget pode ser um controle deslizante ou uma caixa de rotação.

> **Nota**
>
> Some layers, such as GeoPackage or ESRI File Geodatabase, with predefined **range Field Domains** will be automatically recognized by QGIS and assigned a **Range** widget for the relevant fields. The widget will be prefilled with the minimum and maximum values specified in the domain.

- **Relation Reference**: This is the default widget assigned to the referencing field (i.e., the foreign key in the child layer) when a *relation* is set. It provides direct access to the parent feature's form which in turn embeds the list and form of its children. The number of entries in the widget can be limited for efficiency, and if limit is not set, all entries will be loaded. The field defined in the display expression is used as representable value in classification renderers.
- ** Edição de texto ** (padrão): Isso abre um campo de edição de texto que permite o uso de texto simples ou várias linhas. Se você escolher várias linhas, também poderá escolher o conteúdo html.
- ** Valores únicos **: Você pode selecionar um dos valores já usados na tabela de atributos. Se 'Editável' estiver ativado, uma edição de linha é mostrada com suporte para preenchimento automático, caso contrário, uma caixa de combinação é usada.
- ** Gerador Uuid **: Gera um campo UUID (Universally Unique Identifiers) somente leitura, se vazio.
- **Value Map**: A combo box with predefined items. The value is stored in the attribute, the description is shown in the combo box. You can define values manually or load them from a layer or a CSV file. The description is used as the representable value, meaning it will be shown in classification renderers.

> **Nota**
>
> Some layers, such as GeoPackage or ESRI File Geodatabase, with predefined **coded Field Domains** will be automatically recognized by QGIS and assigned a **Value Map** widget for the relevant fields.

- **Value Relation**: Offers values from a related table in a combobox. You can select layer, key column and value column. The key column is the stored value in the attribute table, while the value column is the label shown in the combobox. Value column is also the representable value that will be shown in the classification renderers. You may also choose to order the list of displayed values by the *Key* column (the default sort order), the *Value* column, or a specific *Field* of the layer. Enable the *Descending order* if you wish the values to be sorted in reverse order.

  Several options are available to change the standard behavior: allow NULL value, allow multiple selections and use of auto-completer. The forms will display either a drop-down list or a line edit field when completer checkbox is enabled.

If a layer that is stored in PostgreSQL, GeoPackage or SpatiaLite is configured to use a value relation widget, but the required layer is not already loaded into the project, QGIS will automatically search for the layer in the same database/connection.

12.1.10 Propriedades de associações

The *Joins* tab allows you to associate features of the current layer to features from another loaded vector layer (or table). The join is based on an attribute that is shared by the layers, in a one-to-one relationship. For more details on joins, please read *Joining features between two layers*.

12.1.11 Propriedades de armazenamento auxiliar

A maneira normal de personalizar o estilo e a rotulação é usar propriedades definidas por dados conforme descrito em:ref:*data_defined*. No entanto, pode não ser possível se os dados subjacentes forem somente leitura. Além disso, configurar essas propriedades definidas por dados pode consumir muito tempo ou não ser desejável! Por exemplo, se você deseja usar totalmente as ferramentas de mapa que vêm com:ref:*label_toolbar*, então você precisa adicionar e configurar mais de 20 campos em sua fonte de dados original (posições X e Y, ângulo de rotação, estilo da fonte, cor e em breve).

O mecanismo de armazenamento auxiliar fornece a solução para essas limitações e configurações inadequadas. Os campos auxiliares são uma forma indireta de gerenciar e armazenar automaticamente essas propriedades definidas por dados (rótulos, diagrama, simbologia …) em um banco de dados SQLite, graças às junções editáveis. Isso permite que você armazene propriedades para camadas que não são editáveis.

Uma guia está disponível na caixa de diálogo de propriedades da camada vetorial para gerenciar o armazenamento auxiliar:

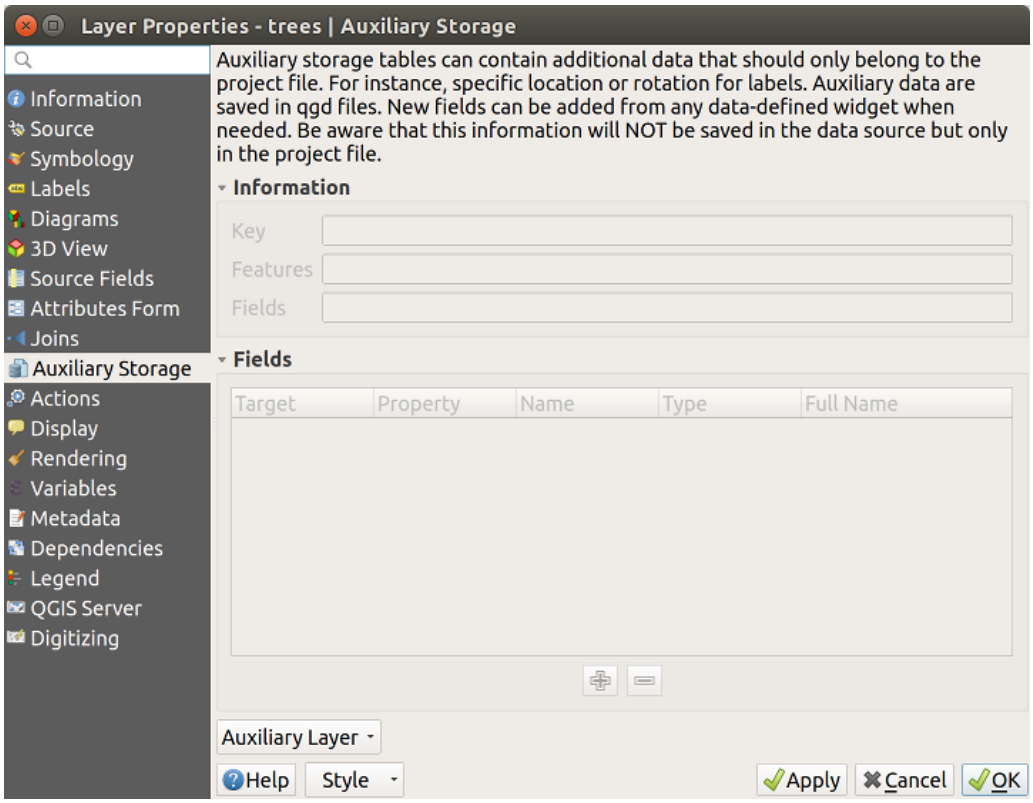


Figura12.48: Guia de armazenamento auxiliar

Marcação

Considerando que a fonte de dados pode ser personalizada graças às propriedades definidas pelos dados sem ser editável, as ferramentas de mapa de rotulagem descritas em:ref:*label_toolbar* estão sempre disponíveis assim que a rotulagem é ativada.

Na verdade, o sistema de armazenamento auxiliar precisa de uma camada auxiliar para armazenar essas propriedades em um banco de dados SQLite (ver: ref: *vector_auxiliary_storage_database*). Seu processo de criação é executado na primeira vez que você clica no mapa enquanto uma ferramenta de mapa de rotulagem está ativada no momento. Em seguida, uma janela é exibida, permitindo que você selecione a chave primária a ser usada para ingressar (para garantir que os recursos sejam identificados exclusivamente):

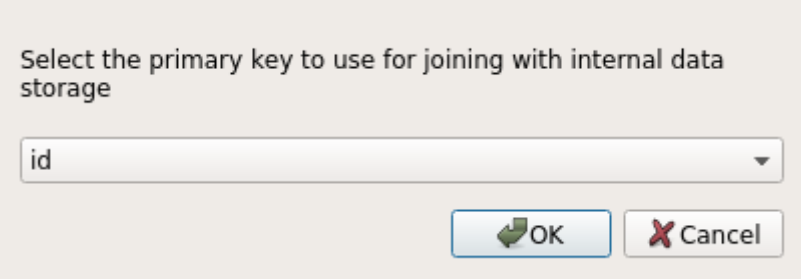


Figura12.49: Caixa de diálogo de criação de camada auxiliar

Assim que uma camada auxiliar é configurada para a fonte de dados atual, você pode recuperar suas informações na guia:

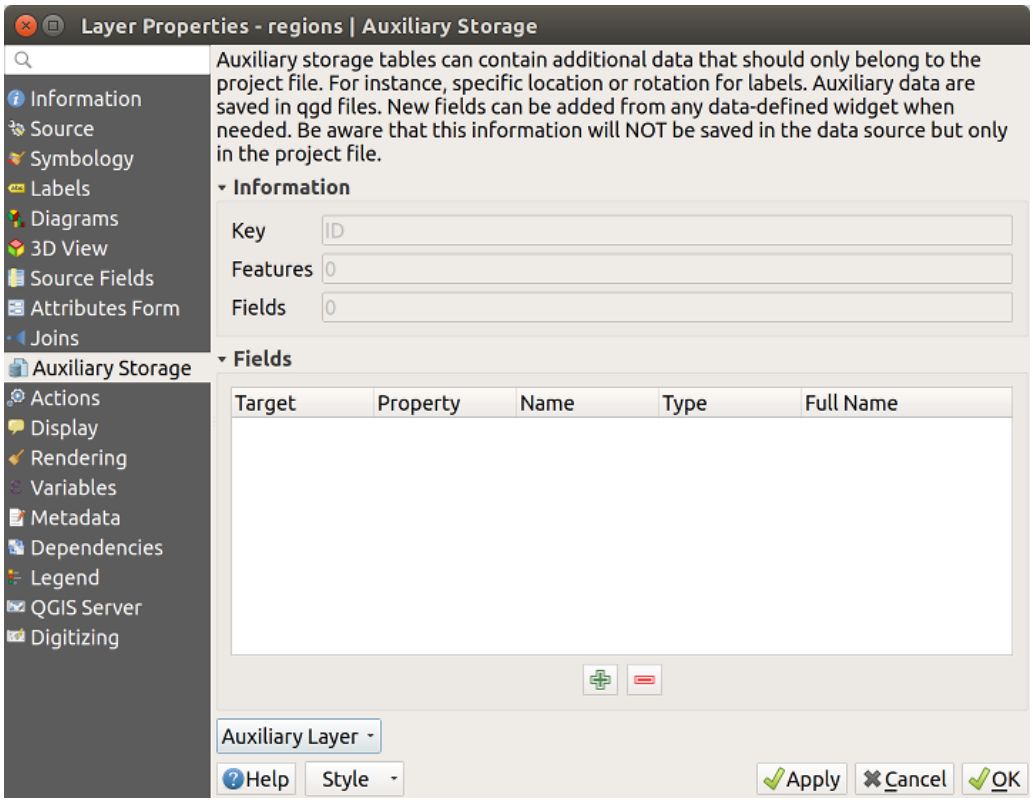


Figura12.50: Chave da camada auxiliar

A camada auxiliar agora tem estas características:

- a chave primária é `` ID``,

- existem recursos `` 0`` usando um campo auxiliar,
- existem campos auxiliares `` 0``.

Agora que a camada auxiliar foi criada, você pode editar os rótulos da camada. Clique em um rótulo enquanto o | changeLabelProperties | :sup:a ferramenta de mapa *Change Label* está ativada, então você pode atualizar as propriedades de estilo como tamanhos, cores e assim por diante. As propriedades definidas por dados correspondentes são criadas e podem ser recuperadas:

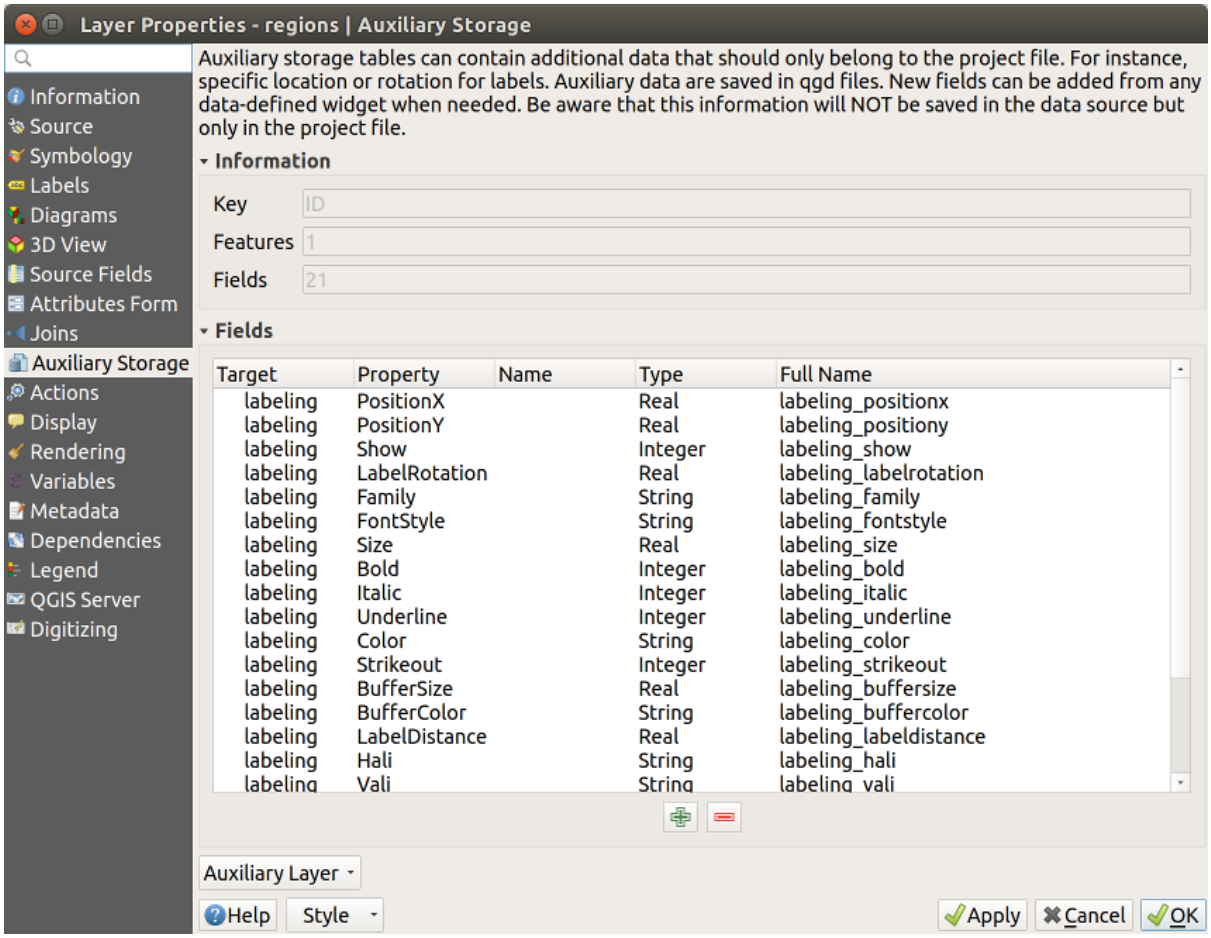


Figura12.51: Campos Auxiliares

Como você pode ver na figura acima, os campos `` 21`` são criados e configurados automaticamente para rotulagem. Por exemplo, o tipo de campo auxiliar `` FontStyle`` é um `` String`` e é denominado `` labeling_fontstyle`` no banco de dados SQLite subjacente. Há também o recurso `` 1`` que está usando esses campos auxiliares.

Observe que o ícone | dataDefineOn | é exibido na guia de propriedades: guilabel: *Labels* indicando que as opções de substituição definidas pelos dados estão configuradas corretamente:

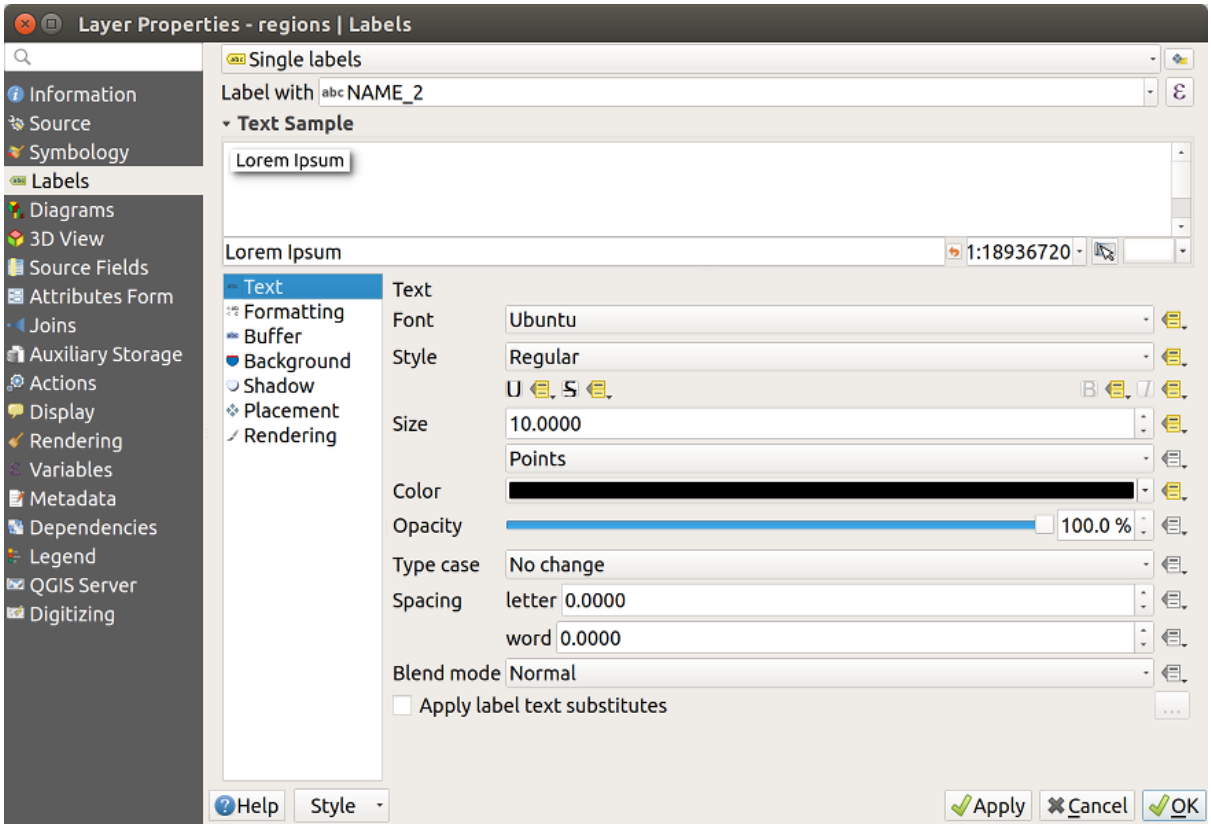


Figura12.52: Propriedades definidas por dados criadas automaticamente

Caso contrário, existe outra maneira de criar um campo auxiliar para uma propriedade específica, graças ao | dataDefine | :sup:botão *data-defined override*. Ao clicar em:guilabel:*Armazenar dados no projeto*, um campo auxiliar é criado automaticamente para o campo: guilabel:` Opacidade`. Se você clicar neste botão e a camada auxiliar ainda não tiver sido criada, uma janela (: numref: *figure_auxiliary_layer_creation*) é primeiro exibida para selecionar a chave primária a ser usada para ingressar.

Simbologia

Como o método descrito acima para personalizar rótulos, os campos auxiliares também podem ser usados para estilizar símbolos e diagramas. Para fazer isso, clique em | dataDefine | [Substituição definida por dados] e selecione: guilabel:` Armazenar dados no projeto` para uma propriedade específica. Por exemplo, o campo:guilabel:*Cor de preenchimento*:

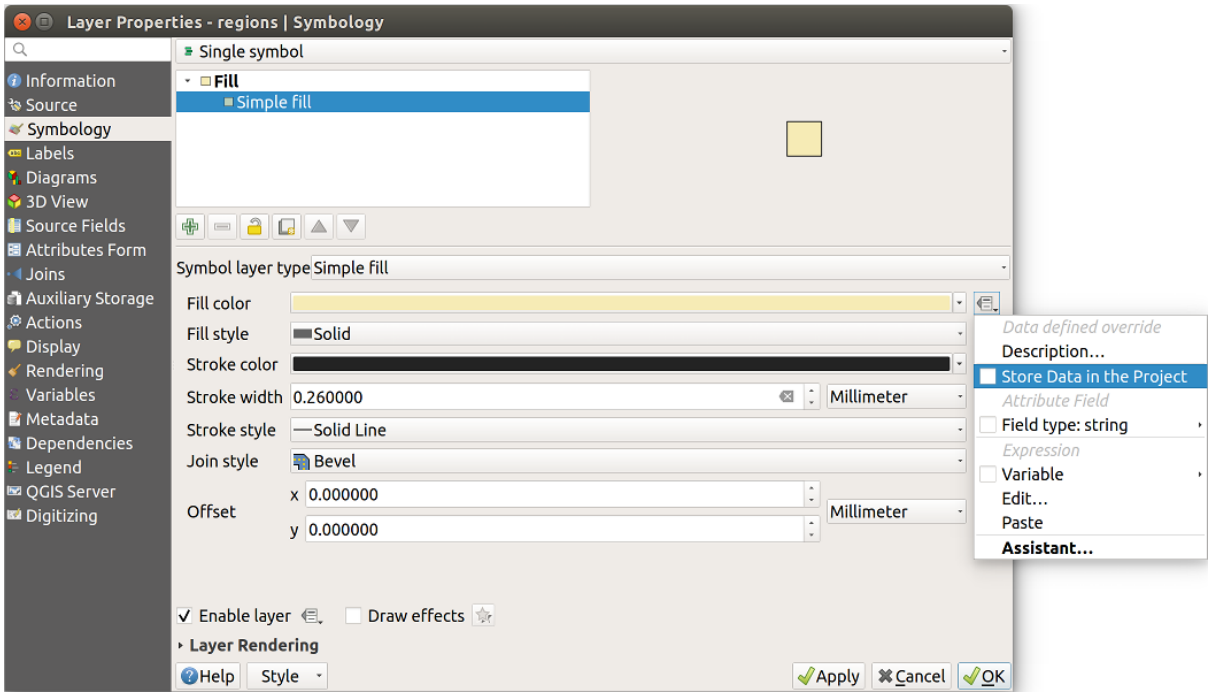


Figura12.53: Menu de propriedades definidas por dados para o símbolo

Existem diferentes atributos para cada símbolo (por exemplo, estilo de preenchimento, cor de preenchimento, cor do traço, etc …), portanto, cada campo auxiliar que representa um atributo requer um nome exclusivo para evitar conflitos. Após selecionar:guilabel:*Armazenar dados no projeto*, uma janela abre e exibe o: guilabel:` Tipo` do campo e solicita que você insira um nome exclusivo para o campo auxiliar. Por exemplo, ao criar um campo auxiliar:guilabel:*Cor de preenchimento*, a seguinte janela é aberta:

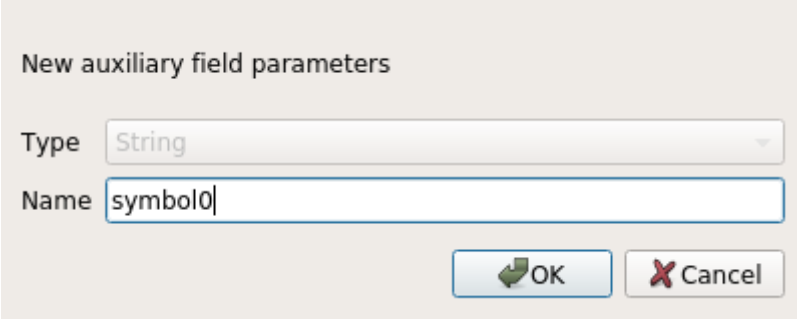


Figura12.54: Nome do campo auxiliar para um símbolo

Depois de criado, o campo auxiliar pode ser recuperado na guia de armazenamento auxiliar:

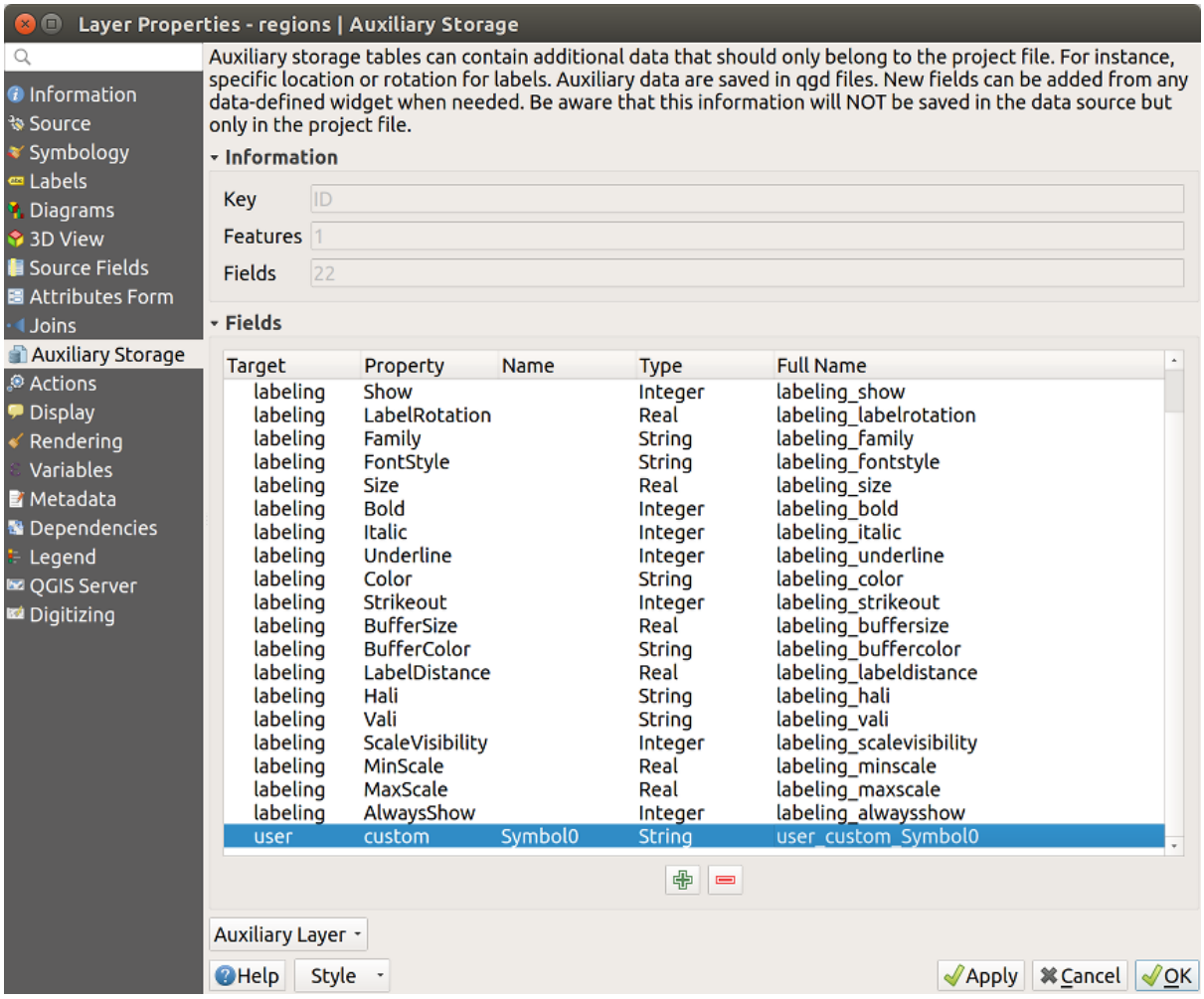


Figura12.55: Símbolo de campo auxiliar

Tabela de atributos e widgets

Os campos auxiliares podem ser editados usando a:ref:*tabela de atributos<sec_attribute_table>*. No entanto, nem todos os campos auxiliares são inicialmente visíveis na tabela de atributos.

Os campos auxiliares que representam os atributos da simbologia, rotulagem, aparência ou diagramas de uma camada aparecerão automaticamente na tabela de atributos. A exceção são os atributos que podem ser modificados usando a: ref: *Barra de ferramentas do rótulo<label_toolbar>* que estão ocultos por padrão. Os campos auxiliares que representam uma `` Cor`` têm um widget ** Cor ** definido por padrão, caso contrário, os campos auxiliares são padronizados para o widget ** Edição de Texto **.

Os campos auxiliares que representam os atributos que podem ser modificados usando a:ref:*Barra de ferramentas do rótulo<label_toolbar>* são ** Ocultos ** na tabela de atributos por padrão. Para tornar um campo visível, abra a:ref:*guia de propriedades do formulário de atributo<vector_attributes_menu>* e altere o valor de um campo auxiliar: guilabel: *Widget Type* de ** Oculto ** para outro valor relevante. Por exemplo, altere ** auxiliary_storage_labeling_size ** para ** Text Edit ** ou altere ** auxiliary_storage_labeling_color ** para o widget ** Color **. Esses campos agora estarão visíveis na tabela de atributos.

Os campos auxiliares na tabela de atributos aparecerão como a seguinte imagem:

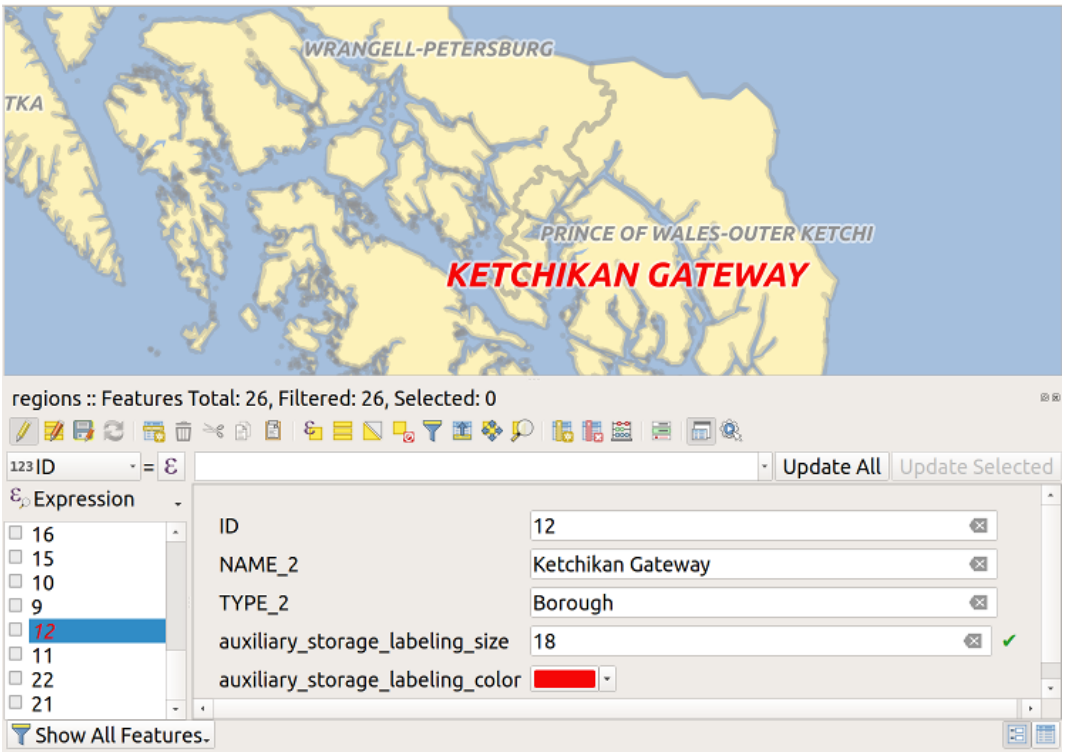


Figura12.56: Formulário com campos auxiliares

Gestão

O menu:guilabel:*Camada Auxiliar* permite que você gerencie os campos auxiliares:

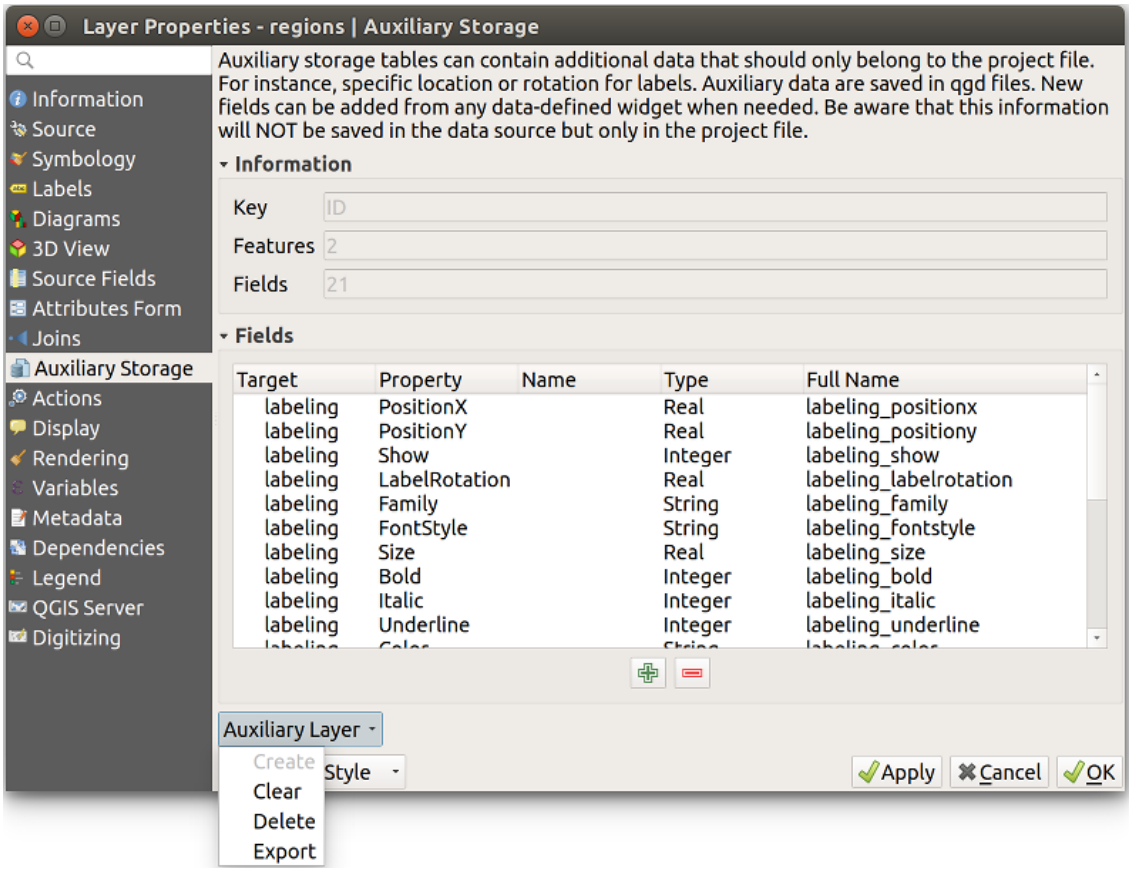


Figura12.57: Gerenciamento de camada auxiliar

O primeiro item:guilabel:*Criar* está desabilitado neste caso porque a camada auxiliar já foi criada. Mas no caso de um novo trabalho, você pode usar esta ação para criar uma camada auxiliar. Conforme explicado em:ref:*vector_auxiliary_storage_key*, uma chave primária será necessária então.

A ação:guilabel:*Clear* permite manter todos os campos auxiliares, mas remover seus conteúdos. Dessa forma, o número de recursos que usam esses campos cairá para `` 0``.

A ação:guilabel:*Delete* remove completamente a camada auxiliar. Em outras palavras, a tabela correspondente é excluída do banco de dados SQLite subjacente e a personalização das propriedades é perdida.

Finalmente, a ação:guilabel:*Export* permite salvar a camada auxiliar como uma: ref:` nova camada vetorial <general_saveas>`. Observe que as geometrias não são armazenadas no armazenamento auxiliar. No entanto, neste caso, as geometrias também são exportadas da fonte de dados original.

Banco de dados de armazenamento auxiliar

Quando você salva seu projeto com o formato `` .qgs``, o banco de dados SQLite usado para armazenamento auxiliar é salvo no mesmo lugar, mas com a extensão `` .qgd``.

Por conveniência, um arquivo pode ser usado graças ao formato `` .qgz``. Neste caso, os arquivos `` .qgd`` e `` .qgs`` são ambos embutidos no arquivo.

12.1.12 Propriedades de Ações

The *Actions* tab provides the ability to perform an action based on the attributes of a feature. This can be used to perform any number of actions, for example, running a program with arguments built from the attributes of a feature or passing parameters to a web reporting tool.

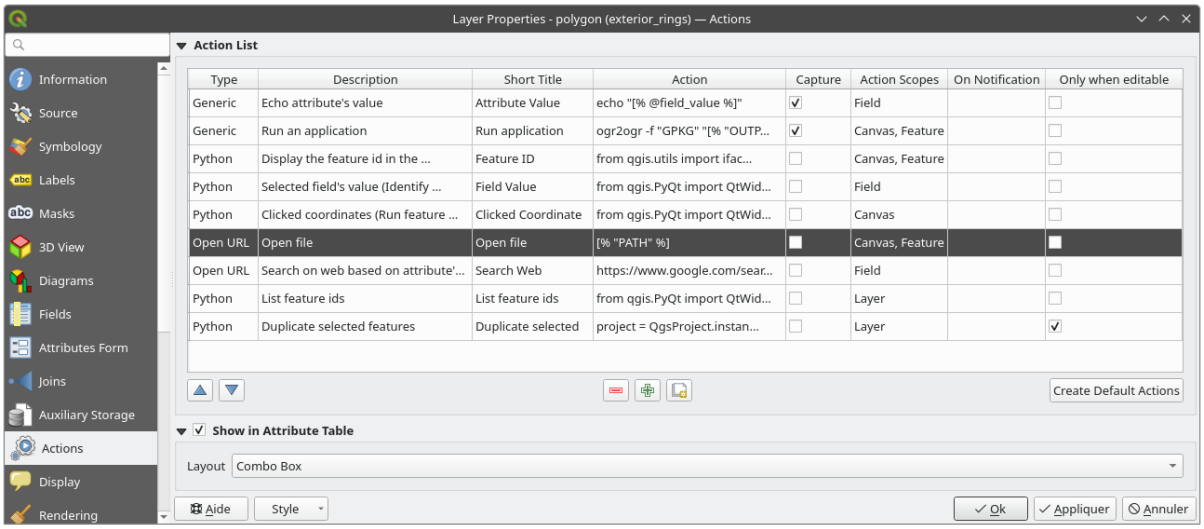


Figura12.58: Diálogo de ação de visão geral com alguns exemplos de ações

Actions are useful when you frequently want to run an external application or view a web page based on one or more values in your vector layer. They are of different types and can be used like this:

- *Generic*, *macOS*, *Windows* and *Unix* actions start an external process.
- *Python* actions execute a Python expression.
- *Generic* and *Python* actions are visible everywhere.
- *macOS*, *Windows* and *Unix* actions are visible only on the respective platform (i.e., you can define three “Edit” actions to open an editor and the users can only see and execute the one “Edit” action for their platform to run the editor).
- *Open URL*: Uses a HTTP GET request to open a provided URL.
- *Submit URL (urlencoded or JSON)*: Same as the *Open URL* action but using a HTTP POST request. Data are posted to a URL, using “application/x-www-form-urlencoded” or “application/json” if the body is a valid JSON.

  An example of action call could be:

  ```
  http://localhost:8000?/[% url_encode(map('file', 'index.php')) %]
  ```

- *Submit URL (multipart)*: Same as the *Open URL* action but using a HTTP POST request. Data are posted to a URL, using “multipart/form-data”.

Existem vários exemplos incluídos na caixa de diálogo. Você pode carregá-los clicando em:guilabel:*Criar ações padrão*. Para editar qualquer um dos exemplos, clique duas vezes em sua linha. Um exemplo é realizar uma pesquisa com base em um valor de atributo. Este conceito é usado na discussão a seguir.

A | caixa de seleção | *Mostrar na Tabela de Atributos* permite que você exiba na caixa de diálogo da tabela de atributos as ações com escopo de recursos verificados, seja como: guilabel:` Combo Box` ou como: guilabel: *Botões Separados* (ver: ref:` configure_table_columns `).

Definindo Ações

To define an attribute action, open the vector *Layer Properties* dialog and click on the *Actions* tab. In the *Actions* tab, click the Add a new action to open the *Edit Action* dialog.

Select the action *Type* and provide a descriptive name for the action. The action itself must contain the name of the application that will be executed when the action is invoked. You can add one or more attribute field values as arguments to the application. When the action is invoked, any set of characters that start with a `%` followed by the name of a field will be replaced by the value of that field. The special characters `%%` will be replaced by the value of the field that was selected from the identify results or attribute table (see *Usando ações*). Double quote marks can be used to group text into a single argument to the program, script or command. Double quotes will be ignored if preceded by a backslash.

Actions can invoke a single process, with arguments, so Boolean operators (such as `&`, `&&`, `;`, `|`) will not work. In UNIX-like operating systems multiple commands can be executed via `bash -c`.

The *Action Scopes* allows you to define where the action should be available. You have following choices:

1. *Field*: action is available when right click in the cell within the attribute table, in the feature form and in the default action button of the main toolbar.
2. *Feature*: action is available when right click in the cell within the attribute table.
3. *Canvas*: action is available in the main action button in the toolbar.
4. *Form*: action is available only in a feature form designed using the *drag-and-drop* mode.
5. *Layer*: action is available in the action button in the attribute table toolbar. Be aware that this type of action involves the entire layer and not the single features.

If you have field names that are substrings of other field names (e.g., `col1` and `col10`), you should indicate that by surrounding the field name (and the % character) with square brackets (e.g., `[%col10]`). This will prevent the `%col10` field name from being mistaken for the `%col1` field name with a `0` on the end. The brackets will be removed by QGIS when it substitutes in the value of the field. If you want the substituted field to be surrounded by square brackets, use a second set like this: `[[%col10]]`.

Using the *Identify Features* tool, you can open the *Identify Results* dialog. It includes a *(Derived)* item that contains information relevant to the layer type. The values in this item can be accessed in a similar way to the other fields by proceeding the derived field name with `(Derived).`. For example, a point layer has an `X` and `Y` field, and the values of these fields can be used in the action with `%(Derived).X` and `%(Derived).Y`. The derived attributes are only available from the *Identify Results* dialog box, not the *Attribute Table* dialog box.

Two example actions are shown below:

- `konqueror https://www.google.com/search?q=%nam`
- `konqueror https://www.google.com/search?q=%%`

In the first example, the web browser konqueror is invoked and passed a URL to open. The URL performs a Google search on the value of the `nam` field from our vector layer. Note that the application or script called by the action must be in the path, or you must provide the full path. To be certain, we could rewrite the first example as: `/opt/kde3/bin/konqueror https://www.google.com/search?q=%nam`. This will ensure that the konqueror application will be executed when the action is invoked.

O segundo exemplo usa a notação % % , que não depende de um campo específico para o seu valor. Quando a ação é consultado, o % % será substituído pelo valor do campo selecionado na tabela de identificar resultados ou atributo.

Duplicating Actions

QGIS allows you to duplicate existing actions. To duplicate an attribute action, open the vector *Layer Properties* dialog and click on the *Actions* tab. In the *Actions* tab, click the Duplicate an action to open the *Duplicate Action* dialog. You must have selected at least one existing action in order to create a duplicate.

In the dialogue that appears, make any changes that are necessary. See *Definindo Ações* for further information. Once finished, press *OK* to create a duplicate of the action with any changes that you made. If you did not edit the description, or if you changed it to be identical to the description of any other existing action, "_1" will be added to the end of it.

Usando ações

O QGIS oferece várias maneiras de executar ações que você ativou em uma camada. Dependendo das configurações, eles podem estar disponíveis:

- no menu suspenso do | ação Executar| :sup:botão *Executar ação de recurso* na: guilabel:` Barra de ferramentas Atributos` ou:guilabel:*Tabela de Atributos*;
- when right-clicking a feature with the Identify Features tool (see *Identificando Feições* for more information);
- from the *Identify Results* panel, under the *Actions* section;
- as items of an *Actions* column in the *Attribute Table* dialog.

Se você está chamando uma ação que usa o `% ``% notação, clique com o botão direito sobre o valor do campo na:guilabel:` Identificar resultados de diálogo ou o:guilabel:"Tabela de atributos` diálogo que você deseja passar para a aplicação ou script.

Here is another example that pulls data out of a vector layer and inserts it into a file using bash and the `echo` command (so it will only work on or perhaps). The layer in question has fields for a species name `taxon_name`, latitude `lat` and longitude `long`. We would like to be able to make a spatial selection of localities and export these field values to a text file for the selected record (shown in yellow in the QGIS map area). Here is the action to achieve this:

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

Depois de selecionar algumas localidades e executar a ação em cada um, a abertura do arquivo de saída irái mostrar algo como isto:

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

As an exercise, we can create an action that does a Google search on the `lakes` layer. First, we need to determine the URL required to perform a search on a keyword. This is easily done by just going to Google and doing a simple search, then grabbing the URL from the address bar in your browser. From this little effort, we see that the format is https://www.google.com/search?q=QGIS, where `QGIS` is the search term. Armed with this information, we can proceed:

1. Verifique se a camada `lagos` está carregada.
2. Open the *Layer Properties* dialog by double-clicking on the layer in the legend, or right-click and choose *Properties* from the pop-up menu.
3. Click on the *Actions* tab.
4. Click Add a new action.
5. Choose the *Open URL* action type,
6. Digite um nome para a ação, por exemplo `` Google Search ``.

7. Additionally you can add a *Short Name* or even an *Icon*.
8. Choose the *Action Scopes*. See *Definindo Ações* for further information. Leave the default settings for this example.
9. For the action, add the URL used for doing a Google search, up to but not including the search term: `https://www.google.com/search?q=`
10. The text in the *Action* field should now look like this:

```
https://www.google.com/search?q=
```

11. Clique na caixa suspensa que contém os nomes dos campos da camada `` lagos``. Ele está localizado à esquerda do botão:guilabel:*Insert*.
12. From the drop-down box, select *NAMES* and click *Insert*.
13. Your action text now looks like this:

```
https://www.google.com/search?q=[%NAMES%]
```

14. Para finalizar e adicionar a ação, clique no botão:guilabel:*OK*.

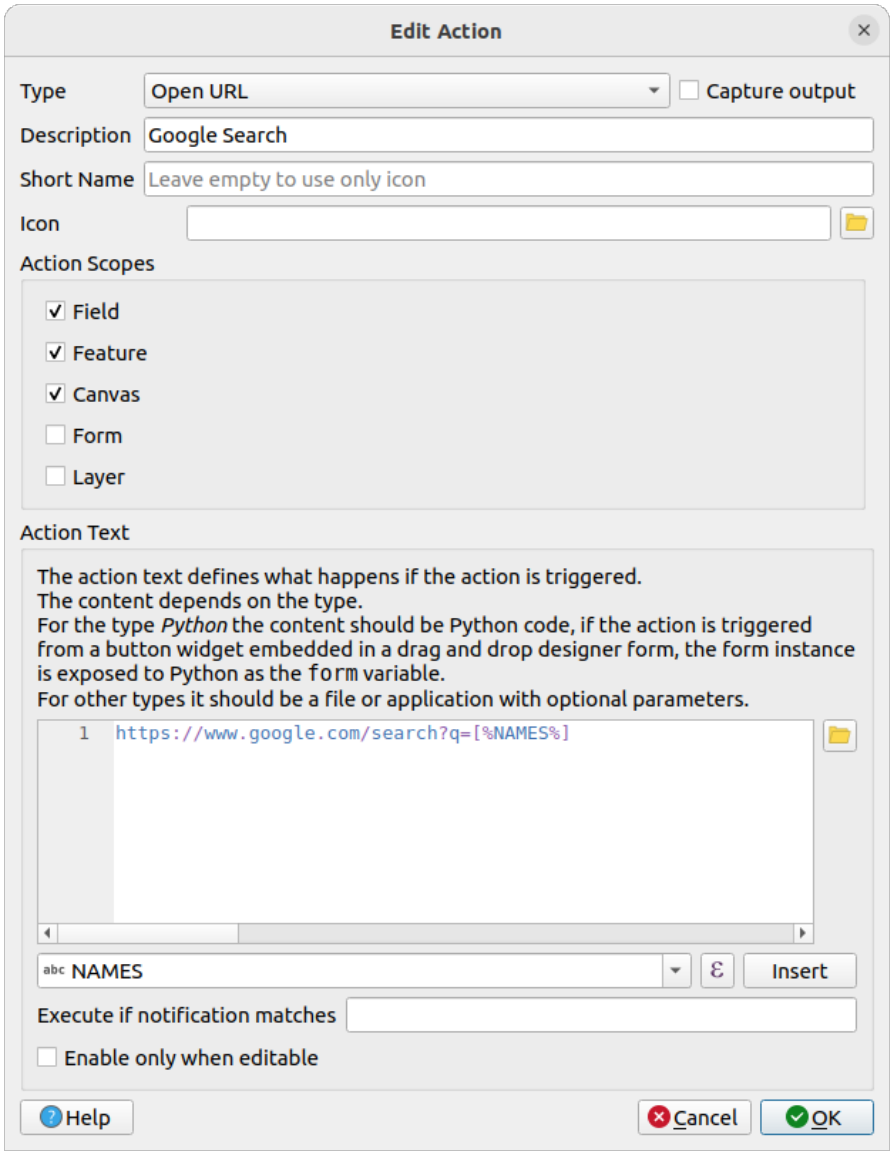


Figura12.59: Caixa de diálogo Editar Ação Configurada com o Exemplo

This completes the action, and it is ready to use.

Close the *Layer Properties* dialog and zoom in to an area of interest. Make sure the `lakes` layer is active and identify a lake. In the result box you'll now see that our action is visible:

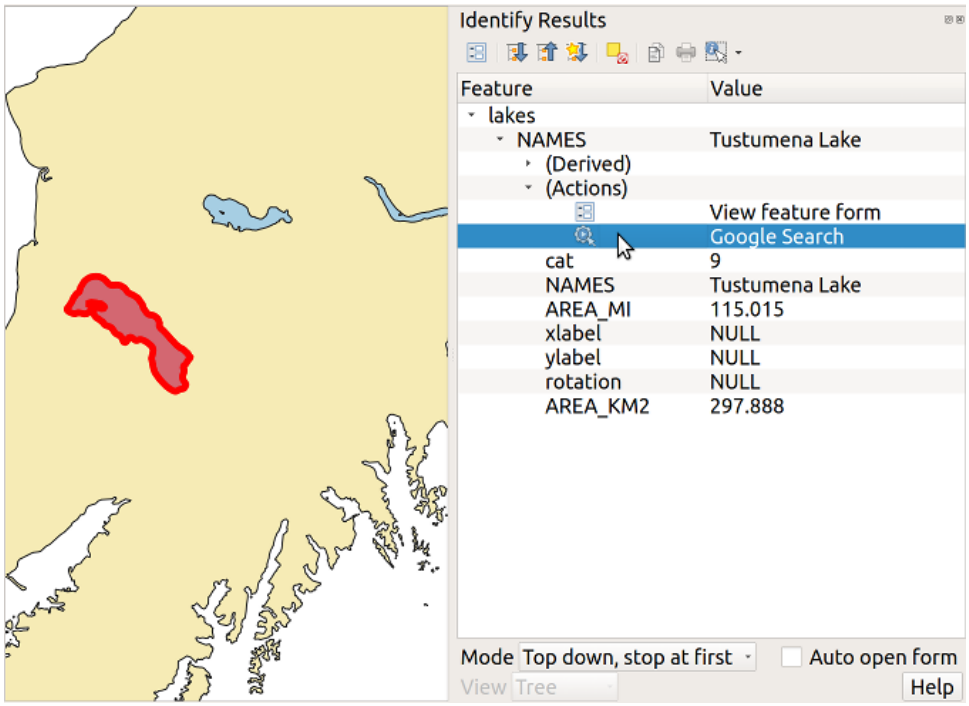


Figura12.60: Selecione o recurso e escolha a ação

When we click on the action, it brings up our default browser and navigates to the URL https://www.google.com/search?q=Tustumena. It is also possible to add further attribute fields to the action. Therefore, you can add a + to the end of the action text, select another field and click on *Insert Field*. In this example, there is just no other field available that would make sense to search for.

Você pode definir várias ações para uma camada, e cada um vai aparecer no diálogo *Identificar resultados*.

You can also invoke actions from the attribute table by selecting a row and right-clicking, then choosing the action from the pop-up menu.

Há todos os tipos de usos para as ações. Por exemplo, se você tem uma camada de pontos que contém as localizações de imagens ou fotos, juntamente com um nome de arquivo, você pode criar uma ação para lançar um visualizador para exibir a imagem. Você também pode usar ações para lançar relatórios baseados na web para um campo de atributo ou combinação de campos, especificando-as da mesma forma que fizemos no nosso exemplo de busca Google.

Nós também podemos fazer exemplos mais complexos, por exemplo, usando ações **Python**.

Usually, when we create an action to open a file with an external application, we can use absolute paths, or eventually relative paths. In the second case, the path is relative to the location of the external program executable file. But what about if we need to use relative paths, relative to the selected layer (a file-based one, like Shapefile or SpatiaLite)? The following code will do the trick:

```
command = "firefox"
imagerelpath = "images_test/test_image.jpg"
layer = qgis.utils.iface.activeLayer()
import os.path
layerpath = layer.source() if layer.providerType() == 'ogr'
  else (qgis.core.QgsDataSourceURI(layer.source()).database()
    if layer.providerType() == 'spatialite' else None)
path = os.path.dirname(str(layerpath))
image = os.path.join(path,imagerelpath)
import subprocess
subprocess.Popen( [command, image ] )
```

Nós apenas temos que lembrar que a ação é um do tipo variáveis *Python* e o *comando* e *imagerelpath* devem ser alteradas para atender as necessidades.

Mas o que dizer se o caminho relativo deve ser relativo ao arquivo de projeto (salvo)? O código da ação Python seria:

```
command = "firefox"
imagerelpath = "images_test/test_image.jpg"
projectpath = qgis.core.QgsProject.instance().fileName()
import os.path
path = os.path.dirname(str(projectpath)) if projectpath != '' else None
image = os.path.join(path, imagerelpath)
import subprocess
subprocess.Popen( [command, image ] )
```

Outro exemplo de ação Python é a que nos permite adicionar novas camadas ao projeto. Por exemplo, os exemplos a seguir irão adicionar ao projeto, respectivamente, um vetor e raster. Os nomes dos arquivos a serem adicionados ao projeto e os nomes a serem indicados para as camadas são conduzidos de dados (*nome do arquivo* e *nome da camada* são nomes da tabela de atributos do vetor, onde a ação foi criada coluna):

```
qgis.utils.iface.addVectorLayer('/yourpath/[% "filename" %].shp',
  '[% "layername" %]', 'ogr')
```

Para adicionar um raster (uma imagem TIF neste exemplo), ele se torna:

```
qgis.utils.iface.addRasterLayer('/yourpath/[% "filename" %].tif',
  '[% "layername" %]')
```

12.1.13 Display Properties

The *Display* tab helps you configure fields to use for feature identification:

- The *Display name*: based on a field or an *expression*. By default, this is set to the first field in the layer if no field with `<name>` component exists. This is used as:
  - the label shown on top of the feature information in the *Identify tool* results
  - the field used in the *locator bar* when looking for features in all layers
  - the feature identifier in the attribute table *form view*
  - the feature identifier when the map or layout is exported to a layered output format such as Geospatial PDF
  - the map tip information, i.e. the message displayed in the map canvas when hovering over a feature of the active layer with the Show Map Tips icon pressed. Applicable when *Enable Map Tips* is active and no *HTML Map Tip* is set.
- *Enable Map Tips* controls whether to display map tips for the layer
- The *HTML Map Tip* provides a complex and full HTML text editor for map tips, mixing QGIS expressions and html styles and tags (multiline, fonts, images, hyperlink, tables, …). You can check the result of your code sample in the *Preview* frame (also convenient for previewing the *Display name* output). Additionally, you can select and edit existing expressions using the *Insert/Edit Expression* button.

> **Nota**
>
> Understanding the *Insert/Edit Expression* button behavior
>
> If you select some text within an expression (between "[%" and "%]"), or if no text is selected but the cursor is inside an expression, the whole expression will be automatically selected for editing. If the cursor or a selected text is outside an expression, the dialog opens with the selection.

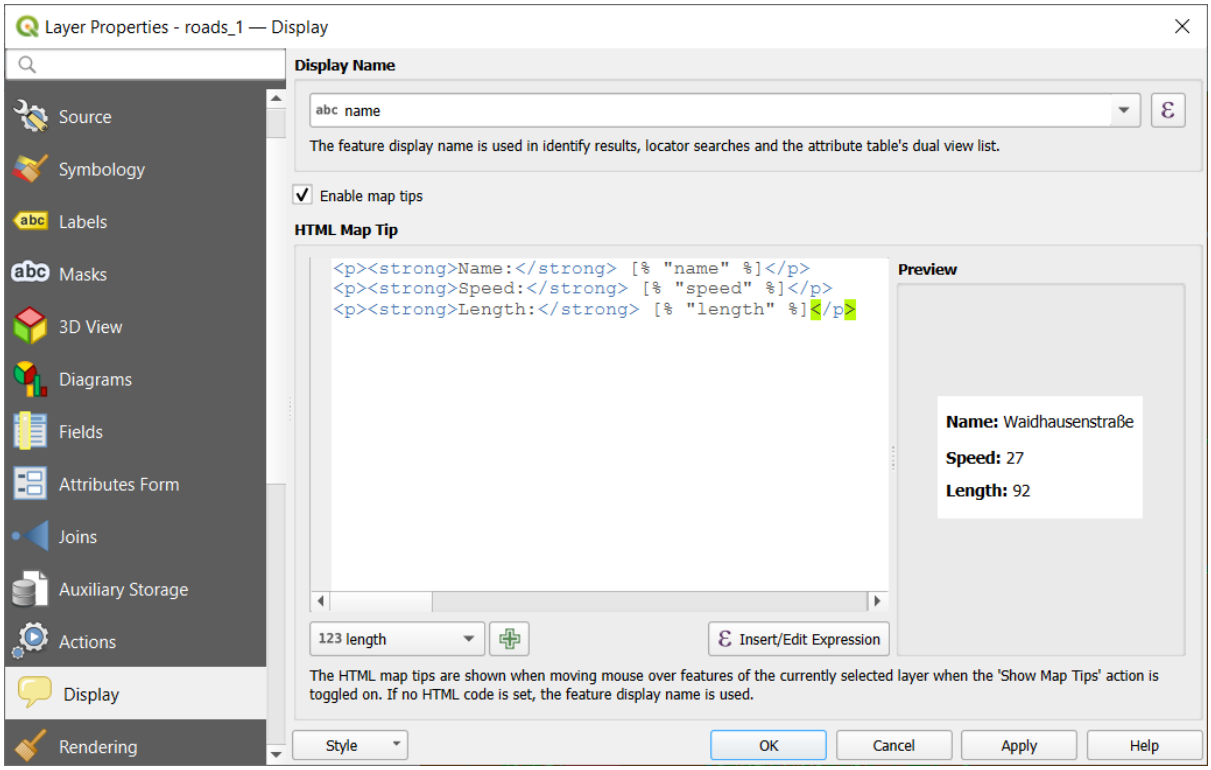


Figura12.61: HTML code for map tip

To display map tips:

1. Select the menu option *View* ► *Show Map Tips* or click on the Show Map Tips icon of the *Attributes Toolbar*.
2. Make sure that the layer you target is active and has the *Enable Map Tips* property checked.
3. Move over a feature, and the corresponding information will be displayed over.

Map tip is a cross-layer feature meaning that once activated, it stays on and applies to any map tip enabled layer in the project until it's toggled off.

Figura12.62: Map tip made with HTML code

12.1.14 Propriedades de Renderização

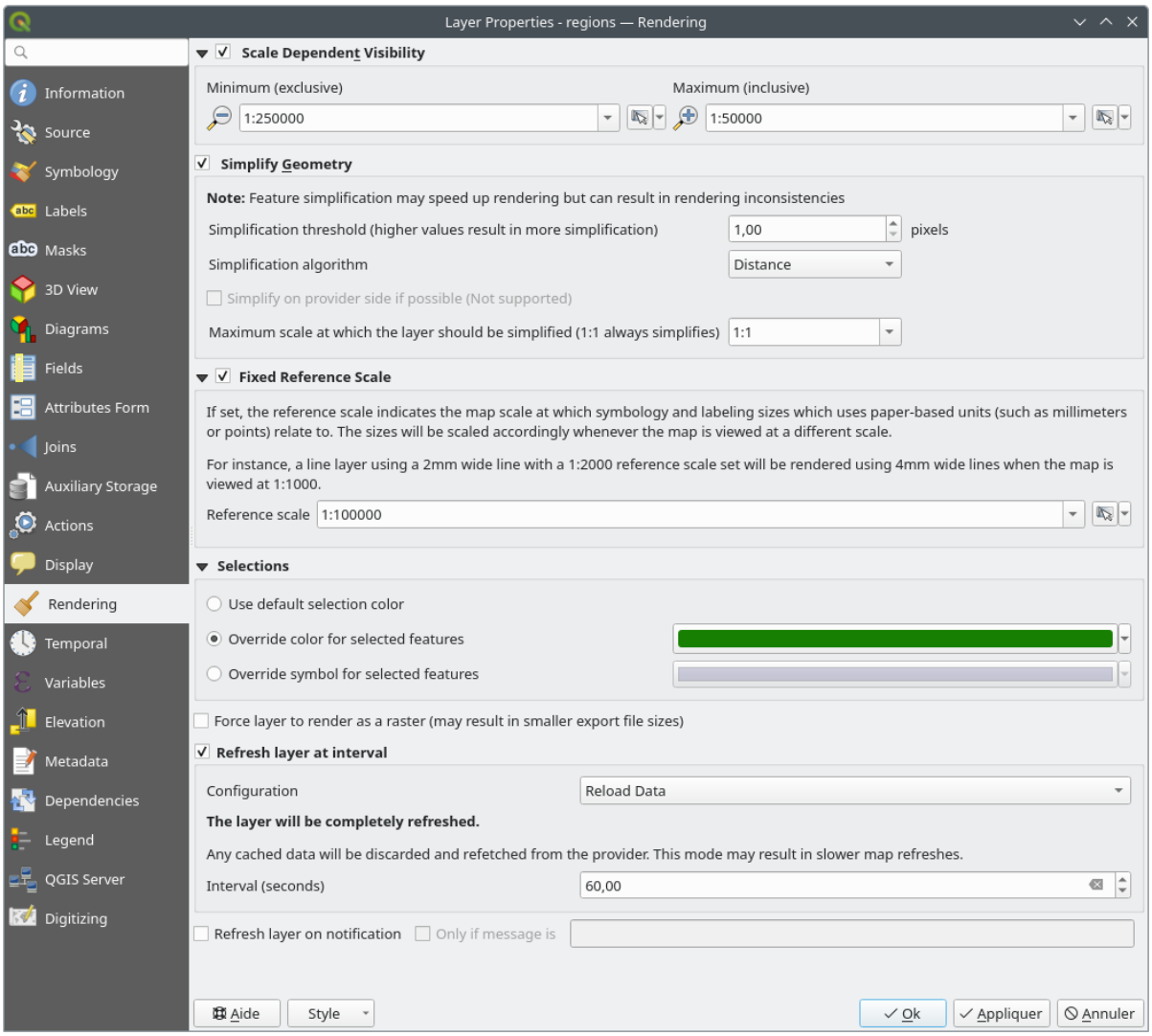


Figura12.63: Layer Rendering Properties dialog

The *Rendering* tab offers following properties:

- Under *Scale dependent visibility*, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Visibility Scale Selector* for more information.

  > **Nota**
  >
  > You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

- QGIS offers support for on-the-fly feature generalisation. This can improve rendering times when drawing many complex features at small scales. This feature can be enabled or disabled in the layer settings using the *Simplify geometry* option. There is also a global setting that enables generalisation by default for newly added layers (see *global simplification* for more information).

> **Nota**
>
> Feature generalisation may introduce artefacts into your rendered output in some cases. These may include slivers between polygons and inaccurate rendering when using offset-based symbol layers.

- The *Fixed reference scale* indicates the map scale at which symbology and labeling sizes which uses paper-based units (such as millimeters or points) relate to. The sizes will be scaled accordingly whenever the map is viewed at a different scale.

  For instance, a line layer using a 2mm wide line with a 1:2000 *Reference scale* set will be rendered using 4mm wide lines when the map is viewed at 1:1000.

- The *Selections* group allows you to control whether a specific color or symbol should be used in place of the defaults (*Project properties* ► *General* ► *Selection color*) for a particular layer. This is useful to improve the visibility of selected features with certain symbology:

  - *Use default selection color*
  - *Override color for selected features*: e.g. if the layer uses a yellow color by default and the standard yellow selection is not visible.
  - *Override symbol for selected features*: e.g. if a line layer uses a thin symbol, and coloring the lines might not make them visible enough, overriding the symbol with a thicker line can help. Also, if the layer uses raster symbology or gradient fills/lines/shapeburst with color ramp symbology, then the default selection color is not applied at all; being able to set a specific simpler symbol to use for selected features in the layer can help.

- Rendering extremely detailed layers (e.g. polygon layers with a huge number of nodes), can cause layout exports in PDF/SVG format to be huge as all nodes are included in the exported file. This can also make the resultant file very slow to work with/open in other programs.

  Checking *Force layer to render as raster* forces these layers to be rasterised so that the exported files won't have to include all the nodes contained in these layers and the rendering is therefore sped up.

  You can also do this by forcing the layout to export as a raster, but that is an all-or-nothing solution, given that the rasterisation is applied to all layers. Alternatively, you can rely on geometry simplification in *layout export settings*.

- *Refresh layer at interval*: controls whether and how regular a layer can be refreshed. Available *Configuration* options are:

  - *Reload data*: the layer will be completely refreshed. Any cached data will be discarded and refetched from the provider. This mode may result in slower map refreshes.
  - *Redraw layer only*: this mode is useful for animation or when the layer's style will be updated at regular intervals. Canvas updates are deferred in order to avoid refreshing multiple times if more than one layer has an auto update interval set.

  It is also possible to set the *Interval (seconds)* between consecutive refreshments.

- Depending on the data provider (e.g. PostgreSQL), notifications can be sent to QGIS when changes are applied to the data source, out of QGIS. Use the *Refresh layer on notification* option to trigger an update. You can also limit the layer refresh to a specific message set in the *Only if message is* text box.

12.1.15 Propriedades Temporais

The *Temporal* tab provides options to control the rendering of the layer over time. Such dynamic rendering requires the *temporal navigation* to be enabled over the map canvas.

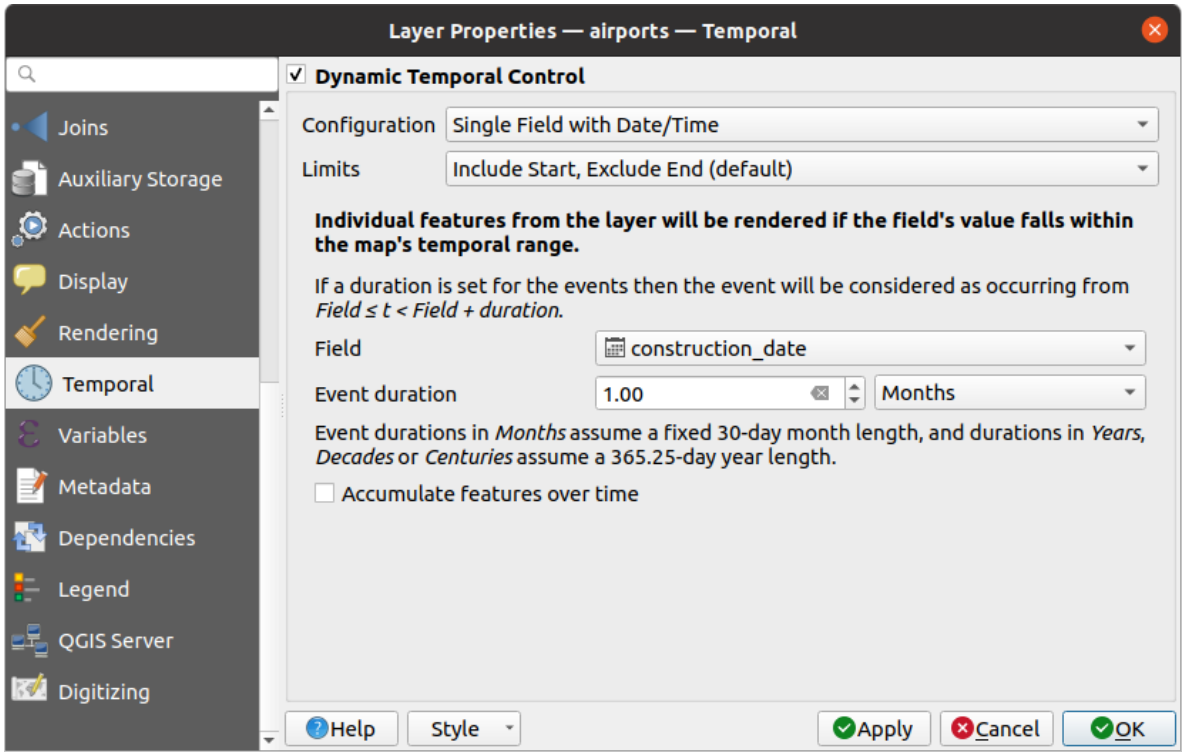


Figura12.64: Vector layer temporal properties dialog

Check the *Dynamic Temporal Control* option to configure the vector layer temporal rendering. Depending on the structure of your dataset, you may want to use one of the provided *Configuration* options:

- *Fixed time range*: all the features are rendered if the map canvas temporal frame overlaps the given *Start date* and *End date* range.
- *Single field with date/time*: features are rendered if their *Field*'s value falls within the map canvas temporal frame. An *Event duration* can be set. With checking the *Accumulate features over time* option, all features which occur before or within the map's temporal range will continue to be rendered. The event duration is thus ignored.
- *Separate fields for start and end date/time*: features are rendered if the range specified by their *Start field* and *End field* values overlaps the map canvas temporal.
- *Separate fields for start and event duration*: features are rendered if the range defined by their *Start field* and *Event duration field* values overlaps the map canvas temporal.
- *Data/hora de início e fim das expressões*: as características são renderizadas se o intervalo de tempo especificado pelos campos *Expressão de início* e *Expressão de fim* se sobrepõem ao tempo da tela do mapa.
- *Redraw layer only*: the layer is redrawn at each new animation frame but no time-based filtering is applied to the features. It's useful when the layer uses time-based expression values for renderer settings (e.g. data-defined symbology).

It is also possible to set the *Limits* of the features time range as:

- *Inclua início, exclua fim*
- :guilabel:`Inclua o início, inclua o fim'.

12.1.16 Propriedades das Variáveis

The *Variables* tab lists all the variables available at the layer's level (which includes all global and project's variables).

It also allows the user to manage layer-level variables. Click the button to add a new custom layer-level variable. Likewise, select a custom layer-level variable from the list and click the button to remove it.

More information on variables usage in the General Tools *Armazenando valores em Variáveis* section.

12.1.17 Elevation Properties

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and its appearance in the *profile tool charts*. Specifically, you can set:

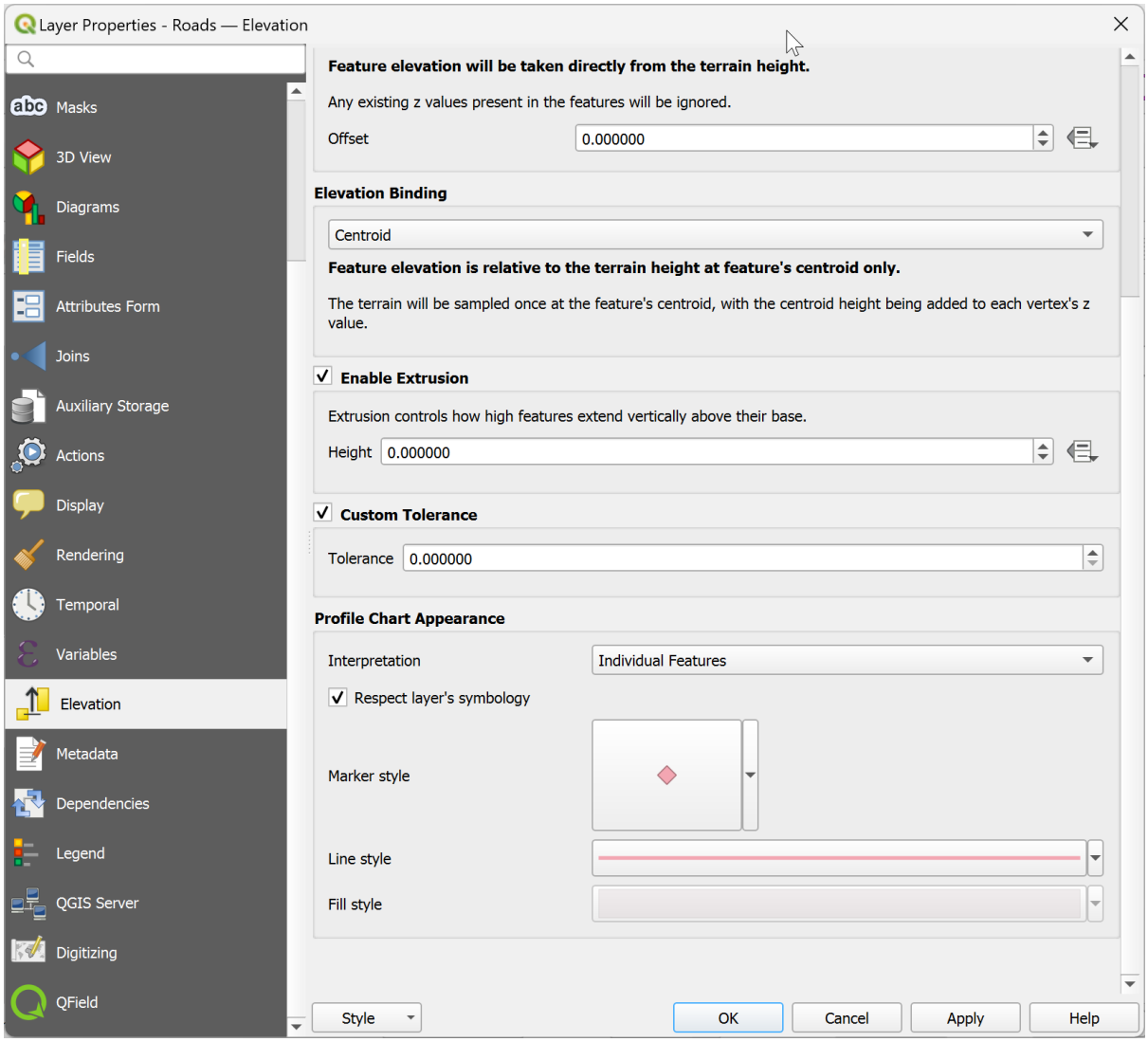


Figura12.65: Vector layer elevation properties dialog

- *Vertical Reference System*: If the CRS of your vector layer is a compound one (including a Z dimension), then the vertical CRS used for the layer will be the vertical component of the layer CRS. In this case, you cannot

manually set a different vertical CRS. If your layer CRS is horizontal (2D), then you can select a specific vertical CRS by clicking on the Select CRS. Vertical reference systems are supported for vector layers by:

  - *Elevation profiles*
  - *Identify Tool Results*
  - *3D map views*

- *Elevation Clamping*: defines how and whether the features altitude should be:
  - *Clamped to terrain*: takes elevation directly from the terrain height and ignores any existing Z values in the features. A data-defined *Offset* value from the terrain can also be filled.
  - *Relative to terrain*: any existing Z values in the features are added to the terrain height. A *Scale* factor followed by a data-defined *Offset* can be used to adjust the elevation. This option is not available for 2D geometry layers.
  - *Absolute*: ignores the terrain height and directly takes Z values from the features for the elevation. A *Scale* factor followed by a data-defined *Offset* can be used to adjust the elevation. For 2D geometry layers (with no Z values), a data-defined *Base height* can instead be set.
- *Enable extrusion*: you can set a *Height* to control how high features vertically extend above their base. This is convenient to indicate that a 2D geometry layers, e.g. a polygon building footprints layer, actually represents 3D objects.
- *Custom tolerance*: allows you to override the global sampling tolerance defined in the elevation profile settings, specifically for this vector layer. When enabled, you can set *Tolerance* that determines how the layer is sampled along the profile line. This is useful when you need finer or coarser sampling than the global default, for example, to improve performance or accuracy for complex geometries.
- *Elevation Binding*: only relevant when combining an *Elevation clamping* relying on the terrain with a line or polygon layer, this option controls how feature elevation is set relative to the terrain height. The terrain can be sampled:
  - at the feature's *Centroid*, with the centroid height being added to each vertex's z value
  - at every individual *Vertex* before being added to the vertex's z value
- *Profile Chart Appearance*: controls how features are rendered when drawing a profile chart. Two main *Interpretation* modes are available:
  - as *Individual features*: samples discrete positions where the cross section profile line intersects the vector features. That intersection can be represented as point, line or surface depending on the layer type and whether an extrusion is applied.

    With checking *Respect layer symbology*, features will be rendered on the profile chart with their corresponding *layer styling* (allowing e.g. categorized classes to be visible on the profile chart). If the profile symbol type does not match the layer's renderer symbol types, only the symbol color from the renderer is applied to the profile symbol.

    Depending on the layer settings, profile symbols can be represented with a custom style, using:

    * *Marker style*: for non-extruded point and line features, and for non-extruded polygon features touched by the profile line
    * *Line style*: for extruded point and line features, and for non-extruded polygon features intersected by the profile line
    * *Fill style*: for extruded polygon features
  - as *Continuous Surface (e.g. contours)*: the elevation chart will be rendered as a surface instead of separate features by joining the sampled elevation results into a continuous line. This can enhance visualisation and is designed for vector layers which represent a continuous elevation surface, e.g. contour lines or surveyed elevation points. The profile *Style* can be set as:
    * a *Line* with a specific *Line style*

* an elevation surface rendered using a fill symbol either above (*Fill above*) or below (*Fill below*) the elevation curve line. The surface symbology is represented using:
  - a *Fill style*
  - and a *Limit*: the maximum (respectively minimum) altitude determining how high the fill surface will be

Moreover, you can check *Show markers at sampled points* to make them visible over the interpretation line and assign them a *Marker style*.

12.1.18 Propriedades dos Metadados

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metadados* for more information.

12.1.19 Propriedades de Dependências

The *Dependencies* tab allows to declare data dependencies between layers. A data dependency occurs when a data modification in a layer, not by direct user manipulation, may modify data of other layers. This is the case for instance when geometry of a layer is updated by a database trigger or custom PyQGIS scripting after modification of another layer's geometry.

In the *Dependencies* tab, you can select any layers which may externally alter the data in the current layer. Correctly specifying dependent layers allows QGIS to invalidate caches for this layer when the dependent layers are altered.

12.1.20 Propriedades da legenda

The *Legend* properties tab provides you with advanced settings for the *Layers panel* and/or the *print layout legend*. These options include:

- Depending on the symbology applied to the layer, you may end up with several entries in the legend, not necessarily readable/useful to display. The *Legend placeholder image* helps you *select an image* for replacement, displayed both in the *Layers* panel and the print layout legend.
- *Show label legend*: Displays overviews of the different label settings as entries in the legends. The *label style* is previewed along with the description.
- *Text on symbols*: In some cases it can be useful to add extra information to the symbols in the legend. With this frame, you can affect to any of the symbols used in the layer symbology a text that is displayed over the symbol, in both *Layers* panel and print layout legend. This mapping is done by typing each text next to the symbol in the table widget or filling the table using the *Set Labels from Expression* button. Text appearance is handled through the font and color selector widgets of the *Text Format* button.

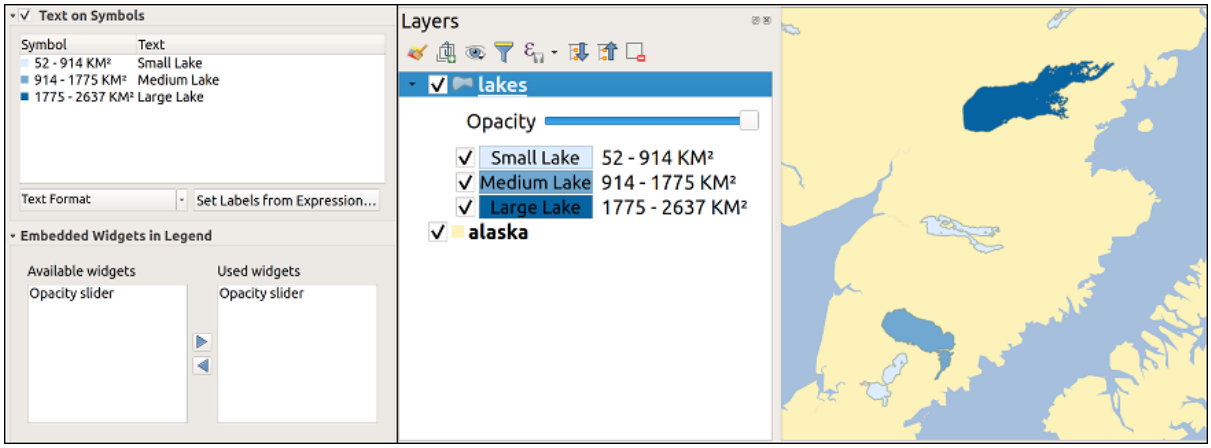


Figura12.66: Setting text on symbols (left) and its rendering in the *Layers* panel (right)

- a list of widgets you can embed within the layer tree in the Layers panel. The idea is to have a way to quickly access some actions that are often used with the layer (setup transparency, filtering, selection, style or other stuff…).

  By default, QGIS provides transparency widget but this can be extended by plugins registering their own widgets and assign custom actions to layers they manage.

12.1.21 Propriedades do Servidor QGIS

The *QGIS Server* tab consists of *Description*, *Attribution*, *Metadata URL*, and *Legend URL* sections.

From the *Description* section, you can change the *Short name* used to reference the layer in requests (to learn more about short names, read services_basics_short_name). You can also add or edit a *Title*, an alternative *WFS Title* and *Abstract* for the layer, or define a *Keyword list* here. These keyword lists can be used in a metadata catalog. If you want to use a title from an XML metadata file, you have to fill in a link in the *Data URL* field.

Use *Attribution* to get attribute data from an XML metadata catalog.

In *Metadata URL*, you can add the general paths to the XML metadata catalog. This information will be saved in the QGIS project file for subsequent sessions and will be used for QGIS Server.

In the *Legend URL* section, you can provide the url of a legend image in the url field. You can use the Format drop-down option to apply the appropriate format of the image. Currently png, jpg and jpeg image formats are supported.

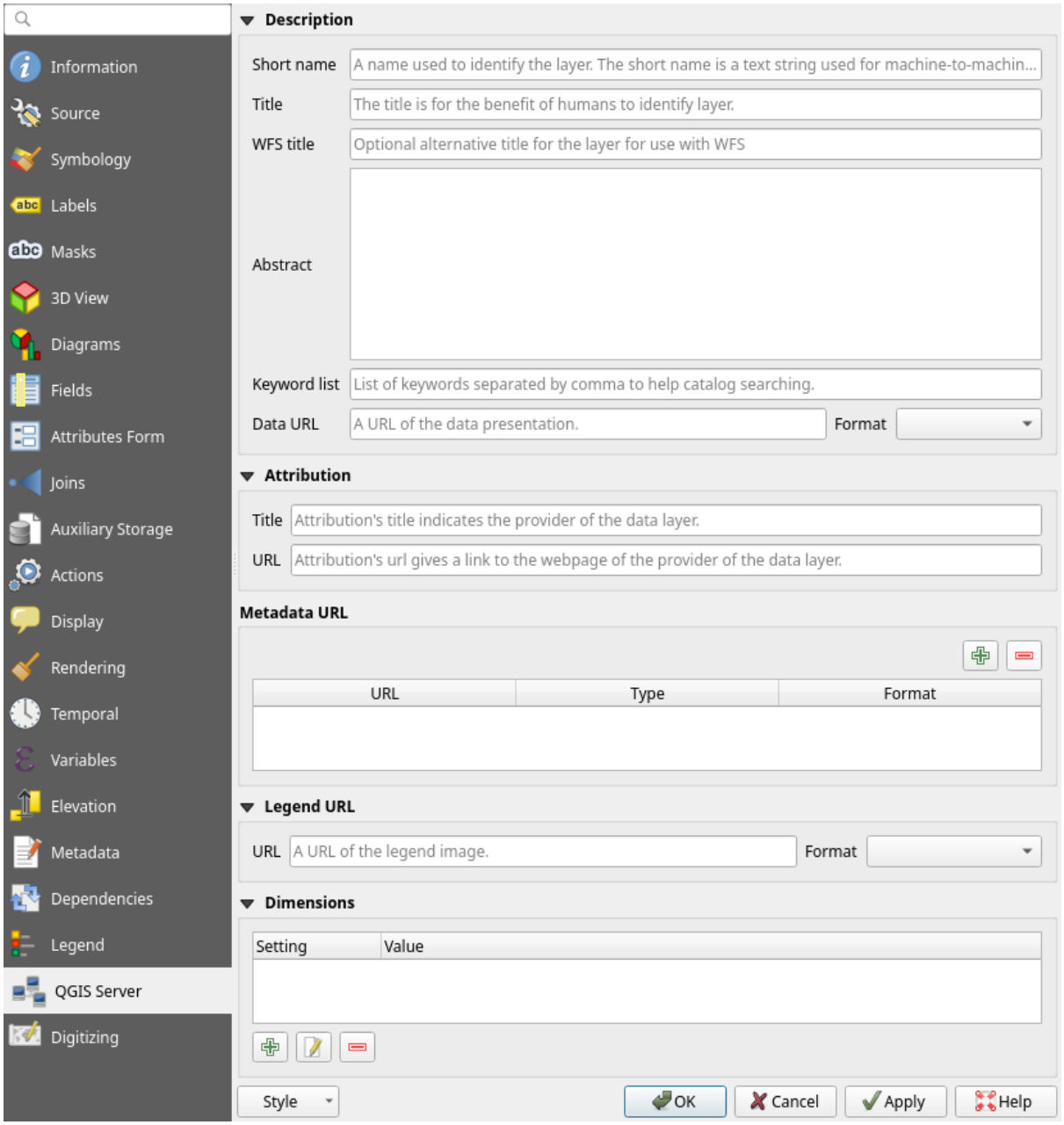


Figura12.67: QGIS Server tab in vector layers properties dialog

Para saber mais sobre o Servidor QGIS, leia o:ref:*QGIS-Server-manual*.

12.1.22 Propriedades de Digitalização

The *Digitizing* tab gives access to options that help to ensure the quality of digitized geometries.

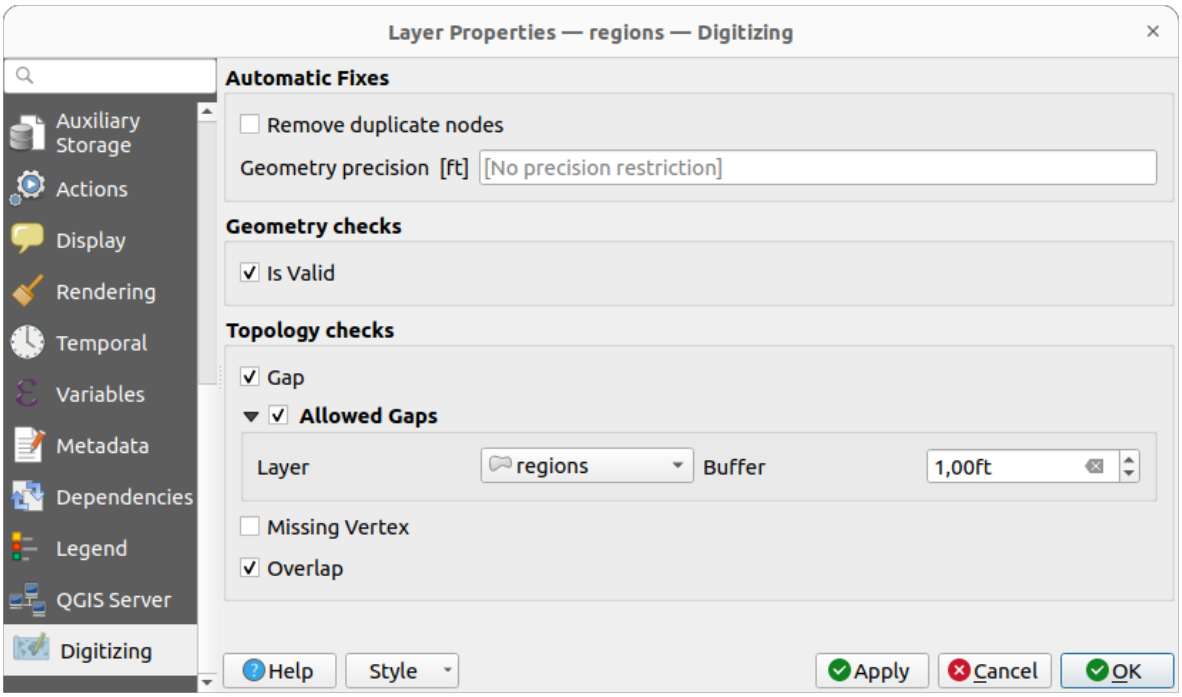


Figura12.68: The QGIS Digitizing tab in the vector layers properties dialog

Correções Automáticas

Options in the *Automatic Fixes* section will directly affect the vertices of any geometry which is added or modified. If the *Remove duplicate nodes* option is checked, any two subsequent vertices with exactly the same coordinates will be removed. If the *Geometry precision* is set, all vertices will be rounded to the closest multiple of the configured geometry precision. The rounding will happen in the layer coordinate reference system. Z and M values are not rounded. With many map tools, a grid is shown on the canvas while digitizing.

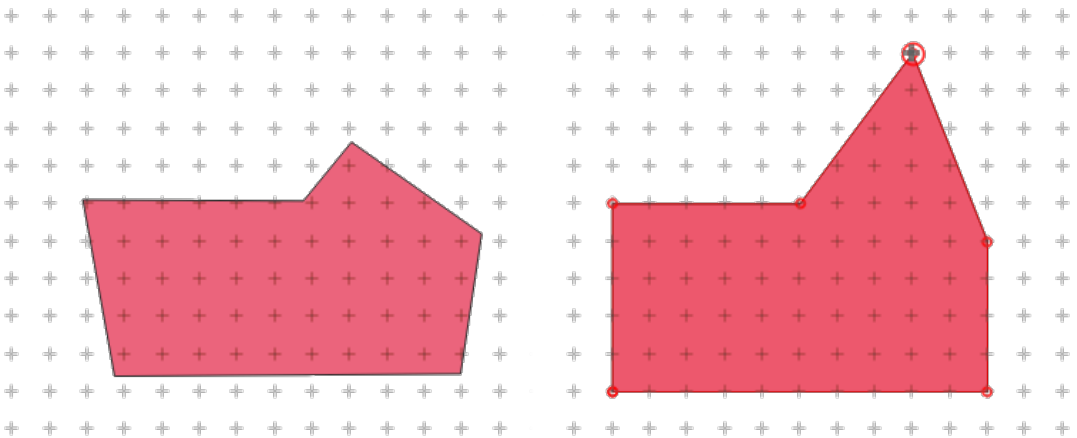


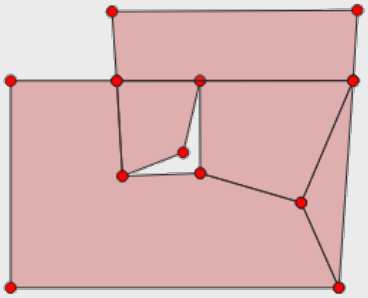
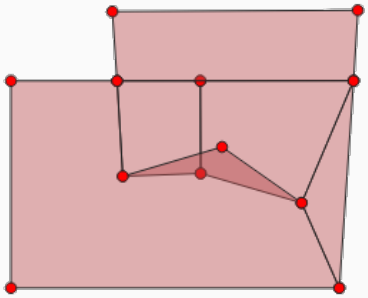
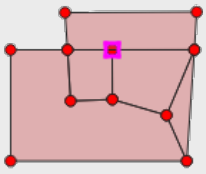
Figura12.69: Movendo o vértice superior encaixa todos os vértices na grade

Verificações de Geometria

In the *Geometry checks* section, additional validations on a per geometry basis can be activated. Immediately after any geometry modification, failures in these checks are reported to the user in the *Geometry validation* panel. As long as a check is failing, it is not possible to save the layer. The *Is valid* check will run basic validity checks like self intersection on geometries.

Verificações de Topologia

In the *Topology checks* section, additional topology validation checks can be activated. Topology checks will be executed when the user saves the layer. Check errors will be reported in the *Geometry validation* panel. As long as validation errors are present, the layer can not be saved. Topology checks are executed in the area of the bounding box of the modified features. Since other features may be present in the same area, topological errors concerning these features are reported as well as errors introduced in the current edit session.

Opção de verificação de topologia	Ilustração
The *Gap* check will check for gaps between neighbouring polygons.	
The *Overlap* check will check for overlaps between neighbouring polygons.	
The *Missing vertex* check will check for shared boundaries of neighbouring polygons where one border misses a vertex which is present on the other one.	

Gap check exceptions

Sometimes it is desirable to keep gaps inside an area in a polygon layer that otherwise is fully covered by polygons. For example, a land use layer may have acceptable holes for lakes. It is possible to define areas that are ignored in the gap check. Since gaps inside these areas are allowed, we will refer to them as *Allowed Gaps* areas.

In the options for the gap checks under *Allowed Gaps*, an *Allowed Gaps layer* can be configured.

Whenever the gap check is executed, gaps which are covered by one or more polygons in the *Allowed Gaps Layer* are not reported as topology errors.

It is also possible to configure an additional *Buffer*. This buffer is applied to each polygon on the *Allowed Gaps Layer*. This makes it possible to make the tests less susceptible to small changes in the outlines at the borders of gaps.

When *Allowed Gaps* are enabled, an additional button (*Add Allowed Gap*) for detected gap errors is available in the geometry validation dock, where gaps are reported during digitizing. If the *Add Allowed Gap* button is pushed, a new polygon with the geometry of the detected gap is inserted into the *Allowed Gaps Layer*. This makes it possible to quickly flag gaps as allowed.

Painel de validação geométrica

The *Geometry Validation* panel is triggered when any of the abovementioned digitizing checks finds an error. The dialog provides you with the list of errors and their description, and you can to browse the list using the keyboard arrows or dedicated arrows.

Você precisará resolver todos os problemas antes de salvar as edições na camada. Para fazer isso:

1. Selecione um erro, e é possível:
   - *Zoom to Feature(s)*
   - *Zoom to problem*
2. Pick the usual *digitizing tools* to fix the issue.

12.2 Trabalhando com a tabela de atributos

A tabela de atributos exibe informações sobre as feições de uma camada selecionada. Cada linha da tabela representa uma feição (com ou sem geometria) e cada coluna contém uma informação específica sobre a feição. As feições da tabela podem ser pesquisadas, selecionadas, movidas ou mesmo editadas.

12.2.1 Prefácio: Tabelas espaciais e não espaciais

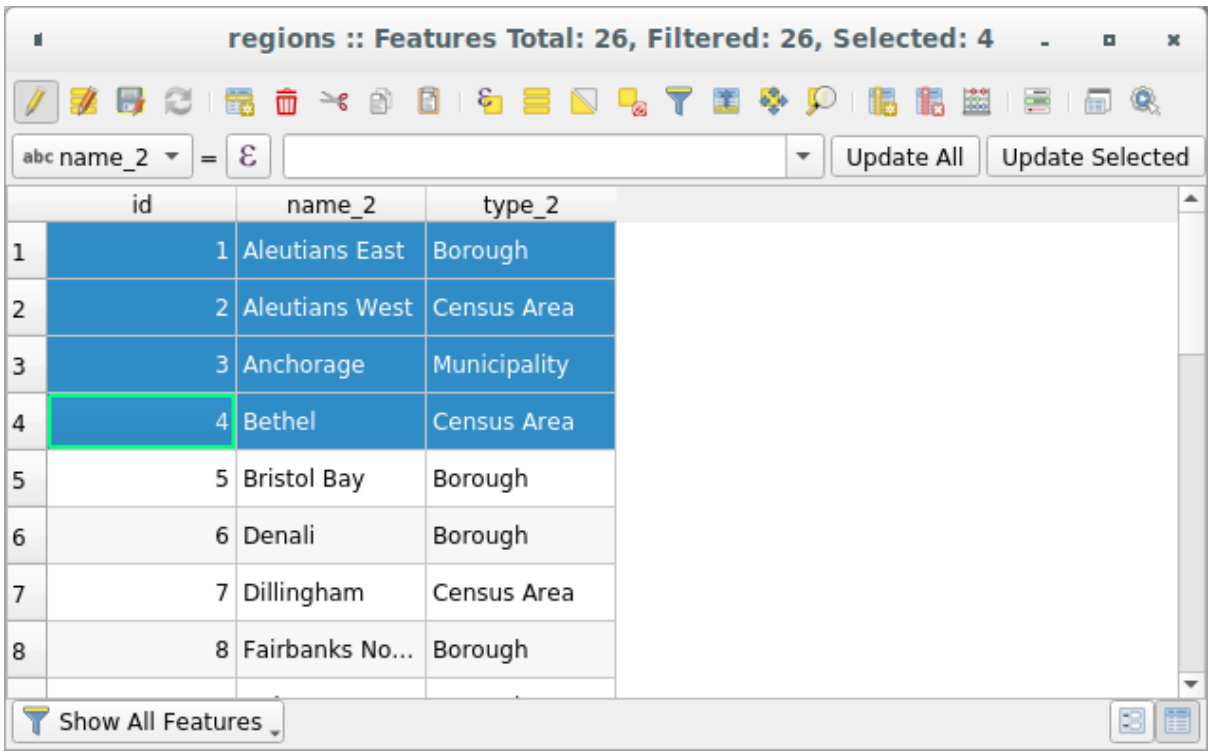
QGIS allows you to load spatial and non-spatial layers. This currently includes tables supported by GDAL and delimited text, as well as the PostgreSQL, MS SQL Server, SpatiaLite and Oracle providers. All loaded layers are listed in the *Layers* panel. Whether a layer is spatially enabled or not determines whether you can interact with it on the map.

As tabelas não espaciais podem ser navegadas e editadas usando a visualização da tabela de atributos. Além disso, eles podem ser usados para pesquisas de campo. Por exemplo, você pode usar colunas de uma tabela não espacial para definir valores de atributos ou um intervalo de valores permitidos para serem adicionados a uma camada vetorial específica durante a digitalização. Dê uma olhada no widget de edição na seção *Propriedades do formulário de atributos* para descobrir mais.

12.2.2 Apresentando a Interface da Tabela de Atributos

Para abrir a tabela de atributos para uma camada vetorial, ative a camada clicando nela *Painel Camadas*. Em seguida, no menu principal: seleção de menus: menu *Camada*, escolha *Abrir tabela de atributos*. Também é possível clicar com o botão direito do mouse na camada e escolher *Abrir Tabela de Atributo* no menu suspenso ou clique em :guilabel:botão *Abrir tabela de atributos* na barra de ferramentas Atributos. Se você preferir atalhos, `F6` abrirá a tabela de atributos. `Shift+F6` abrirá a tabela de atributos filtrada para os recursos selecionados e: kbd:*Ctrl+F6* abrirá a tabela de atributos filtrados para os recursos visíveis.

Isso abrirá uma nova janela que exibe os atributos da feição para a camada (*figure_attributes_table*). De acordo com a configuração em :menuselection:menu *Configurações –> Opções –> Fonte de dados*, a tabela de atributos será aberta em uma janela encaixada ou em uma janela regular. O número total de feições na camada e o número de feições atualmente selecionados/filtrados são mostrados no título da tabela de atributos, bem como se a camada é espacialmente limitada.



	id	name_2	type_2
1	1	Aleutians East	Borough
2	2	Aleutians West	Census Area
3	3	Anchorage	Municipality
4	4	Bethel	Census Area
5	5	Bristol Bay	Borough
6	6	Denali	Borough
7	7	Dillingham	Census Area
8	8	Fairbanks No...	Borough

Figura12.70: Tabela de atributos para a camada de regiões

Os botões na parte superior da janela da tabela de atributos fornecem a seguinte funcionalidade:

Tabela12.1: Ferramentas Disponíveis

Ícone	Etiqueta	Finalidade	Atalh
	Alternar modo de edição	Habilitar funcionalidades de edição	`Ctrl`
	Alternar modo multi edição	Atualize vários campos de várias feições	
	Salvar Edições	Salvar modificações atuais	
	Recarregar a tabela		
	Adicionar feição	Adicionar nova feição sem geometria	
	Excluir feições selecionadas	Remover feições selecionadas para a camada	

continua na próx

Tabela 12.1 – continuação da página anterior

Ícone	Etiqueta	Finalidade	Atalh
	Cortar feições selecionadas para área de transferência		Ctrl
	Copiar feições selecionadas para a área de transferência		Ctrl
	Colar feições da área de transferência	Inserir novas feições das copiadas	Ctrl
	Selecionar feições usando uma expressão		
	Selecionar Todas	Selecionar todas feições em uma camada	Ctrl
	Seleção invertida	Inverter a seleção atual na camada	Ctrl
	Desselecionar todas	Desselecionar todas as feições na camada atual	Ctrl
	Filtrar/Selecionar feições usando formulário		Ctrl
	Mover selecionadas para o topo	Mover linhas selecionadas para o todo da tabela	
	Mapa panorâmico para selecionar linhas		Ctrl
	Aproximar mapa para selecionar linhas		Ctrl
	Novo campo	Adicionar um novo campo para uma fonte de dados	Ctrl
	Excluir campo	Remover um campo a partir da fonte de dados	
	Organizar colunas	Mostrar/ocultar campos da tabela de atributos	
	Abrir calculadora de campo	Atualização de campo para muitas feições em uma linha	Ctrl
	Formatação condicional	Ativar formatação de tabela	
	Tabela de Atributos do Dock	Permite encaixar/desencaixar a tabela de atributos	
	Actions	Listas de ações relatadas para uma camada	

Atenção

Depending on the format of the data and the GDAL library built with your QGIS version, some tools may not be available.

Abaixo desses botões está a barra Cálculo de campo rápido (ativada apenas em *modo de edição*), que permite aplicar rapidamente cálculos a todos ou parte das feições da camada. Esta barra usa o mesmo: ref:*expressões<vector_expressions>`como o |calcula campo| :sup:`Calculadora de campo* (consulte: ref:` calcule_campos_valores`).

Visualização da tabela versus visualização do formulário

O QGIS fornece dois modos de exibição para manipular facilmente os dados na tabela de atributos:

- The Table view, displays values of multiple features in a tabular mode, each row representing a feature and each column a field. A right-click on the column header allows you to *configure the table display* while a right-click on a cell provides *interaction with the feature*.

  The attribute table supports `Shift+Mouse Wheel` scrolling in table view mode to switch between vertical and horizontal scrolling movements. This can also be achieved replacing the mouse with the trackpad on macOS.

- The Form view shows *feature identifiers* in a first panel and displays only the attributes of the clicked identifier in the second one. There is a pull-down menu at the top of the first panel where the “identifier” can be specified

using an attribute (*Column preview*) or an *Expression*. The pull-down also includes the last 10 expressions for re-use. Form view uses the layer fields configuration (see *Propriedades do formulário de atributos*).

Você pode navegar pelos identificadores de feições com as setas na parte inferior do primeiro painel. Os atributos de feições são atualizados no segundo painel conforme você avança. Também é possível identificar ou mover para a feição ativa o na tela do mapa pressionando qualquer botão na parte inferior:

- Highlight current feature if visible in the map canvas
- Automatically pan to current feature
- Zoom para a feição atual

Você pode alternar de um modo para outro clicando no ícone correspondente no canto inferior direito da caixa de diálogo.

Você também pode especificar o modo *Visualização padrão* na abertura da tabela de atributos em :menuselection:menu *Configurações -> Opções -> Fonte de Dados*. Pode ser 'Lembrar última visualização', 'Tabela' ou 'Formulário'.

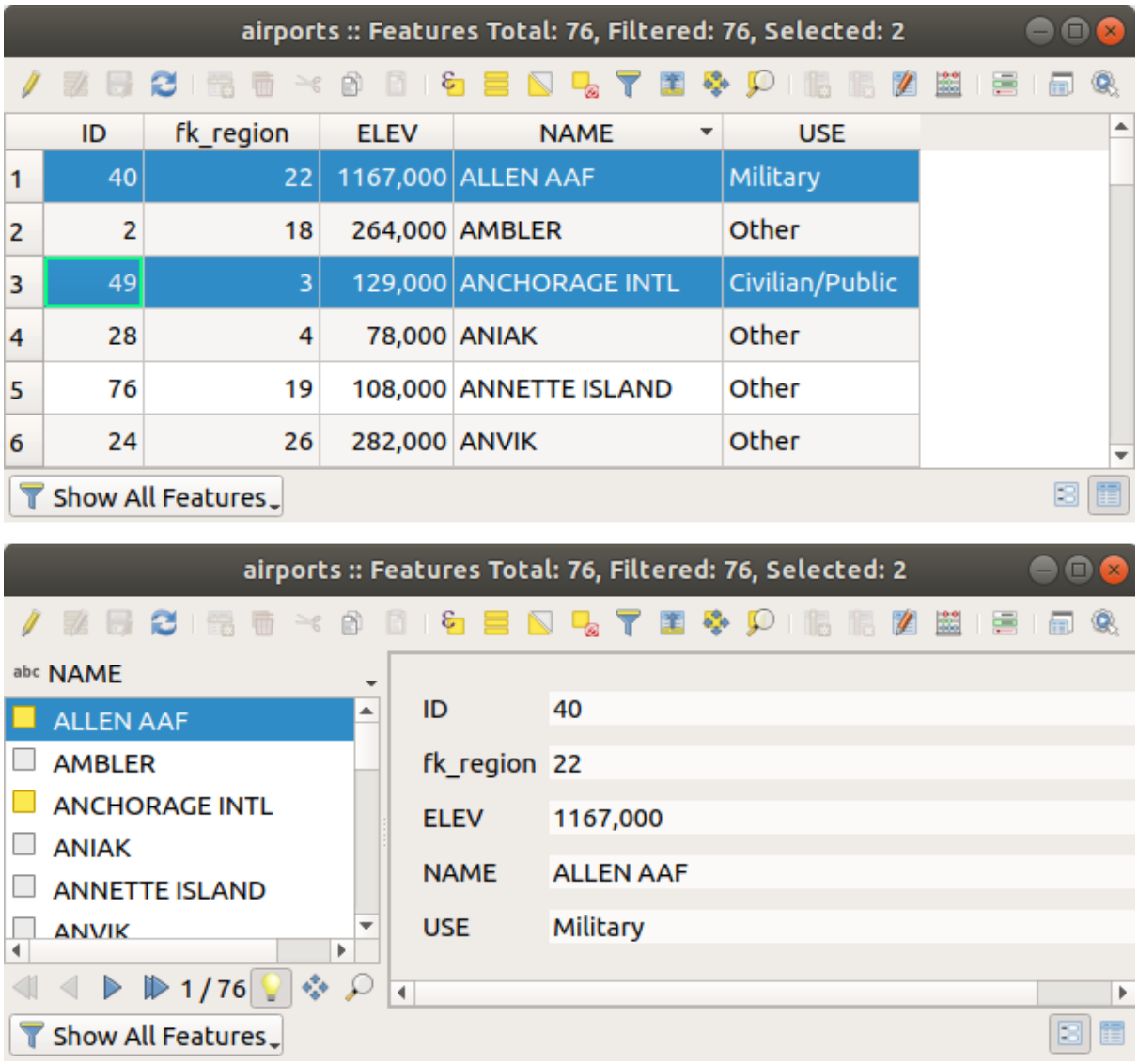


Figura12.71: Tabela de atributos na exibição de tabela (superior) vs exibição de formulário (inferior)

Configurando as colunas

Right-click in a column header when in table view to have access to tools that help you control:

- the *tamanho(s) da(s) coluna(s)*
- a *coluna(s) visibilidade e ordem*
- the *sort order of the data*

Redimensionando larguras de colunas

A largura das colunas pode ser definida através de um clique com o botão direito do mouse no cabeçalho da coluna e selecione:

- *Defina a largura…* para inserir o valor desejado. Por padrão, o valor atual é exibido no widget
- *Definir todas as larguras de coluna…* para o mesmo valor
- *Autodimensionar* para redimensionar da melhor forma possível a coluna.
- *Auto dimensionar todas as colunas*

O tamanho de uma coluna também pode ser alterado arrastando o limite à direita de seu título. O novo tamanho da coluna é mantido para a camada e restaurado na próxima abertura da tabela de atributos.

In the *Data Source Settings*, you can choose to *Autosize all columns by default when opening attribute table*, which will make “Autosize All Columns” the default view every time attribute tables are opened in QGIS.

Ocultando e organizando colunas e permitindo ações

By right-clicking in a column header, you can choose to *Hide column* from the attribute table (in “table view” mode). For more advanced controls, press the Organize columns… button from the dialog toolbar or choose *Organize columns…* in a column header contextual menu.

In the new dialog, you can:

- check/uncheck columns you want to show or hide: a hidden column will disappear from every instance of the attribute table dialog until it is actively restored. It is also possible to:
  - choose *Show All* to display all the fields (columns) and actions in the table
  - choose *Hide All* to hide all the fields (columns) and actions in the table
  - use the *Toggle selection* to invert visibility of the current selection of columns. You can use *keyboard combination* for selecting multiple columns.
- arraste e solte itens para reordenar as colunas na tabela de atributos. Observe que essa alteração é para a renderização da tabela e não altera a ordem dos campos na fonte de dados da camada
- add a new virtual *Actions* column that displays in each row a drop-down box or a button list of enabled actions. See *Propriedades de Ações* for more information about actions.

Sorting rows

The rows can be sorted by any column, by clicking on the column header. A small arrow indicates the sort order (downward pointing means descending values from the top row down, upward pointing means ascending values from the top row down). You can also choose to sort the rows with the *Sort...* option of the column header context menu and write an expression. E.g. to sort the rows using multiple columns you can write `concat(col0, col1)`.

Na exibição de formulário, o identificador de feições pode ser classificado usando o :guilabel:opção Classificar por expressão de visualização.

Note that sorting the rows only affects the table rendering and does not alter the features order in the layer datasource.

> **Dica**
>
> **Classificação com base em colunas de diferentes tipos**
>
> Trying to sort an attribute table based on columns of string and numeric types may lead to unexpected result because of the `concat("USE", "ID")` expression returning string values (ie, `'Borough105' < 'Borough6'`). You can workaround this by using eg `concat("USE", lpad("ID", 3, 0))` which returns `'Borough105' > 'Borough006'`.

Formatação de células da tabela usando condições

As configurações de formatação condicional podem ser usadas para realçar nas feições da tabela de atributos nos quais você deseja enfatizar, usando condições personalizadas nas feições:

- geometria (por exemplo, identificando características de várias partes, áreas pequenas ou em uma extensão de mapa definida …);
- or field value (e.g., comparing values to a threshold, identifying empty cells, duplicates, …).

You can enable the conditional formatting panel clicking on Conditional formatting button at the top right of the attributes window in table view (not triggered in form view).

O novo painel permite que o usuário adicione novas regras para formatar a renderização de *Campo* ou *Linha completa*. Adicionar nova regra abre um formulário para definir:

- o nome da regra;
- uma condição usando qualquer uma das funções *construtor de expressões*;
- a formatação: pode ser escolhida de uma lista de formatos predefinidos ou criada com base em propriedades como:
  - cores de fundo e texto;
  - uso de ícone;
  - negrito, itálico, sublinhado ou strikeout;
  - fonte.

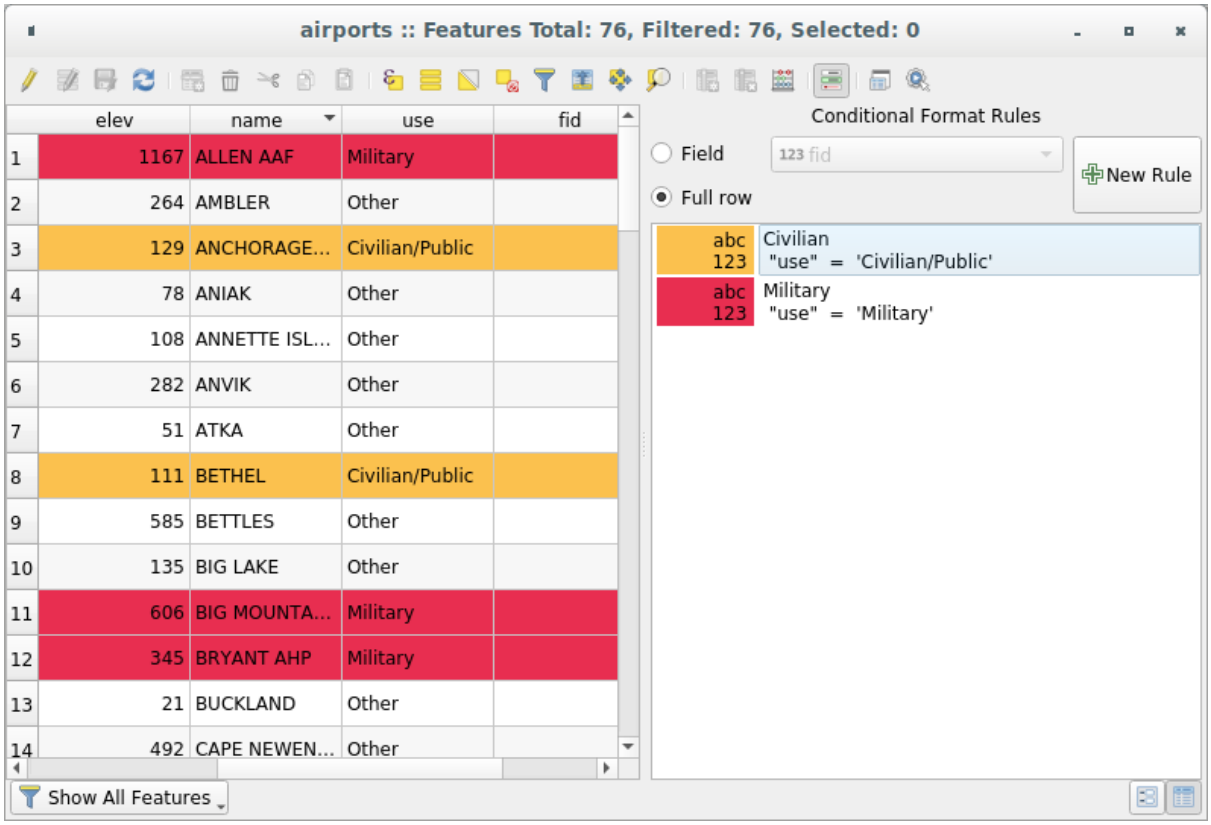


Figura12.72: Formatação condicional de uma tabela de atributo

12.2.3 Interagindo com feições em uma tabela de atributos

Selecionando características

Na exibição de tabela, cada linha da tabela de atributos exibe os atributos de um recurso exclusivo da camada. Selecionar uma linha seleciona o recurso e, da mesma forma, selecionar um recurso na tela do mapa (no caso de camada ativada para geometria) seleciona a linha na tabela de atributos. Se o conjunto de recursos selecionados na tela do mapa (ou tabela de atributos) for alterado, a seleção também será atualizada na tabela de atributos (ou tela do mapa) de acordo.

As linhas podem ser selecionadas clicando no número da linha no lado esquerdo da linha. **Múltiplas linhas** podem ser marcadas pressionando a tecla: kbd:*Ctrl*. Uma **seleção contínua** pode ser feita pressionando a tecla: kbd:*Shift* e clicando em vários cabeçalhos de linha no lado esquerdo das linhas. Todas as linhas entre a posição atual do cursor e a linha clicada são selecionadas. Mover a posição do cursor na tabela de atributos, clicando em uma célula na tabela, não altera a seleção de linha. Alterar a seleção na tela principal não move a posição do cursor na tabela de atributos.

Na exibição de formulário da tabela de atributos, as feições são identificados por padrão no painel esquerdo pelo valor do campo exibido (consulte:ref:*maptips*). Esse identificador pode ser substituído usando a lista suspensa na parte superior do painel, selecionando um campo existente ou usando uma expressão personalizada. Você também pode optar por classificar a lista de recursos no menu suspenso.

Clique em um valor no painel esquerdo para exibir os atributos da feição no caminho certo. Para selecionar uma feição, você precisa clicar dentro do símbolo quadrado à esquerda do identificador. Por padrão, o símbolo fica amarelo. Como na exibição de tabela, você pode executar várias seleções de recursos usando as combinações de teclado expostas anteriormente.

Beyond selecting features with the mouse, you can perform automatic selection based on feature's attribute using tools available in the attribute table toolbar, such as (see section *Seleção automática* and subsequent for more information and use case):

- *Selecione por expressão…*
- *Selecionar Feição pelo Valor…*
- *Desmarcar todas as feições da camada*
- *Selecionar Todas as Feições*
- *Inverter Feições Selecionadas.*

It is also possible to *select features using forms*.

Filtragem de feições

Depois de selecionar as feições na tabela de atributos, você pode exibir apenas esses registros na tabela. Isso pode ser feito facilmente usando o item *Mostrar feições selecionadas* na lista suspensa no canto inferior esquerdo da caixa de diálogo da tabela de atributos. Esta lista oferece os seguintes filtros:

- *Mostrar Todas as Feições*
- *Show Selected Features* - same as using *Open Attribute Table (Selected Features)* from the *Layer* menu or the *Attributes Toolbar* or pressing `Shift+F6`
- *Show Features visible on map* - same as using *Open Attribute Table (Visible Features)* from the *Layer* menu or the *Attributes Toolbar* or pressing `Ctrl+F6`
- *Show Features with Failing Constraints* - features will be filtered to only show the ones which have failing *constraints*. Depending on whether the unmet constraint is hard or soft, failing field values are displayed in respectively dark or light orange cells.
- *Show Edited and New Features* - same as using *Open Attribute Table (Edited and New Features)* from the *Layer* menu or the *Attributes Toolbar*
- *Field Filter* - allows the user to filter based on value of a field: choose a column from a list, type or select a value and press `Enter` to filter. Then, only the features matching `num_field = value` or `string_field ilike '%value%'` expression are shown in the attribute table. You can check *Case sensitive* to be less permissive with strings.
- *Advanced filter (Expression)* - Opens the expression builder dialog. Within it, you can create *complex expressions* to match table rows. For example, you can filter the table using more than one field. When applied, the filter expression will show up at the bottom of the form.
- *Stored filter expressions* ►: a shortcut to *saved expressions* frequently used for filtering your attribute table.

Também é possível:ref:*filtrar recursos usando formulários <filter_select_form>*.

Nota

A filtragem de registros da tabela de atributos não filtra feições da camada; eles são simplesmente ocultados momentaneamente da tabela e podem ser acessados na tela do mapa ou removendo o filtro. Para filtros que ocultam recursos da camada, use o *Query Builder*.

Dica

Atualize a filtragem da fonte de dados com `Mostrar feições visíveis no mapa`

Quando, por motivos de desempenho, os recursos mostrados na tabela de atributos são espacialmente limitados à extensão da tela em sua abertura (consulte:ref:*Opções da fonte de dados<tip_table_filtering>* para obter instruções), selecionando:guilabel:*Mostrar feições visíveis no mapa* em um nova extensão de tela atualiza a restrição espacial.

Armazenamento de expressões de filtro

Expressions you use for attribute table filtering can be saved for further calls. When using *Field Filter* or *Advanced Filter (expression)* entries, the expression used is displayed in a text widget in the bottom of the attribute table dialog. Press the Save expression with text as name next to the box to save the expression in the project. Pressing the drop-down menu next to the button allows to save the expression with a custom name (*Save expression as…*). Once a saved expression is displayed, the button is triggered and its drop-down menu allows you to *Edit the expression* and name if any, or *Delete stored expression*.

Saved filter expressions are saved in the project and available through the *Stored filter expressions* menu of the attribute table. They are different from the *user expressions*, shared by all projects of the active user profile.

Filtrando e selecionando feições usando formulários

Clicando no Filtre/selecione recursos usando o formulário ou pressione :kbd:Ctrl+F` fará com que a caixa de diálogo da tabela de atributos mude para a exibição do formulário e substitua cada widget por sua variante de pesquisa.

A partir deste ponto, a funcionalidade desta ferramenta é semelhante à descrita em:ref:*select_by_value*, onde é possível encontrar descrições de todos os operadores e selecionar modos.

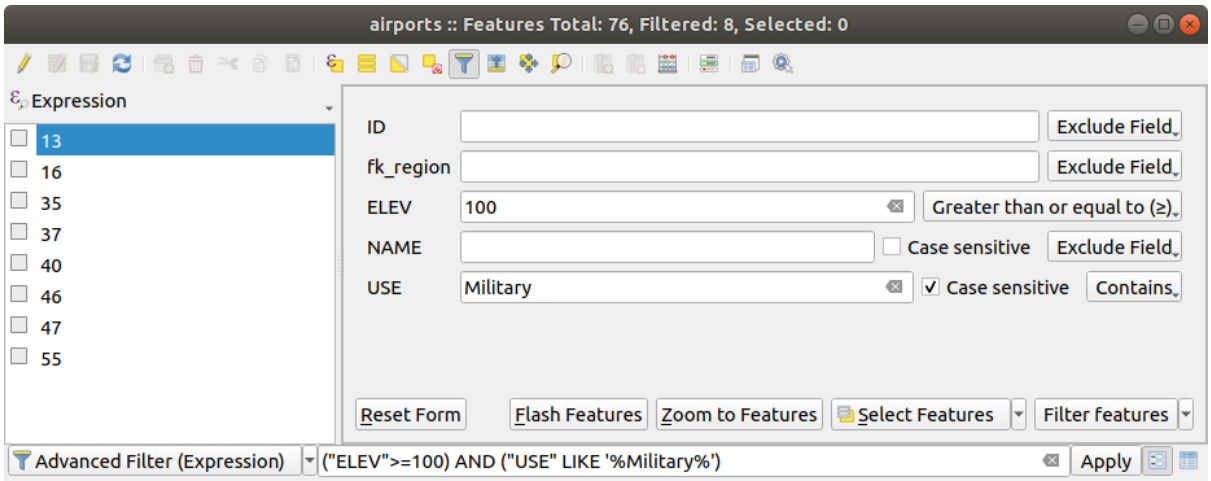


Figura12.73: Tabela de atributos filtrada pelo formulário de filtro

Ao selecionar / filtrar feições da tabela de atributos, existe um botão *Filtro de feições* que permite definir e refinar filtros. Seu uso aciona a opção:guilabel:*Filtro avançado (Expressão)* e exibe a expressão de filtro correspondente em um widget de texto editável na parte inferior do formulário.

Se já houver feições filtradas, você poderá refinar o filtro usando a lista suspensa ao lado do botão *Filtro de feições*. As opções são:

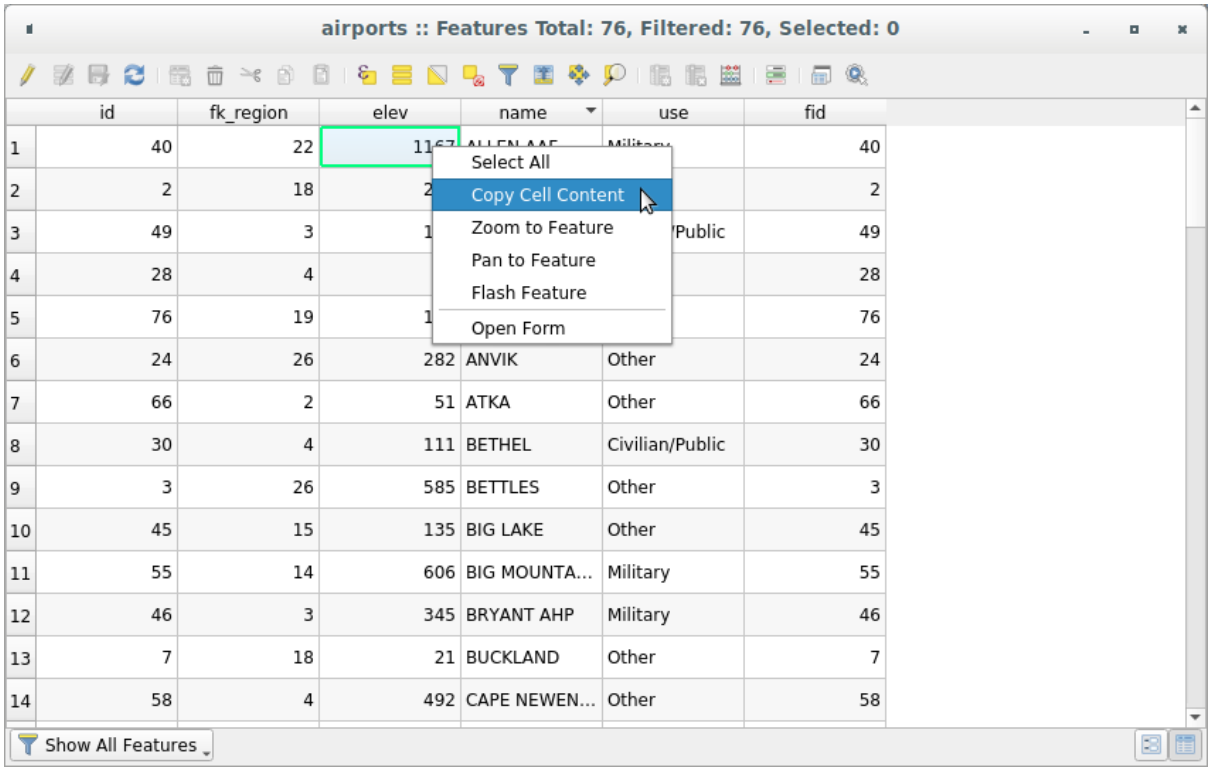
- *Filtrar dentro de ("AND")*
- *Estender filtro ("OR")*

Para limpar o filtro, selecione a opção *Mostrar todas as feições* no menu suspenso inferior esquerdo ou limpe a expressão e clique em *Aplicar* ou pressione `Enter`.

More actions on features

Users have several possibilities to manipulate feature in an attribute table. Right-click in a cell and you can:

- *Selecionar todas* (`Ctrl+A`) as feições;
- Copie o conteúdo de uma célula na área de transferência com: guilabel:*Copiar conteúdo da célula*;
- *Aproximar na feição* sem ter que selecioná-lo previamente;
- *Panorâmica da feição* sem ter que selecioná-lo previamente;
- *Destacar feição*, para destacá-la na tela do mapa;
- *Abrir formulário*: alterna a tabela de atributos para a exibição de formulários, com foco na feição clicado.
- Display a *list of actions*, previously enabled in the *Layer properties* ► *Actions* tab.



	id	fk_region	elev	name	use	fid
1	40	22	1167	ALLEN AAF	Military	40
2	2	18	2			2
3	49	3	1		Public	49
4	28	4				28
5	76	19	1			76
6	24	26	282	ANVIK	Other	24
7	66	2	51	ATKA	Other	66
8	30	4	111	BETHEL	Civilian/Public	30
9	3	26	585	BETTLES	Other	3
10	45	15	135	BIG LAKE	Other	45
11	55	14	606	BIG MOUNTA...	Military	55
12	46	3	345	BRYANT AHP	Military	46
13	7	18	21	BUCKLAND	Other	7
14	58	4	492	CAPE NEWEN...	Other	58

Figura12.74: Copiar botão de conteúdo da célula

If you want to use attribute data in external programs (such as Excel, LibreOffice, or a custom web application), select one or more row(s) and use the Copy selected rows to clipboard button or press `Ctrl+C`. Moreover, in *Settings* ► *Options* ► *Data Sources* menu you can define the format to paste to with the *Copy features as* option. More details at *Configurações de fonte de informação*.

12.2.4 Editando valores de atributo

In order to modify data in an attribute table, you should first toggle the layer into edit. Press the Toggle Editing button. Depending on the layer geometry type and the clipboard state, a few more tools are enabled in the attribute table top toolbar.

Editing attribute values can then be done by:

- digitando o novo valor diretamente na célula, esteja a tabela de atributos na visualização de tabela ou formulário. As mudanças são feitas célula por célula, feição por feição;
- using the *field calculator*: update in a row a field that may already exist or to be created but for multiple features. It can be used to create virtual fields;
- using the quick field *calculation bar*: same as above but for only existing field;
- or using the *multi edit* mode: update in a row multiple fields for multiple features.

Putting the layer into edit mode will also allow you to Paste features from clipboard (`Ctrl+V`) Cut selected rows to clipboard (`Ctrl+X`) or Delete selected features. More details at *Cortando, Copiando e Colando Elementos*.

Utilizando a Calculadora de Campo

The Field Calculator button in the attribute table allows you to perform calculations on the basis of existing attribute values or defined functions, for instance, to calculate length or area of geometry features. The results can be used to update an existing field, or written to a new field (that can be a *virtual* one).

The field calculator is available on any layer that supports edit. When you click on the field calculator icon the dialog opens (see Fig.12.75). If the layer is not in edit mode, a warning is displayed and using the field calculator will cause the layer to be put in edit mode before the calculation is made.

Based on the *Expression Builder* dialog, the field calculator dialog offers a complete interface to define an expression and apply it to an existing or a newly created field. To use the field calculator dialog, you must select whether you want to:

1. aplicar o cálculo sobre toda a camada ou apenas sobre as características selecionadas
2. criar um novo campo para o cálculo ou atualizar um já existente.

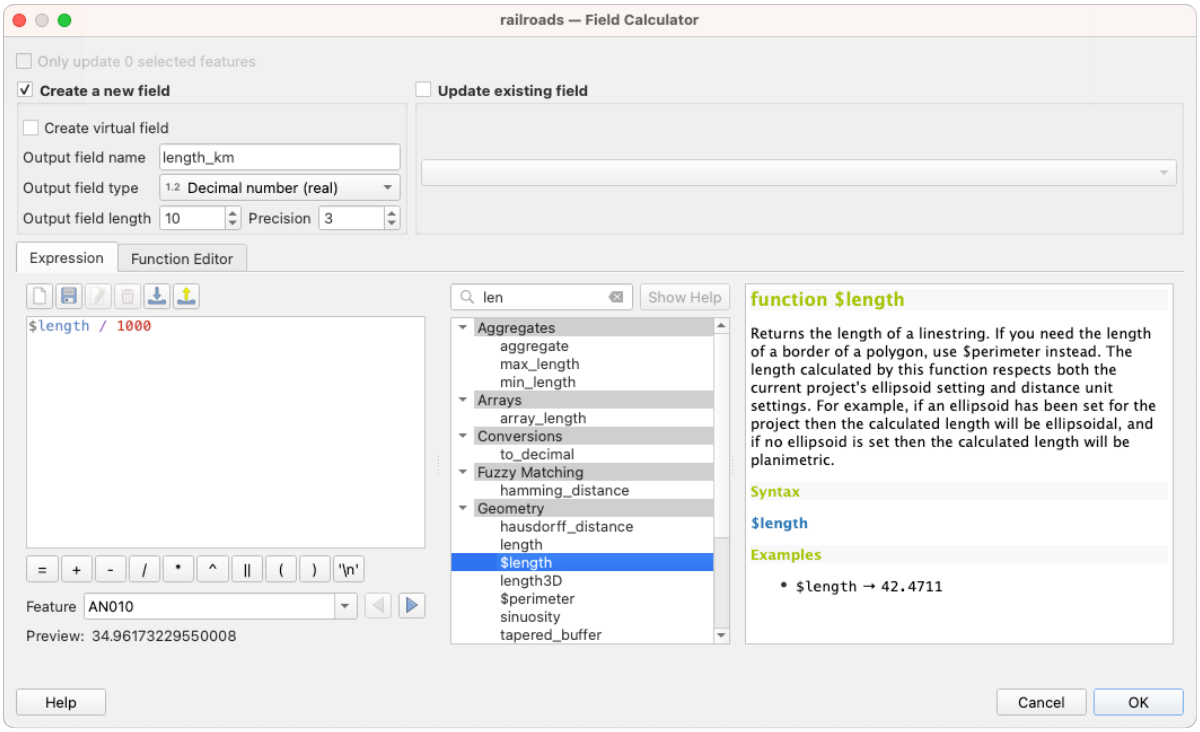


Figura12.75: Calculadora de campo

If you choose to add a new field, you need to enter a field name, a field type (integer, real, date or string) and if needed, the total field length and the field precision. For example, if you choose a field length of 10 and a field precision of 3, it means you have 7 digits before the dot, and 3 digits for the decimal part.

A short example illustrates how field calculator works when using the *Expression* tab. We want to calculate the length in km of the `railroads` layer from the QGIS sample dataset:

1. Load the shapefile `railroads.shp` in QGIS and press Open Attribute Table.
2. Click on Toggle editing mode and open the Field Calculator dialog.
3. Select the *Create a new field* checkbox to save the calculations into a new field.
4. Set *Output field name* to `length_km`
5. Selecione `Número decimal (real)` como *Tipo de campo de saída*`.
6. Set the *Output field length* to `10` and the *Precision* to `3`
7. Double click on `$length` in the *Geometry* group to add the length of the geometry into the Field calculator expression box (you will begin to see a preview of the output, up to 60 characters, below the expression box updating in real-time as the expression is assembled).
8. Complete the expression by typing `/ 1000` in the Field calculator expression box and click *OK*.
9. You can now find a new *length_km* field in the attribute table.

Criando um campo virtual

A virtual field is a field based on an expression calculated on the fly, meaning that its value is automatically updated as soon as an underlying parameter changes. The expression applies to all the features in the layer and is set once; you no longer need to recalculate the field each time underlying values change. For example, you may want to use a virtual field if you need area to be evaluated as you digitize features or to automatically calculate a duration between dates that may change (e.g., using `now()` function).

Creating a virtual field is done through the *Field calculator* dialog and follows the *same procedure* as for regular fields. Simply remember to check the *Create virtual field* option and use a field type compatible with the data your expression would generate.

Editing a virtual field is done through the *Fields* tab of the layer properties dialog (see *Propriedades dos campos*). The expression defining the field is exposed in the *Comment* column, and pressing the button next to it opens an expression editor window for update.

Nota

Uso de campos virtuais

- A field can be set virtual only at its creation.
- Os campos virtuais não são permanentes nos atributos da camada, o que significa que eles são salvos e disponibilizados apenas no arquivo de projeto em que foram criados.

Usando a barra de cálculo rápido de campo

While Field calculator is always available, the quick field calculation bar on top of the attribute table is only visible if the layer is in edit mode. Thanks to the expression engine, it offers a quicker access to edit an already existing field:

1. Selecione o campo a ser atualizado na lista suspensa.
2. Fill the textbox with a value, an expression you directly write or build using the expression button.
3. Click on *Update All*, *Update Selected* or *Update Filtered* button according to your need.

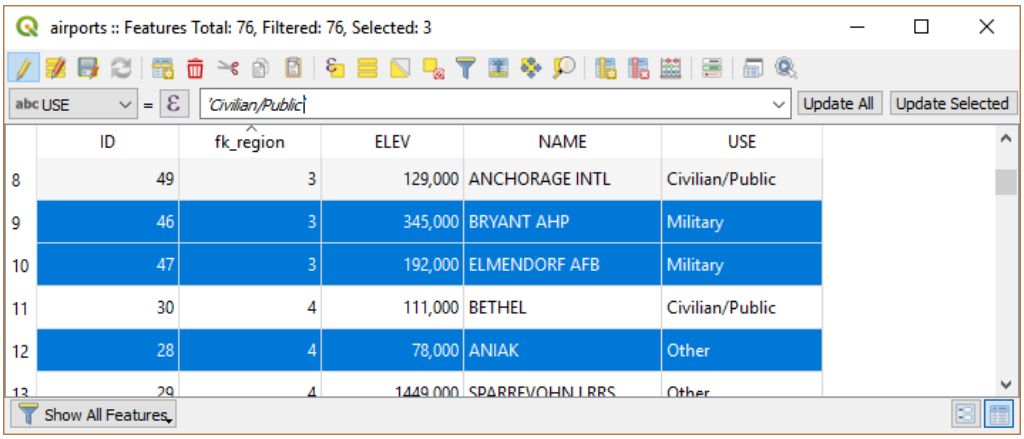


Figura12.76: Barra de cálculo rápido de campo

Editando vários campos

Ao contrário das ferramentas anteriores, o modo de edição múltipla permite que vários atributos de diferentes recursos sejam editados simultaneamente. Quando a camada é alternada para edição, os recursos de edição múltipla ficam acessíveis:

- using the Toggle multi edit mode button from the toolbar inside the attribute table dialog;
- or selecting *Edit* ► *Modify attributes of selected features* menu.

> **Nota**
>
> Unlike the tool from the attribute table, hitting the *Edit* ► *Modify Attributes of Selected Features* option provides you with a modal dialog to fill attributes changes. Hence, features selection is required before execution.

Para editar vários campos em uma linha:

1. Selecione as feições que você deseja editar.
2. From the attribute table toolbar, click the button. This will toggle the dialog to its form view. Feature selection could also be made at this step.
3. At the right side of the attribute table, fields (and values) of selected features are shown. New widgets appear next to each field allowing for display of the current multi edit state:

   - The field contains different values for selected features. It's shown empty and each feature will keep its original value. You can reset the value of the field from the drop-down list of the widget.
   - All selected features have the same value for this field and the value displayed in the form will be kept.
   - The field has been edited and the entered value will be applied to all the selected features. A message appears at the top of the dialog, inviting you to either apply or reset your modification.

   Clicking any of these widgets allows you to either set the current value for the field or reset to original value, meaning that you can roll back changes on a field-by-field basis.

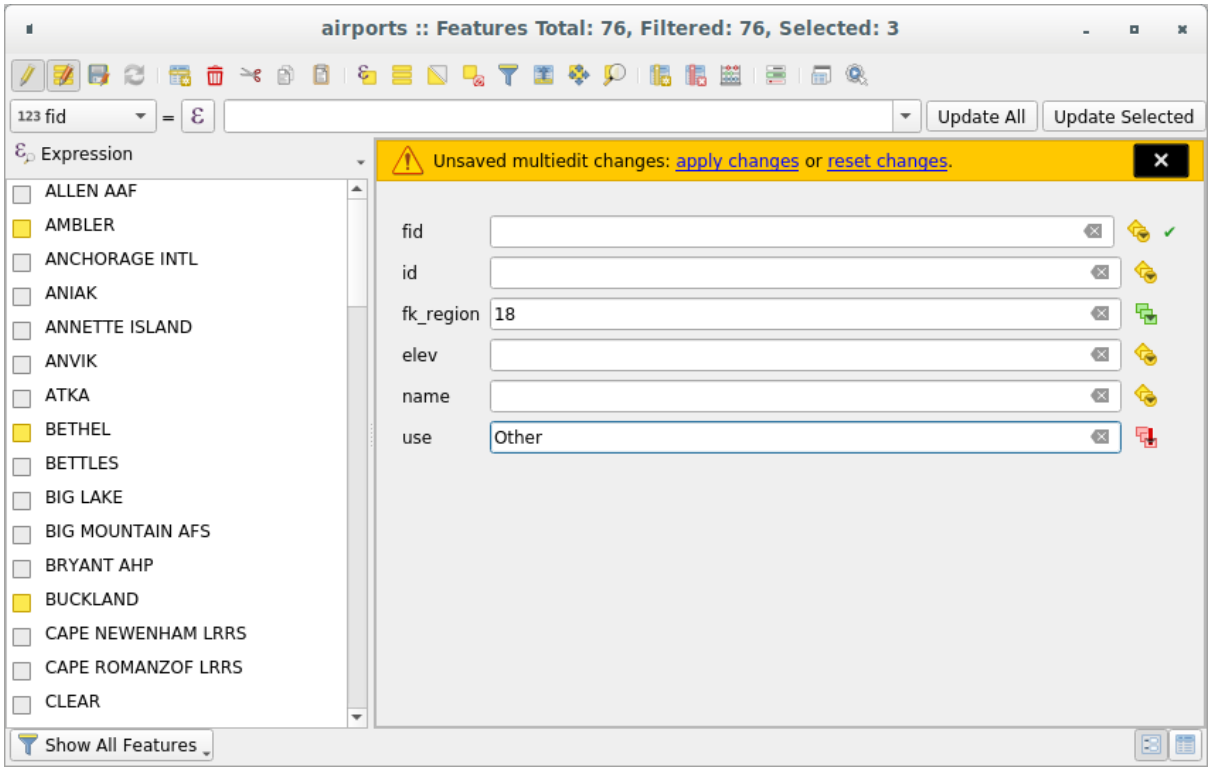


Figura12.77: Campos de edição de múltiplas feições

4. Faça as alterações nos campos desejados.
5. Clique em **Aplicar alterações** no texto da mensagem superior ou em qualquer outro recurso no painel esquerdo.

As alterações serão aplicadas a **todos os recursos selecionados**. Se nenhum recurso for selecionado, toda a tabela será atualizada com suas alterações. As modificações são feitas como um único comando de edição. Então, pressionando |desfazer| Desfazer reverterá as alterações de atributo para todos os recursos selecionados de uma só vez.

Nota

O modo de edição múltipla só está disponível para formulários gerados automaticamente e de arrastar e soltar (veja personalizar_formulário); ele não é suportado por formulários de interface do usuário personalizados.

12.2.5 Exploring features attributes through the Identify Tool

The *Identify features* tool can be used to display all attributes of a feature in the map canvas. It is a quick way to view and verify all data without having to search for it in the attribute table.

To use the *Identify features* tool for vector layers, follow these steps:

1. Select the vector layer in the Layers panel.
2. Click on the *Identify features* tool in the toolbar or press `Ctrl+Shift+I`.
3. Click on a feature in the map view.

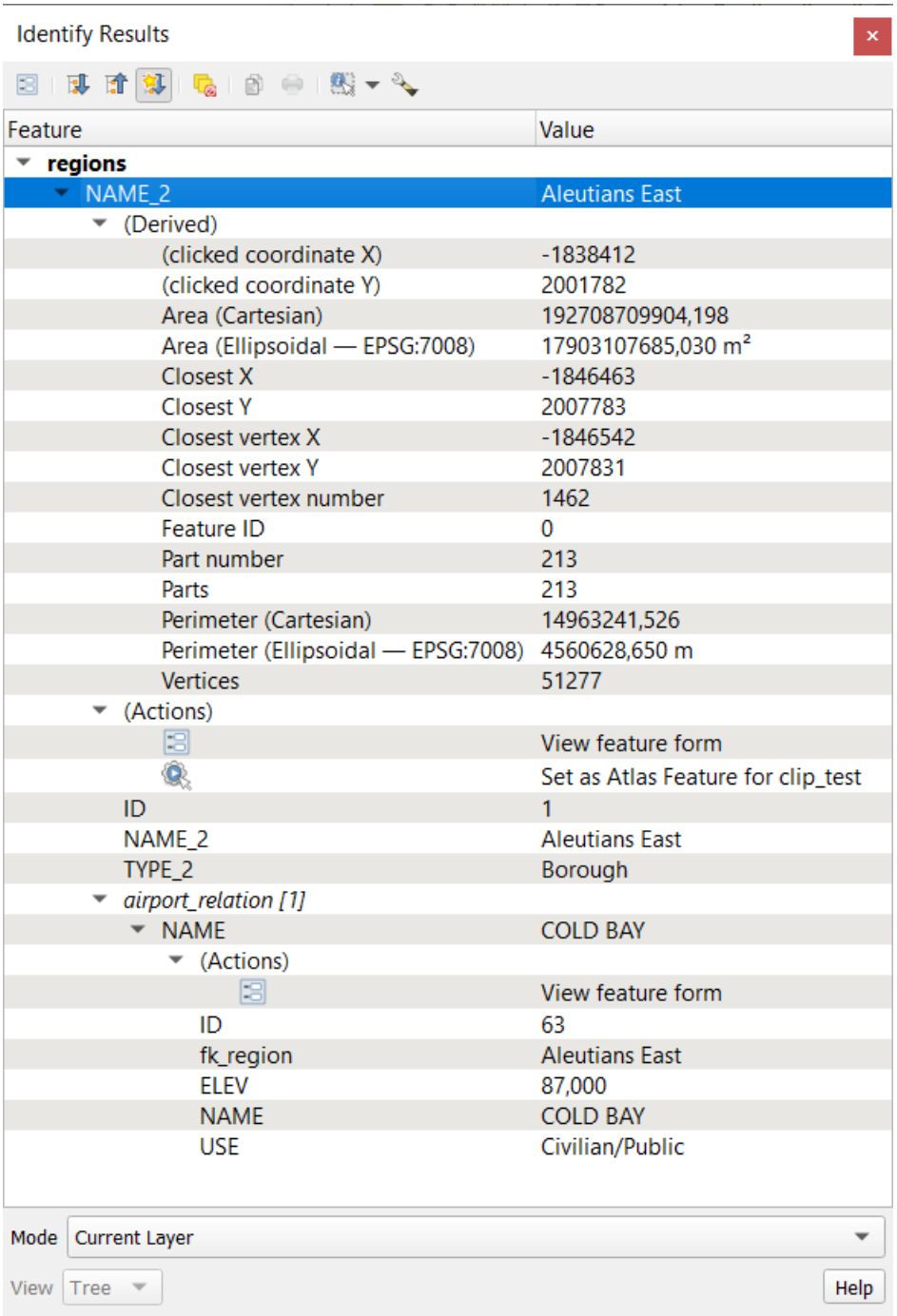


Figura12.78: Identify Results dialog

The *Identify results* panel will display different features information in a tree view. There are two columns in the panel: *Feature* and *Value*. The first item is the name of the layer and its children are its identified feature(s). Each feature is identified by the *display name* field along with its value. Other information about the feature follows:

- **Derived** section - those are the information calculated or derived from other information in the layer. For example, the area of a polygon or the length of a line. General information that can be found in this section:
  - Depending on the geometry type, the cartesian measurements of length, perimeter or area in the layer's CRS units. For 3D line vectors, the cartesian line length is available.
  - Depending on the geometry type and if an ellipsoid is set in the *Project Properties* dialog (*General* ► *Measurements*), ellipsoidal values of length, perimeter, or area using the specified units.

  - The count of geometry parts in the feature and the number of the part clicked.
  - The count of vertices in the feature.
  - Coordinate information, using the project properties *Coordinates display* settings:
    * `X` and `Y` coordinate values of the clicked point
    * the number of the closest vertex to the clicked point
    * `X` and `Y` coordinate values of the closest vertex (and `Z`/`M` if applicable)
    * if you click on a curved segment, the radius of that section is also displayed.
    * if both the vector layer and the project have vertical datums set and they differ, the `Z` value will be displayed for both datums.

- **Actions**: Actions can be added to the identify feature windows. The action is run by clicking on the action label. By default, only one action is added, namely `View feature form` for editing. You can define more actions in the layer's properties dialog (see *Propriedades de Ações*).
- **Data attributes**: This is the list of attribute fields and values for the feature that has been clicked.

  Nota

  Links in the feature's attributes are clickable from the *Identify Results* panel and will open in your default web browser.

- When a vector layer has defined *relations*, the *Identify Results* panel can display both **referenced** and **referencing** related features. To view these relations, ensure that the *Show relations* option is enabled in the *Identify Settings*. Available information for each related feature include:
  - the name of the relation and the linked layer
  - the entry in the referenced or referencing field, identifying the feature
  - the activated actions
  - the list of attribute fields and values

  You can expand related features to explore connected records through multiple levels, including many-to-many (n:m) and polymorphic relations. Only nodes you explicitly expand are loaded, preventing excessive nesting.

  To focus on a specific related record, right-click it and choose *Identify Feature*. This re-centers the identify results on the selected feature, starting a new identification tree from it and effectively limiting the visible nesting depth. Features already shown in ancestor nodes are automatically omitted to avoid duplicates or circular relations.

12.3 Editando

QGIS has various capabilities for editing OGR, SpatiaLite, PostgreSQL, MS SQL Server and Oracle Spatial vector layers and tables. They can be of 2D or 3D geometry type.

Nota

O procedimento para editar camadas do GRASS é diferente - consulte a seção *Digitalizando e editando uma camada vetorial GRASS* para mais detalhes.

Atenção

Edições Simultâneas

QGIS does not track if somebody else is editing the same feature at the same time as you are. The last person to save the edits wins.

Dica

Validando Edições

Continuous validation can be activated on a layer basis in the *Layer Properties* ► *Digitizing* tab. More at *Propriedades de Digitalização*.

12.3.1 Setting the snapping tolerance and search radius

Under the *Settings* ► *Options...* ► *Digitizing* menu, QGIS provides a number of parameters to configure default behaviour of editing tools. More information at *Definições de digitalização*.

For optimal and accurate editing of vector layer geometries, we need to set an appropriate value of snapping tolerance and search radius for features vertices. The *Snapping* group provides related options, namely handling of the snapping tolerance and the search radius.

- Snapping tolerance: When you add a new vertex or move an existing one, the snapping tolerance is the distance QGIS uses to search for the closest vertex or segment you are trying to connect to. If you are not within the snapping tolerance, QGIS will leave the vertex where you release the mouse button, instead of snapping it to an existing vertex or segment.

  The tolerance setting affects all tools that work with snapping and applies by default to new layers and projets. It can however be overridden at layer level (see *Snapping and Digitizing Options*).

- Search radius: *Search radius for vertex edits* is the distance QGIS uses to `search` for the vertex to select when you click on the map. If you are not within the search radius, QGIS will not find and select any vertex for editing.

A tolerância ao snap e o raio de pesquisa são definidos em `` unidades de mapa`` ou `` pixels``. Pode ser necessário experimentar para acertar. Se você especificar uma tolerância muito grande, o QGIS poderá se ajustar ao vértice errado, especialmente se você estiver lidando com um grande número de vértices nas proximidades. Quanto menor o raio de pesquisa, mais difícil será atingir o que você deseja mover.

12.3.2 Snapping and Digitizing Options

Global *snapping and digitizing settings* (snapping mode, tolerance value, and units…) can be overridden in the project from the *Project* ► *Snapping Options...* menu. In the *Snapping and Digitizing Options*, you can also configure some other properties (snapping layers, scale limit, topology…) The *Snapping Toolbar* gives access to most of these features.

By default, snapping is disabled in a project until you press the Enable snapping button or press `S`. The snapping mode, tolerance value, and units can also be configured in this toolbar.

Snapping properties

Existem três opções para selecionar as camadas nas quais ajustar:

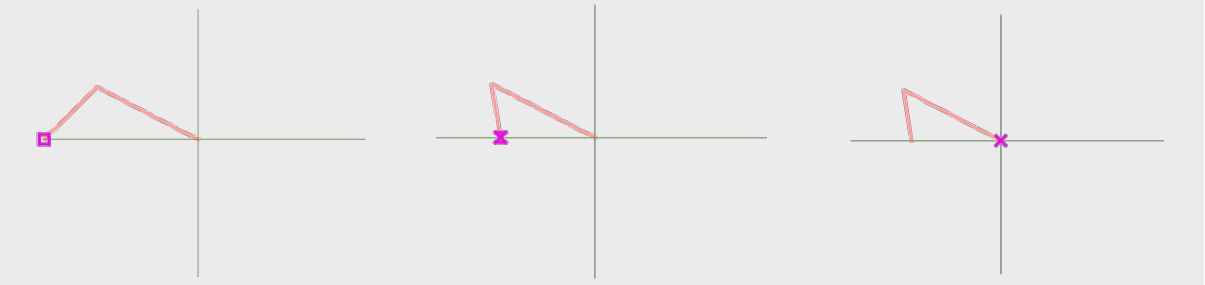
- *Todas as camadas*: configuração rápida para todas as camadas visíveis no projeto, para que o ponteiro se encaixe em todos os vértices e / ou segmentos. Na maioria dos casos, é suficiente usar esse modo de ajuste, mas tenha cuidado ao usá-lo em projetos com muitas camadas de vetores, pois isso pode afetar o desempenho.
- *Camada atual*: somente a camada ativa é usada, uma maneira conveniente de garantir consistência topológica na camada que está sendo editada.
- *Advanced Configuration*: allows you to enable and adjust snapping mode, tolerance and units, overlaps and scales of snapping on a layer basis (see Fig.12.79). If you need to edit a layer and snap its vertices to another, make sure that the target layer is checked and increase the snapping tolerance to a higher value. Snapping will not occur to a layer that is not checked in the snapping options dialog.

When moving or creating vertex, you can opt for the following snapping modes:

- *Vertex*
- *Segment*: snaps along a line or a polygon perimeter. If topological editing is enabled, then a new vertex is added at the snapping location.
- *Area*: guarantees that the snap point lies anywhere on a polygon's area, not necessarily on its boundary
- *Centroid*: snaps to the centroid of the geometry of a feature. In case of a multipart geometry, the target point may be distinct from the existing parts.
- *Middle of Segments* on line or polygon feature
- *Line Endpoints*: snaps to the first or last vertex of every part of a line or polygon feature.

QGIS will show different *snap* icons depending on the kind of *snap*:

Tabela12.2: Snapping icons



Snapping to a vertex: box icon	Snapping to a segment: hourglass icon	Snapping to an intersection: cross icon

Note that it is possible to change the color of these icons in the *Digitizing* part of the global settings.

Os valores de tolerância podem ser definidos nas `` unidades de mapa`` do projeto ou em `` pixels``. A vantagem de escolher `` pixels `` é que ele mantém o snap constante em diferentes escalas do mapa. 10 a 12 pixels é normalmente um bom valor, mas depende do DPI da sua tela. O uso de unidades de mapa permite que a tolerância esteja relacionada às distâncias reais do solo. Por exemplo, se você tem uma distância mínima entre elementos, essa opção pode ser útil para garantir que você não adicione vértices muito próximos um do outro.

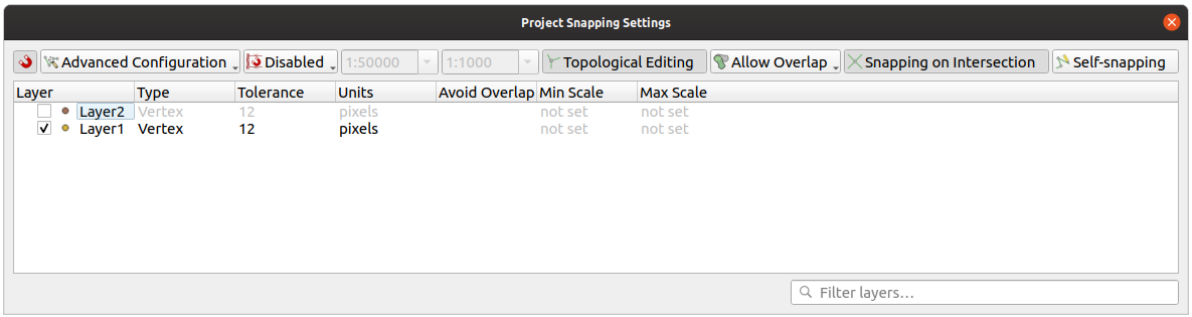


Figura12.79: Opções de ajuste (modo de configuração avançada)

Nota

Por padrão, somente os recursos visíveis (os recursos cujo estilo é exibido, exceto as camadas em que a simbologia é "Nenhum símbolo") podem ser ajustados. Você pode ativar o snap em recursos invisíveis marcando | desmarcado | *Habilite o snap em recursos invisíveis* na seção: menuselection:` Settings -> Options -> Digitizing`

Dica

** Ativar ajuste por padrão **

Você pode definir o encaixe para ser ativado por padrão em todos os novos projetos na:menuselection:*Settings -> Options -> Digitizing* Você também pode definir o modo de snap padrão, o valor de tolerância e as unidades, que preencherão a caixa de diálogo:guilabel:opções de snap.

Activar atracção nas intersecções

Outra opção disponível é usar | snappingIntersection | *encaixe na interseção*, que permite encaixar nas interseções geométricas das camadas habilitadas para encaixe, mesmo que não haja vértices nas interseções.

Limit snapping to a scale range

In some cases snapping can become very slow. This is often caused by the amount of features in some layers that require a heavy index to compute and maintain. Some parameters exist to enable snapping only when the map view is inside a relevant scale range. This allows to only do the costly index computation related to snapping at a scale where drawing is relevant.

Scale limit to snapping is configured in *Project* ► *Snapping Options…*. Limiting snapping to scale is only available in *Advanced Configuration* mode.

To limit snapping to a scale range you have three modes available:

- *Disabled*: Snapping is enabled whatever the current map scale is. This is the default mode.
- *Global*: Snapping is limited and only enabled when the current scale of the map is between a global minimum and a global maximum value. When selecting this mode two widgets become available to configure the range of scales in which snapping is enabled.
- *Per layer*: The snapping scale range limit is defined for each layer. When selecting this mode two columns become available to configure the minimum and maximum scales for each layer.

Please note that the minimum and maximum scales follow the QGIS convention: minimum scale is the most "zoomed out" scale while maximum scale is the most "zoomed in". A minimum or maximum scale that is set to "0" or "not set" is considered not limiting.

Self-snapping

The Self-snapping option allows you to snap to the geometry that is being edited. Combined with the *advanced digitizing panel*, this provides a handy way to digitize new edges relative to the previous edges or vertices. Self-snapping can cause invalid geometries, use with caution.

Snapping on custom grid

A snapping distance can also be customized on a layer basis in the *Digitizing* tab of the layer properties dialog. With setting the *Geometry precision* distance, you enable a dotted grid visible when the map canvas is at a coherent scale for display. Snapping can then be performed on the dots of the grid: an added or modified geometry will have all of its vertices snapped automatically to the closest node of the grid. More information at *Propriedades de Digitalização*.

12.3.3 Edição Topológica

In addition to these snapping options, the *Snapping options…* dialog (*Project* ► *Snapping options*) and the *Snapping* toolbar allow you to enable / disable some other topological functionalities.

Ativar edição topológica

A | edição topopológica | :sup:botão *Edição topológica* ajuda na edição e manutenção de recursos com limites comuns. Com esta opção ativada, o QGIS 'detecta' limites compartilhados. Quando você move vértices / segmentos comuns, o QGIS também os move nas geometrias dos recursos vizinhos.

A edição topológica funciona com recursos de diferentes camadas, desde que as camadas estejam visíveis e no modo de edição.

In layer with Z or M values, topological editing will interpolate the Z or M value of the vertex based on the value of the edge used for the connection.

Overlapping control

Overlapping prevents you from drawing new features that overlap existing ones in the selected layer, speeding up digitizing of adjacent polygons. It can be controlled by the overlap tool. Three modes are available:

1. *Allow Overlap* (default)
2. *Avoid Overlap on Active Layer*: prevents any overlap with other features from the layer being edited. Digitize the new geometries so that they overlap their neighbours and QGIS will cut the overlapping part(s) of the new geometries and snap them to the boundary of the existing features. The advantage is that you don't have to digitize the common vertices on boundary.
3. *Follow Advanced Configuration*: allows the overlapping setting to be set on a layer basis in the *Advanced configuration* view mode.

> **Nota**
>
> Se a nova geometria for totalmente coberta pelas existentes, ela será limpa e o QGIS mostrará uma mensagem de erro.

Aviso

** Use com cuidado a opção ****Evitar sobreposição* ** **

Como essa opção cortará novas geometrias sobrepostas de qualquer camada de polígono, você poderá obter geometrias inesperadas se esquecer de desmarcá-la quando não for mais necessária.

Rastreamento automático

Geralmente, ao usar as ferramentas de captura de mapas (adicionar recurso, adicionar peça, adicionar anel, remodelar e dividir), você precisa clicar em cada vértice do recurso. Com o modo de rastreamento automático, você pode acelerar o processo de digitalização, pois não precisa mais colocar manualmente todos os vértices durante a digitalização:

1. Ative o | rastreio | Tracing (na barra de ferramentas: guilabel:` Snapping`) pressionando o ícone ou pressionando: kbd: tecla *T*.
2. Ajustar para um vértice ou segmento de um recurso que você deseja rastrear.
3. Move the mouse over another vertex or segment you'd like to snap and, instead of the usual straight line, the digitizing rubber band represents a path from the last point you snapped to the current position. The tool also works with curved geometries.

   O QGIS realmente usa a topologia de recursos subjacente para criar o caminho mais curto entre os dois pontos. O rastreamento exige que o snap seja ativado em camadas rastreáveis para criar o caminho. Você também deve encaixar em um vértice ou segmento existente ao digitalizar e garantir que os dois nós sejam topologicamente conectáveis pelas bordas dos recursos existentes; caso contrário, o QGIS não poderá conectá-los e, portanto, traçará uma única linha reta.
4. Clique e QGIS posiciona os vértices intermediários seguindo o caminho exibido.

Desdobre o | rastreio | Ative o rastreio e defina a opção: guilabel:` Offset` para digitalizar um caminho paralelo aos recursos em vez de rastreá-los. Um valor positivo desloca o novo desenho para o lado esquerdo da direção de rastreamento e um valor negativo faz o oposto.

Nota

** Ajuste as configurações de escala ou ajuste do mapa para um rastreamento ideal **

Se houver muitos recursos na exibição do mapa, o rastreamento será desativado para evitar a preparação potencialmente longa da estrutura de rastreamento e a sobrecarga de memória. Após ampliar ou desativar algumas camadas, o rastreamento é ativado novamente.

Nota

** Não adiciona pontos topológicos **

Esta ferramenta não adiciona pontos às geometrias poligonais existentes, mesmo que:guilabel:*Edição topológica* esteja ativada. Se a precisão da geometria estiver ativada na camada editada, a geometria resultante poderá não seguir exatamente uma geometria existente.

Dica

** Ative ou desative rapidamente o rastreio automático pressionando a tecla **: kbd: *T* **

Pressionando a tecla: kbd: *T*, o rastreamento pode ser ativado / desativado a qualquer momento (mesmo durante a digitalização de um recurso), para que seja possível digitalizar partes do recurso com o rastreamento ativado e outras partes com o rastreamento desativado. As ferramentas se comportam como de costume quando o rastreamento está desativado.

Dica

Converter traçado em geometrias curvas

By using *Settings* ► *Options* ► *Digitizing* ► *Tracing* you can create curved geometries while digitizing. See *digitizing options*.

12.3.4 Digitalizar uma camada existente

Por padrão, o QGIS carrega camadas somente leitura. Essa é uma salvaguarda para evitar a edição acidental de uma camada, se houver um deslizamento do mouse. No entanto, você pode optar por editar qualquer camada, desde que o provedor de dados a suporte (consulte:ref:*supported_format*), e a fonte de dados subjacente é gravável (ou seja, seus arquivos não são somente leitura).

Dica

** Restrinja a permissão de edição em camadas dentro de um projeto **

Na tabela:menuselection:*Projeto -> Propriedades … -> Fontes de dados -> Recursos de camadas*, você pode optar por definir qualquer camada somente leitura, independentemente da permissão do provedor. Essa pode ser uma maneira prática, em um ambiente de vários usuários, para evitar que usuários não autorizados editem camadas por engano (por exemplo, Shapefile), portanto, dados potencialmente corrompidos. Observe que essa configuração se aplica apenas ao projeto atual.

Em geral, as ferramentas para editar camadas vetoriais são divididas em uma barra de ferramentas de digitalização e uma avançada, descritas na seção: ref: *sec_advanced_edit*. Você pode selecionar e desmarcar os dois em: menuelection: *View -> Barras de ferramentas ->*.

Usando as ferramentas básicas de digitalização, você pode executar as seguintes funções:

Tabela12.3: Vector layer basic editing toolbar

Ferramenta	Finalidade	Ferramenta	Finalidade
	Acesso para salvar, reverter ou cancelar mudanças em todas as camadas ou em camadas selecionadas simultaneamente		Turn on or off edit status of selected layer(s) based on the active layer status
	Salvar edições na camada ativa		
	Digitize using straight segments		Digitize using curve lines
	Enable freehand digitizing		Digitize polygon of regular shape
	Adicionar novo registro		Adicionar Feição: Ponto de Captura
	Adicionar recurso: Linha de captura		Adicionar recurso: Capturar polígono
	Ferramenta de Vértice (todas as camadas)		Ferramenta Vértice (camada atual)
Show Vertex Editor	Set whether the vertex editor panel should auto-open		Modifique os atributos de todos os recursos selecionados simultaneamente
	Excluir feições selecionadas da camada ativa		Recortar feições da camada ativa
	Copiar as feições selecionadas da camada ativa		Colar feições na camada ativa
	Undo changes in the active layer		Redo changes in active layer

Observe que, ao usar qualquer uma das ferramentas de digitalização, você ainda pode:ref:*ampliar ou deslocar* na tela do mapa sem perder o foco na ferramenta.

All editing sessions start by choosing the Toggle editing option found in the context menu of a given layer, from the attribute table dialog, the digitizing toolbar or the *Layer* menu.

Quando a camada estiver no modo de edição, botões de ferramentas adicionais na barra de ferramentas de edição ficarão disponíveis e os marcadores aparecerão nos vértices de todos os recursos, a menos que:guilabel:*Mostrar marcadores apenas para os recursos selecionados* opção em: menuelection:` Settings -> Opções … -> O menu Digitalizar` está marcado.

Dica

Salvar Regularmente

Remember to Save Layer Edits regularly. This will also check that your data source can accept all the changes.

Geometry editing techniques

When a geometry drawing tool (mainly the ones that add, split, reshape features) is enabled for a line or polygon based layer, you can select the technique for adding new vertices:

- The Digitize with Segment: draws straight segment whose start and end points are defined by left clicks.
- The Digitize with Curve: draws curve line based on three consecutive nodes defined by left clicks (start, point along the arc, end). If the geometry type does not support curves, then consecutive smaller segments are used to approximate the curvature.
- The Stream Digitizing: draws lines in freehand mode, i.e. nodes are added following cursor movement in the map canvas and a *Streaming Tolerance*. The streaming tolerance defines the spacing between consecutive vertices. Currently, the only supported unit is pixels (`px`). Only the starting left click and the ending right click are necessary in this mode.
- The Digitize Shape: triggers tools on the *Shape Digitizing Toolbar* to draw a polygon of a regular shape.

The selected technique remains while switching among the digitizing tools. You can combine any of the first three methods while drawing the same geometry.

Adicionando Elementos

Dependendo do tipo de camada, você pode usar o | newTableRow | Adicionar registro, | capturePoint | Adicionar recurso de ponto, | captureLine | Adicionar recurso de linha ou | capturePolygon | : sup: ícones Adicionar recurso de polígono na barra de ferramentas para adicionar novos recursos à camada atual.

To add a geometryless feature, click on the Add Record button and you can enter attributes in the feature form that opens.

To create features with the spatially enabled tools, you first digitize the geometry then enter its attributes. To digitize the geometry:

1. (Optional as it is the default) Select the Digitize With Segment geometry drawing method
2. Left-click on the map area to create the first point of your new feature. For point features, this should be enough and trigger, if required, the feature form to fill in their attributes.
3. For line or polygon geometries, keep on left-clicking for each additional point you wish to capture. You can rely on the *snapping to features* options, the *snap-to-grid* or the *advanced digitizing* panel to accurately position each vertex.

   Along with drawing straight segments between nodes you click one by one, lines and polygons can be:

   - *traced automatically*, accelerating the digitization. This will create consecutive straight lines between the vertices you place, following existing features.
   - free-hand digitized, pressing `R` or activating Stream Digitizing.
   - drawn as curve, pressing `Ctrl+Shift+G` or activating Digitize with Curve.

   Nota

   While digitizing line or polygon geometries, you can switch back and forth between the geometry drawing methods, allowing you to create features mixing straight segments, free-hand ones and curved parts.

4. Press `Delete` or `Backspace` key to revert the last node(s) you may wrongly add.

5. Quando você terminar de adicionar pontos, clique com o botão direito do mouse em qualquer lugar na área do mapa para confirmar que você terminou de inserir a geometria desse recurso.

> **Dica**
>
> ** Personalize a faixa de borracha digitalizada **
>
> Ao capturar o polígono, o elástico vermelho por padrão pode ocultar recursos ou locais subjacentes que você deseja capturar um ponto. Isso pode ser corrigido configurando uma opacidade mais baixa (ou canal alfa) no menu: guilabel: *Cor de preenchimento* do elástico em: seleção de menu: *Configurações -> Opções -> Digitalização*. Você também pode evitar o uso do elástico marcando:guilabel:*Não atualize o elástico durante a edição do nó*.

6. For line feature pressing `Shift` + right-click will close the line automatically.
7. The attribute window will appear, allowing you to enter the information for the new feature. Fig.12.80 shows setting attributes for a fictitious new river. However, in the *Digitizing* menu under the *Settings* ► *Options* menu, you can also:
   - *Suppress attributes pop-up windows after each created feature* to avoid the form opening;
   - ou | caixa de seleção | *Reutilizar os últimos valores de atributo inseridos* para ter campos automaticamente preenchidos na abertura do formulário e apenas digitar valores alterados.

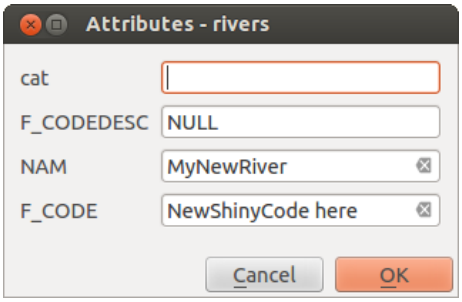


Figura12.80: Caixa de diálogo Inserir valores de atributo após digitalizar um novo recurso de vetor

Ferramenta Vertex

O QGIS fornece duas ferramentas para interagir com os vértices das feições vetoriais:

- Vertex Tool (Current Layer): only overlaid features in the active layer (in the *Layers* panel) are affected
- Vertex Tool (All Layers): any overlaid features in all editable layers are affected. This allows you to edit features without switching the active layer or edit multiple layers at once (e.g., country and their regions boundaries)

For any editable vector layer, the vertex tools provide manipulation capabilities of feature vertices similar to CAD programs. It is possible to select multiple vertices at once and to move, add or delete them altogether. The vertex tools also support the topological editing feature. They are selection persistent, so when some operation is done, selection stays active for this feature and tool.

> **Dica**
>
> **Drawing a series of new vertices**
>
> The vertex tool does not support *automatic tracing* and is optimized for editing individual vertices and moving or deleting an arbitrary selection of multiple vertices. Try using the Reshape Features tool when you need to replace

or insert a series of new vertices, especially if you want to trace other features.

It is important to set the property *Settings* ► *Options* ► *Digitizing* ► *Search Radius:* 1,00 to a number greater than zero. Otherwise, QGIS will not be able to tell which vertex is being edited and will display a warning.

Dica

Marcadores de Vértice

QGIS supports different kinds of vertex markers: 'Semi-transparent circle', 'Cross' and 'None'. To change the marker style, choose *Options* from the *Settings* menu, click on the *Digitizing* tab and select the appropriate entry.

Operações Básicas

Dada uma camada no modo de edição, comece ativando a ferramenta vértice. Círculos vermelhos aparecerão ao passar o mouse sobre os vértices.

- **Selecionando os vértices**: Você pode selecionar os vértices por:
  - Clicking on them one at a time holding `Shift` key pressed
  - Click-and-dragging a rectangle surrounding the target vertices
  - Drawing a polygon surrounding the target vertices: Hold `Alt` and click using the vertex tool to start digitizing a polygon. Each subsequent click adds a new vertex to the rubberband polygon. `Backspace` or `Delete` removes last added rubberband vertex. `Esc` cancels the polygon selection mode, as also does backspacing/deleting all of the rubberband's vertices. Right click finalizes the polygon digitizing and selects all vertices within the rubberband polygon.

  When a vertex is selected, its color changes to blue. To add more vertices to the current selection, hold down the `Shift` key while proceeding as above. To remove vertices from the selection, hold down `Ctrl`.

  Dica

  Feature selection bounds vertex tool

  Vertices can be selected accross different features (or layers). If you are looking for vertices of a specific feature in a crowded place, first select that feature. Then draw the rectangle or polygon selector with the vertex tool around the vertices: only the selected feature's vertices are selected.

  This is also the case if you display the feature in the *vertex editor* panel.

- **Batch vertex selection mode**: The batch selection mode can be activated by pressing `Shift+R`. Select a first node with one single click, and then hover **without clicking** another vertex. This will dynamically select all the nodes in between using the shortest path (for polygons).

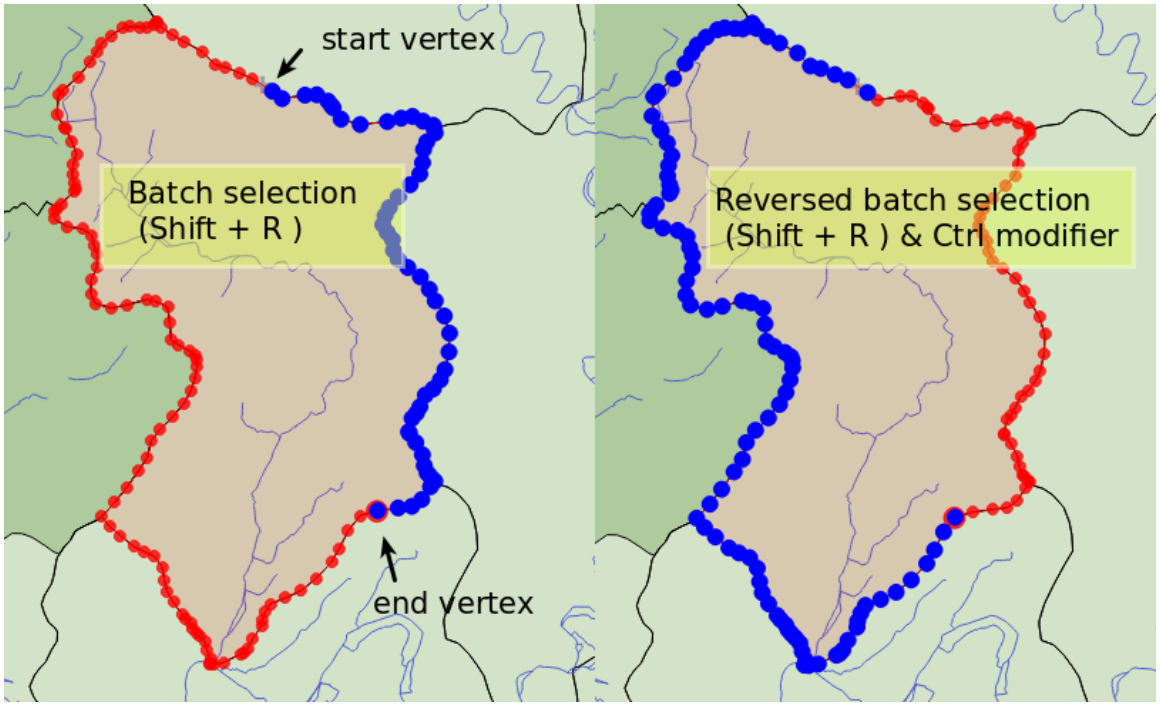


Figura12.81: Batch vertex selection using `Shift+R`

Press `Ctrl` will invert the selection, selecting the longest path along the feature boundary. Ending your node selection with a second click, or pressing `Esc` will escape the batch mode.

- **Adding vertices**: To add a vertex to a line or polygon geometry, hold `Shift` and double-click the place on the segment.

  When hovering a segment, a virtual new node appears on the center. Click on it, move the cursor to a new location and click again to add a new vertex. For lines, a virtual node is also proposed at both extremities: click on it, do subsequent clicks and finish with a right-click; this allows to easily extend an existing line.

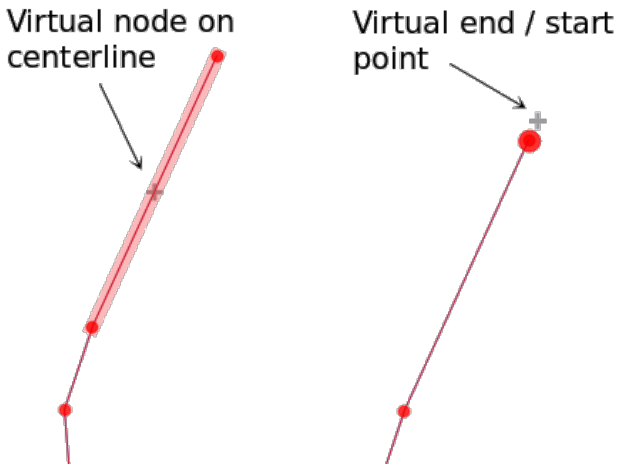


Figura12.82: Nós virtuais para adicionar vértices

- ** Excluindo vértices **: selecione os vértices e clique na tecla: kbd: *Excluir*. A exclusão de todos os vértices de um recurso gera, se compatível com a fonte de dados, um recurso sem geometria. Observe que isso não exclui o recurso completo, apenas a parte da geometria. Para excluir um recurso completo, use o | deleteSelectedFeatures | :sup:ferramenta *Excluir selecionados*.

- **Moving vertices**: Select all the vertices you want to move, click on a selected vertex or edge, and click on the desired new location. You can use the *snapping to feature capabilities* and the *Advanced Digitizing Panel* constraints for distance, angles, exact X and Y location before the second click. All the selected vertices will be translated.

  However, if the *snap-to-grid* option is enabled, selected vertices are snapped to the closest grid intersection to their translated position. Unselected vertices are also moved to their closest grid intersection. There is no simple translation.

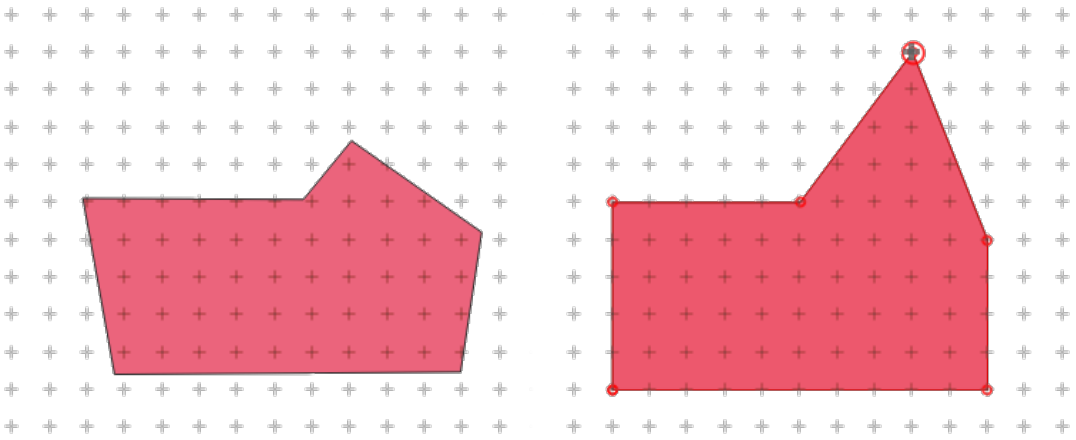


Figura12.83: Movendo o vértice superior encaixa todos os vértices na grade

- **Converting adjacent segments to/from curve**: Select the center vertex of the segment you want to convert, hit the O letter key. If the vertex was in a curve, the curve is converted into straight lines. If the vertex was between two straight lines, they are converted into a curve. A first or a last vertex of a line can't be converted to a center vertex curve. The layer must be compatible with curve geometry type.

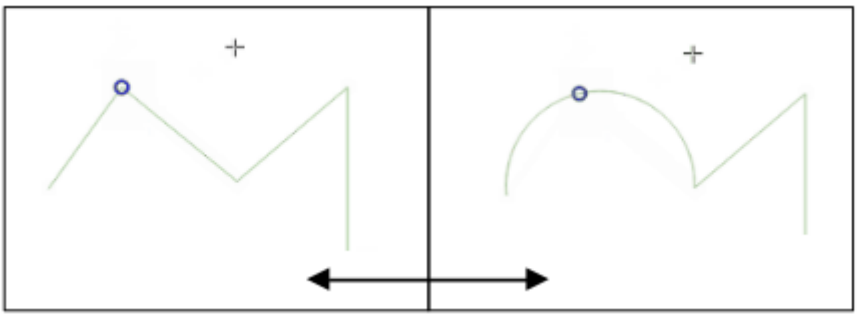


Figura12.84: Switch from curve to straight lines with O letter

Each change made with the vertex tool is stored as a separate entry in the *Undo* dialog. Remember that all operations support topological editing when this is turned on. On-the-fly projection is also supported.

O painel do editor de vértices

With enabling a vertex tool, you also open the *Vertex Editor* panel. Right-clicking over a feature fills the panel with the list of all the vertices of the feature with their *x*, *y* (*z*, *m* if applicable) coordinates and *r* (for the radius, in case of circular geometry). The feature is also made exclusive for editing, meaning that the edit of any other features is disabled:

- Selecting a row in the table does select the corresponding vertex in the map canvas, and vice versa.
- Clicking or dragging over the map canvas will only select or move vertices and segments of that feature
- Change a coordinate in the table and the vertex position is updated. This is a convenient way to edit Z coordinate or M value on vertices.
- You can also select multiple rows and delete them altogether.
- New vertices can only be added to the bound feature

If you do not want the *Vertex Editor* panel to immediately show each time you interact with vertex tools (and potentially hide other panels or disturb panels placement), uncheck the *Auto-open table* entry in the ☰ Options menu at the top of the panel. You can then also close the panel. To reopen the panel, you would need to right-click over a panel or toolbar and select it in the list or tick the *Show vertex editor* entry in the *Digitizing toolbar*.

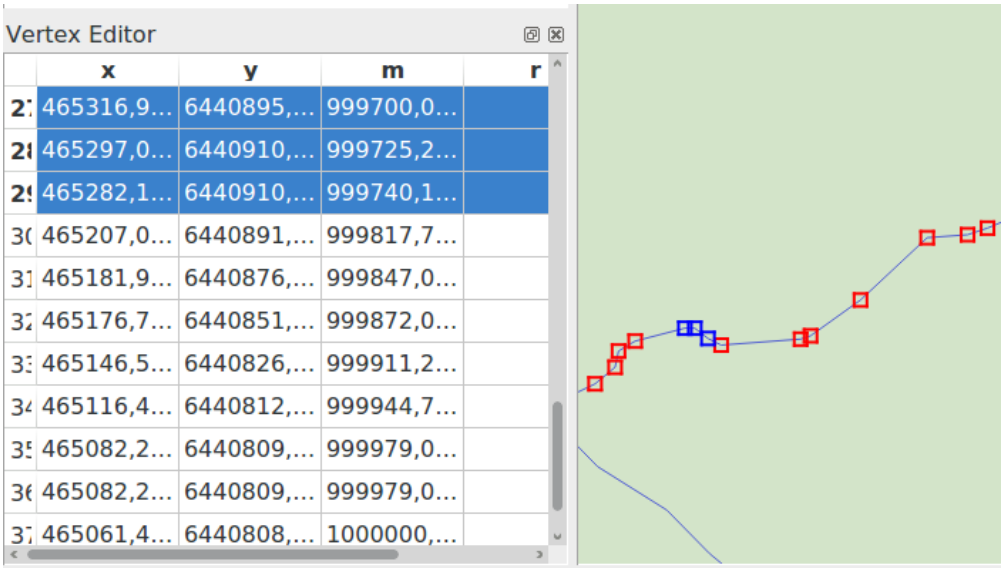


Figura12.85: Painel do editor de vértices mostrando os nós selecionados

Rules of Z coordinate or M value assignment

Digitizing 3D vector features or features with M value is not that different from (X,Y) 2D layers'. Tools and options described in this chapter are still available and help you place the vertex or point in a planar environment. Then you may need to handle the Z coordinate (or M value) assignment:

- By default, QGIS will assign to new vertices the *Default Z value* (respectively *Default M value*) set in the *Settings* ► *Options* ► *Digitizing* tab. If the *Advanced Digitizing Panel* is in use, then the value is taken from its *z* (respectively *m*) widget.
- When snapping to a vertex, the new or moved vertex takes the snapped one's Z or M value.
- When snapping to a segment while the topological editing is on, then the new vertex Z or M value is interpolated along the segment.

- If the *z* (respectively *m*) widget of the *Advanced Digitizing Panel* is locked, then its value is applied to the vertex, taking precedence over any snapped vertex or segment Z or M value.

To edit Z or M values of an existing feature, you can use the *Vertex editor panel*. To create features with custom Z or M values you may want to rely on the *Advanced Digitizing Panel*.

Cortando, Copiando e Colando Elementos

Os recursos selecionados podem ser recortados, copiados e colados entre as camadas no mesmo projeto QGIS, desde que as camadas de destino estejam definidas como | toggleEditing | Alterna a edição de antemão.

Dica

** Transforme o polígono em linha e vice-versa usando copiar / colar **

Copie um recurso de linha e cole-o em uma camada de polígono: O QGIS cola na camada de destino um polígono cujo limite corresponde à geometria fechada do recurso de linha. Essa é uma maneira rápida de gerar geometrias diferentes dos mesmos dados.

Features can also be pasted to external applications as text. That is, the features are represented in CSV format, with the geometry data appearing in the OGC Well-Known Text (WKT) format. WKT and GeoJSON features from outside QGIS can also be pasted to a layer within QGIS.

When would the copy and paste function come in handy? Well, it turns out that you can edit more than one layer at a time and copy/paste features between layers. Why would we want to do this? Say we need to do some work on a new layer but only need one or two lakes, not the 5,000 on our `big_lakes` layer. We can create a new layer and use copy/paste to plop the needed lakes into it.

Como exemplo, vamos copiar alguns lagos para uma nova camada:

1. Carregue a camada que quer copiar a partir (camada de origem)
2. Carregue ou crie a camada que quer copiar para (camada de destino)
3. Começar a editar a camada de destino
4. Ative a camada de origem clicando nela na legenda
5. Use the Select Features by area or single click tool to select the feature(s) on the source layer
6. Click on the Copy Features tool
7. Ative a camada de destino clicando na legenda
8. Clique no | editar Colar | :sup:ferramenta *Colar recursos*
9. Parar a edição e salvar as alterações

What happens if the source and target layers have different schemas (field names and types are not the same)? QGIS populates what matches and ignores the rest. If you don't care about the attributes being copied to the target layer, it doesn't matter how you design the fields and data types. If you want to make sure everything - the feature and its attributes - gets copied, make sure the schemas match.

When you paste features into another layer, QGIS checks the attribute values against the *constraints* defined in the destination layer (for example, `NOT NULL`, `Unique`, or expression constraints). If one or more pasted features contain invalid attribute values that do not meet these constraints, a dialog will appear, listing all affected features and fields. When pasting a single feature, you can decide whether to:

- Edit the invalid values directly in the dialog. The corrected values will be used when pasting.
- *Paste Anyway*: the feature is pasted as-is, even with invalid values.

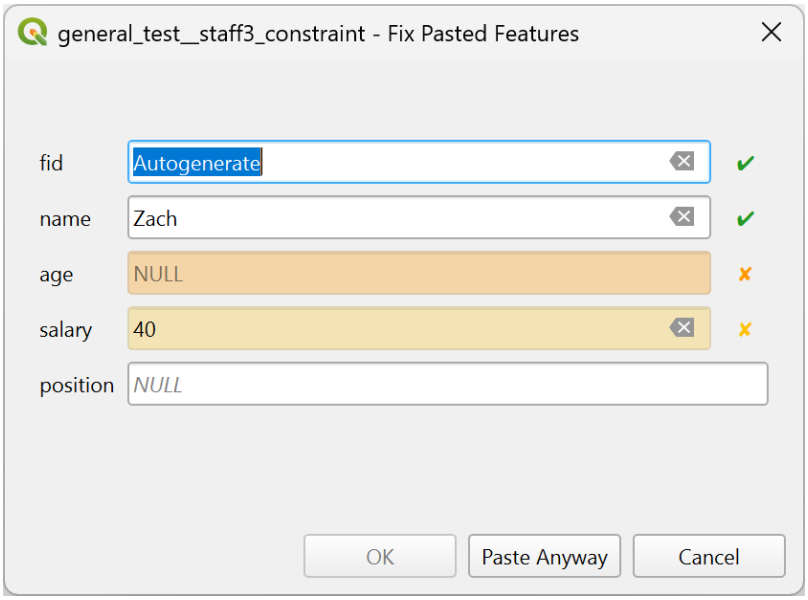


Figura12.86: Dialog for handling invalid attribute values when pasting a single feature

When pasting multiple features, the dialog lists all invalid fields across all features, allowing bulk review and correction. You can decide per field or per feature whether to:

- Edit the invalid values directly in the dialog. The corrected values will be used when pasting.
- *Discard All*: no features pasted.
- *Discard All Invalid*: pasted only features with valid attribute values.
- *Paste All (Including Invalid)*: all features are pasted as-is, even with invalid values.
- *Skip*: skip the specific feature, allowing proceeding with the rest.

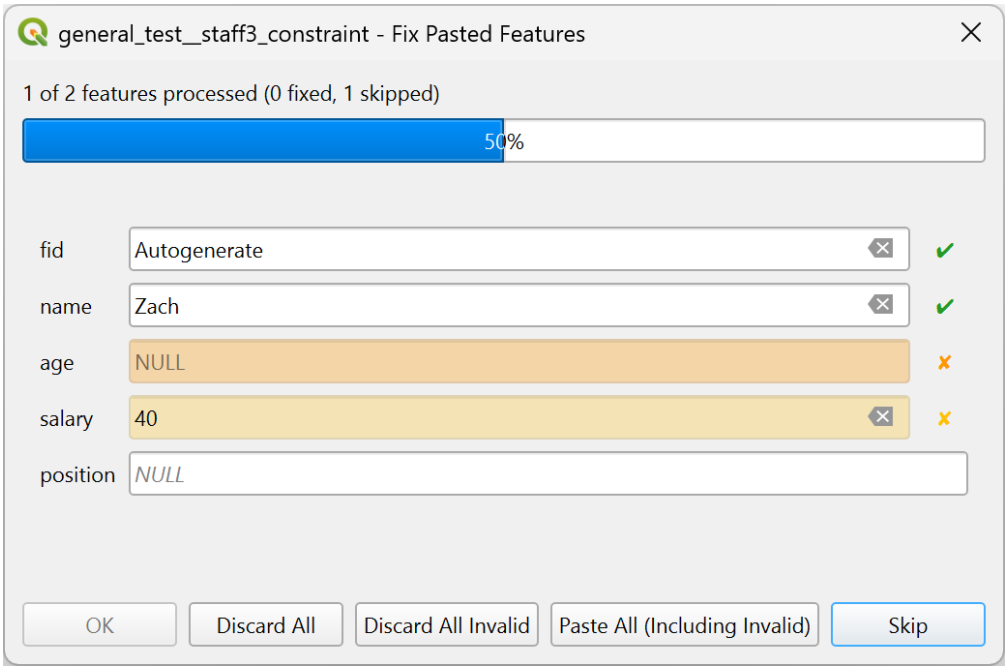


Figura12.87: Dialog for handling invalid attribute values when pasting multiple features

Nota

Congruência dos Elementos Colados

If your source and destination layers use the same projection, then the pasted features will have geometry identical to the source layer. However, if the destination layer is a different projection, then QGIS cannot guarantee the geometry is identical. This is simply because there are small rounding-off errors involved when converting between projections.

Dica

** Copiar atributo da string para outro **

If you have created a new column in your attribute table with type 'string' and want to paste values from another attribute column that has a greater length the length of the column size will be extended to the same amount. This is because the GDAL Shapefile driver knows to auto-extend string and integer fields to dynamically accommodate for the length of the data to be inserted.

Apagando os Elementos Selecionados

Se quisermos excluir um recurso inteiro (atributo e geometria), podemos fazer isso selecionando primeiro a geometria usando o regular | select Rectangle | Selecione Recursos por área ou clique único. A seleção também pode ser feita na tabela de atributos. Depois de definir a seleção, pressione: kbd: *Excluir* ou: kbd: *Backspace* ou use a tecla | delete Features Selecionadas | :sup:ferramenta *Delete Selected* para excluir os recursos. Vários recursos selecionados podem ser excluídos de uma só vez.

The Cut Features tool on the digitizing toolbar can also be used to delete features. This effectively deletes the feature but also places it on a "spatial clipboard". So, we cut the feature to delete. We could then use the Paste Features tool to put it back, giving us a one-level undo capability. Cut, copy, and paste work on the currently selected features, meaning we can operate on more than one at a time.

Retroceder e Retomar

The Undo and Redo tools allows you to undo or redo vector editing operations. There is also a dockable widget, which shows all operations in the undo/redo history (see Fig.12.88). This widget is not displayed by default; it can be displayed by right-clicking on the toolbar and activating the *Undo/Redo Panel* checkbox. The Undo/Redo capability is however active, even if the widget is not displayed.

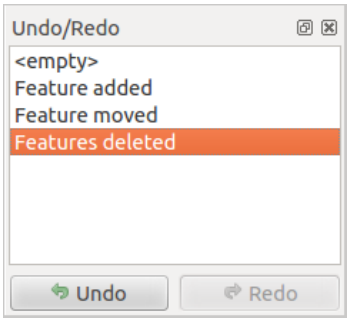


Figura12.88: Refazer e desfazer etapas de digitalização

When Undo is hit or `Ctrl+Z` (or `Cmd+Z`) pressed, the state of all features and attributes are reverted to the state before the reverted operation happened. Changes other than normal vector editing operations (for example, changes done by a plugin) may or may not be reverted, depending on how the changes were performed.

Para usar a tela histórico de desfazer / refazer, basta clicar para selecionar uma operação na lista de histórico. Todas as feições serão revertidas para o estado em que estavam depois da operação selecionada.

Salvando as Camadas Editadas

When a layer is in editing mode, any changes remain in the memory of QGIS. Therefore, they are not committed/saved immediately to the data source or disk. If you want to save edits to the current layer but want to continue editing without leaving the editing mode, you can click the Save Layer Edits button. When you turn editing mode off with Toggle editing (or quit QGIS for that matter), you are also asked if you want to save your changes or discard them.

Se as alterações não puderem ser salvas (por exemplo, disco cheio ou os atributos tiverem valores fora do intervalo), o estado na memória do QGIS será preservado. Isso permite que você ajuste suas edições e tente novamente.

> **Dica**
>
> **Integridade dos dados**
>
> É sempre uma boa idéia fazer backup da fonte de dados antes de começar a editar. Embora os autores do QGIS tenham se esforçado para preservar a integridade dos seus dados, não oferecemos garantia a esse respeito.

Salvando várias camadas ao mesmo tempo

Esse recurso permite a digitalização de várias camadas. Escolha | arquivo Salvar como | *Salvar para camadas selecionadas* para salvar todas as alterações feitas em várias camadas. Você também tem a oportunidade de | reverter edições | *Reversão para camadas selecionadas*, para que a digitalização possa ser retirada para todas as camadas selecionadas. Se você deseja parar de editar as camadas selecionadas, cancele Edições | *Cancelar para a (s) camada (s) selecionada (s)* é uma maneira fácil.

As mesmas funções estão disponíveis para a edição de todas as camadas do projeto.

> **Dica**
>
> ** Use o grupo de transações para editar, salvar ou reverter várias alterações de camada ao mesmo tempo **
>
> When working with layers from the same PostGreSQL database, activate the *Automatically create transaction groups where possible* option in *Project* ► *Properties...* ► *Data Sources* to sync their behavior (enter or exit the edit mode, save or rollback changes at the same time).

12.3.5 Digitalização Avançada

Tabela12.4: Vector layer advanced editing toolbar

Ícone	Finalidade	Ícone	Finalidade
	Ativar ferramentas avançadas de digitalização		
	Mover feições		Copiar e Mover Feição(ões)
	Rodar Elemento(s)		Simplificar elemento
	Scale Feature		
	Adicionar Anel		Adicionar Parte
	Preenchimento Anel		Swap direction
	Apagar Anel		Apagar Parte
	Curva de Afastamento		Refazer elementos
	Dividindo partes		Dividir Elementos
	Juntar Atributos dos Elementos Selecionados		Juntar Elementos Selecionados
	Rodar Símbolos de Pontos		Offset Point Symbols
	Trim or Extend Feature		

Mover feições

The Move Feature(s) tool allows you to move existing features:

1. Selecionar as feições a serem movidas.
2. Click on the map canvas to indicate the origin point of the displacement; you can rely on snapping capabilities to select an accurate point.

   You can also take advantages of the *advanced digitizing constraints* to accurately set the origin point coordinates. In that case:

   1. Primeiro clique no | cad | para ativar o painel.
   2. Digite `` x`` e insira o valor correspondente ao ponto de origem que você deseja usar. Em seguida, pressione o botão | bloqueado | ao lado da opção para bloquear o valor.
   3. Faça o mesmo para a coordenada `y`.
   4. Click on the map canvas and your origin point is placed at the indicated coordinates.
3. Move over the map canvas to indicate the destination point of the displacement, still using snapping mode or, as above, use the advanced digitizing panel which would provide complementary `distance` and `angle` placement constraints to place the end point of the translation.
4. Click on the map canvas: the whole features are moved to new location.

Likewise, you can create a translated copy of the feature(s) using the Copy and Move Feature(s) tool.

Nota

If no feature is selected when you first click on the map canvas with any of the *Move Feature(s)* or *Copy and Move Feature(s)* tools, then only the feature under the mouse is affected by the action. So, if you want to move several features, they should be selected first.

Rodar Elemento(s)

Use the Rotate Feature(s) tool to rotate one or multiple features in the map canvas:

1. Press the Rotate Feature(s) icon
2. Then click on the feature to rotate. The feature's centroid is referenced as rotation center, a preview of the rotated feature is displayed and a widget opens showing the current *Rotation* angle.
3. Click on the map canvas when you are satisfied with the new placement or manually enter the rotation angle in the text box. You can also use the *Snap to* ° box to constrain the rotation values.
4. If you want to rotate several features at once, they shall be selected first, and the rotation is by default around the centroid of their combined geometries.

You can also use an anchor point different from the default feature centroid: press the `Ctrl` button, click on the map canvas and that point will be used as the new rotation center.

If you hold `Shift` before clicking on the map, the rotation will be done in 45 degree steps, which can be modified afterwards in the user input widget.

To abort feature rotation, press the `ESC` button or click on the Rotate Feature(s) icon.

Scale Feature

The Scale Feature tool is similar to the Rotate feature. Though instead of performing a rotation of selected features, it rescales their geometry. The change is performed in relation to the anchor point and the scale ratio can be manually specified in the widget that appears in the upper corner of the canvas.

Simplificar elemento

The Simplify Feature tool allows you to interactively reshape a line or polygon geometry by reducing or densifying the number of vertices, as long as the geometry remains valid:

1. Select the Simplify Feature tool.
2. Click on the feature or drag a rectangle over the features.
3. A dialog pops up allowing you to define the *Method* to apply, ie whether you would like to:
   - *simplify the geometry*, meaning less vertices than the original. Available methods are `Simplify by distance`, `Simplify by snapping to grid` or `simplify by area (Visvalingam)`. You'd then need to indicate the value of *Tolerance* in `Layer units`, `Pixels` or `map units` to use for simplification. The higher the tolerance is the more vertices can be deleted.
   - or *densify the geometries* with new vertices thanks to the `Smooth` option: for each existing vertex, two vertices are placed on each of the segments originated from it, at an *Offset* distance representing the percentage of the segment length. You can also set the number of *Iterations* the placement would be processed: the more iterations, the more vertices and smoother is the feature.

   Settings that you used will be saved when leaving a project or an edit session. So you can go back to the same parameters the next time you simplify a feature.

4. A summary of the modifications that would apply is shown at the bottom of the dialog, listing number of features and number of vertices (before and after the operation and the ratio the change represents). Also, in the map canvas, the expected geometry is displayed over the existing one, using the rubberband color.
5. When the expected geometry fits your needs, click *OK* to apply the modification. Otherwise, to abort the operation, you can either press *Cancel* or right-click in the map canvas.

> **Nota**
>
> Unlike the feature simplification option in *Settings* ► *Options* ► *Rendering* menu which simplifies the geometry just for rendering, the Simplify Feature tool permanently modifies feature's geometry in data source.

Adicionar Parte

You can Add Part to a selected feature generating a multipoint, multiline or multipolygon feature. The new part must be digitized outside the existing one which should be selected beforehand.

The Add Part can also be used to add a geometry to a geometryless feature. First, select the feature in the attribute table and digitize the new geometry with the Add Part tool.

> **Nota**
>
> **Order of vertices in polygon parts**
>
> Unlike the OGC standards, QGIS doesn't constrain vertices of the exterior boundary of a polygon feature to be ordered counterclockwise. Thus, you can find both directions in a layer. However, every parts of the same multipolygon feature will have their outer vertices ordered following the same direction.
>
> You can however use the *Regra de força da mão direita.* algorithm to constrain features of a layer to have vertices of their outer boundaries ordered in the clockwise direction.

Apagar Parte

The Delete Part tool allows you to delete parts from multifeatures (e.g., to delete polygons from a multi-polygon feature). This tool works with all multi-part geometries: point, line and polygon. Furthermore, it can be used to totally remove the geometric component of a feature. To delete a part, simply click within the target part.

Adicionar Anel

You can create ring polygons using the Add Ring icon in the toolbar. This means that inside an existing area, it is possible to digitize further polygons that will occur as a 'hole', so only the area between the boundaries of the outer and inner polygons remains as a ring polygon.

To add a ring:

1. Select the feature(s) to modify
2. Activate the Add Ring tool
3. Draw a polygon within the selected geometries, using the aforementioned *techniques*. A hole appears in the selected geometries.

4. If no geometry is selected when the ring is drawn, then a hole is added to each of the polygons the ring is drawn over.

> **Nota**
>
> **Order of vertices in polygon rings**
>
> Unlike the OGC standards, QGIS doesn't constrain vertices of the exterior boundary of a polygon feature to be ordered counterclockwise. Thus, you can find both directions in a layer. However, every rings of the same (multi)polygon feature will have their vertices ordered in the opposite direction to the outer boundary's.
>
> You can however use the *Regra de força da mão direita.* algorithm to constrain features of a layer to have vertices of their outer boundaries ordered in the clockwise direction, and vertices of their interior rings ordered in the counter-clockwise direction.

Preenchimento Anel

The Fill Ring tool helps you create polygon feature that totally falls within another one without any overlapping area; that is the new feature covers a hole within the existing one. To create such a feature:

1. Select the Fill Ring tool.
2. Desenhe um novo polígono sobre o recurso existente: o QGIS adiciona um anel à sua geometria (como se você usasse a ferramenta | addRing |Add Ring) e cria um novo recurso cuja geometria corresponde ao anel (como se você:ref:*traçado* sobre os limites internos com a ferramenta | capturePolygon |Adicionar recurso de polígono).
3. Or alternatively, if the ring already exists on the feature, place the mouse over the ring and left-click while pressing `Shift`: a new feature filling the hole is drawn at that place.

   The *Feature Attributes* form of the new feature opens, pre-filled with values of the "parent" feature and/or *fields constraints*.

Apagar Anel

The Delete Ring tool allows you to delete rings within an existing polygon, by clicking inside the hole. This tool only works with polygon and multi-polygon features. It doesn't change anything when it is used on the outer ring of the polygon.

Refazer elementos

You can reshape line and polygon features using the Reshape Features tool on the toolbar. For lines, it replaces the line part from the first to the last intersection with the original line.

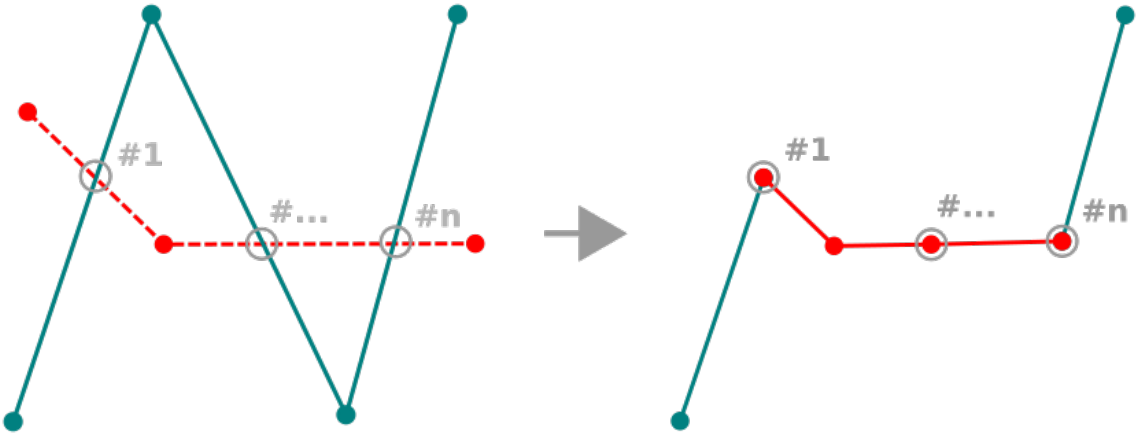


Figura12.89: Remodelar linha

Dica

Extend linestring geometries with reshape tool

Use the Reshape Features tool to extend existing linestring geometries: snap to the first or last vertex of the line and draw a new one. Validate and the feature's geometry becomes the combination of the two lines.

For polygons, it will reshape the polygon's boundary. For it to work, the reshape tool's line must cross the polygon's boundary at least twice. To draw the line, click on the map canvas to add vertexes. To finish it, just right-click. Like with the lines, only the segment between the first and the last intersections is considered. The reshape line's segments that are inside the polygon will result in cropping it, where the ones outside the polygon will extend it.

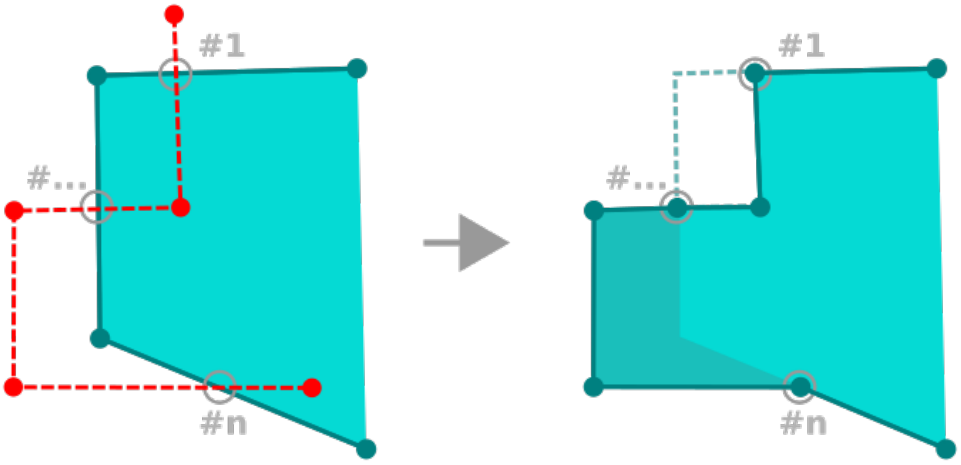


Figura12.90: Remodelar polígono

With polygons, reshaping can sometimes lead to unintended results. It is mainly useful to replace smaller parts of a polygon, not for major overhauls, and the reshape line is not allowed to cross several polygon rings, as this would generate an invalid polygon.

Nota

A ferramenta de mudança de forma pode alterar a posição inicial de um anel ou de uma linha poligonal fechada. Assim, o ponto que está representada 'duas vezes' não será o mesmo mais. Isto pode não ser um problema para a maioria das aplicações, mas é algo a considerar.

Curvas de Afastamento

The Offset Curve tool creates parallel shifts of line layers. The tool can be applied to the edited layer (the geometries are modified) or also to background layers (in which case it creates copies of the lines / rings and adds them to the edited layer). It is thus ideally suited for the creation of distance line layers. The *User Input* dialog pops-up, showing the displacement distance and other settings.

To create a shift of a line or polygon layer, you must first go into editing mode and activate the Offset Curve tool. Then click on a feature to shift it. Move the mouse and click where wanted or enter the desired distance in the user input widget. Holding `Ctrl` during the 2nd click will make an offset copy. Your changes may then be saved with the Save Layer Edits tool.

For geometries on background layers make sure that snapping is on and hold `Ctrl` to select the geometry from the background. Also hold `Ctrl` when doing the second click. Geometries will be converted to the target layer geometry type.

QGIS options dialog (Digitizing tab then **Curve offset tools** section) or the icon in the *User Input* dialog allows you to configure *some parameters* like **Join style**, **Quadrant segments**, **Miter limit** and **End cap style**.

Linha Invertida

Changing the direction of a line geometry can be useful for cartographical purposes (e.g., for orienting symbols or labels that follow the line direction) or when preparing for network analysis where the direction of flow or travel matters.

This tool works only on line layers. For multipart lines, you can reverse individual parts without affecting the other parts of the feature.

Para alterar a direção de uma linha:

1. Select a line layer in the *Layers* panel and make it editable.
2. Activate the reverse line tool by clicking Reverse line in the *Advanced Digitizing* toolbar.
3. Move the pointer over a line feature. The part under the cursor is highlighted.
4. Click on the line: the direction of the line (or part) is reversed.
5. You can reverse other lines by clicking on them.

Dica

For bulk line direction changes, use the *Direção reversa da linha* processing algorithm which can reverse all lines in a layer or a selection at once.

Dividir Elementos

Use the Split Features tool to split a feature into two or more new and independent features, ie. each geometry corresponding to a new row in the attribute table.

Para quebrar feições de linha ou polígono:

1. Select the Split Features tool.
2. Draw a line across the feature(s) you want to split. If a selection is active, only selected features are split. The original feature is then assigned the biggest geometry resulting from the splitting, and new features are created for the remaining parts. Fields of the features are filled/updated according to the datasource provider rules or their *splitting policy*.
3. You can then as usual modify any of the attributes of any resulting feature.

> **Dica**
>
> **Split a polyline into new features in one-click**
>
> Using the Split Features tool, snap and click on an existing vertex of a polyline feature to split that feature into two new features.

Dividindo partes

No QGIS, é possível dividir as partes de um recurso de várias partes para aumentar o número de partes. Basta desenhar uma linha na parte que você deseja dividir usando o | peças separadas | :sup:ícone *Split Parts*.

> **Dica**
>
> ** Divida uma polilinha em novas peças com um clique **
>
> Using the Split Parts tool, snap and click on an existing vertex of a polyline feature to split the feature into two new polylines belonging to the same feature.

Juntar elementos selecionados

The Merge Selected Features tool allows you to create a new feature by merging existing ones: their geometries are merged to generate a new one. If features don't have common boundaries, a multipolygon/multipolyline/multipoint feature is created.

1. Primeiro, selecione os recursos que você deseja combinar.
2. Em seguida, pressione o | mergeFeatures | :sup:botão "Mesclar recursos selecionados".
3. In the new dialog, the *Merge* line at the bottom of the table shows the attributes of the resulting feature. If any *Merge policy* have been defined, they are automatically applied to populate this row. However, you can override these values manually. Modify the values in the following ways:
   - substituindo manualmente o valor na célula correspondente;
   - selecting a row in the table and pressing *Take attributes from selected feature* to use the values of this initial feature;
   - pressing the *Take attributes from the largest geometry* to use the attributes from the longest line feature, the largest polygon, or the multipoints with the most parts;

- pressing *Skip all fields* to use empty attributes;
- expanding the drop down menu at the top of the table, select any of the above options to apply to the corresponding field only. There, you can also choose to aggregate the initial features attributes (Minimum, Maximum, Median, Sum, Count, Concatenation… depending on the type of the field. see *Statistical Summary Panel* for the full list of functions).

> **Nota**
>
> If the layer has default values or clauses present on fields, these are used as the initial value for the merged feature.

4. Press *OK* to apply the modifications. A single (multi)feature is created in the layer, replacing the previously selected ones.

Juntar os atributos dos elementos selecionados

The Merge Attributes of Selected Features tool allows you to apply same attributes to features without merging their boundaries. The dialog is the same as the `Merge Selected Features` tool's except that unlike that tool, selected objects are kept with their geometry while some of their attributes are made identical.

Rodar Símbolos de Pontos

The Rotate Point Symbols allows you to individually change the rotation of point symbols in the map canvas.

1. First, you need to indicate the field to store the rotation value in. This is made by assigning a field to the symbol *data-defined* rotation property:
   1. In the *Layer Properties* ► *Symbology* dialog, browse to the symbol editor dialog.
   2. Click the *Data-defined override* widget near the *Rotation* option of the top *Marker* level (preferably) of the symbol layers.
   3. Choose a field in the *Field Type* combobox. Values of this field are hence used to rotate each feature's symbol accordingly.

      You can also check the Store data in project entry to generate an *auxiliary data storage* field to control the rotation value.

> **Nota**
>
> **Certifique-se de que o mesmo campo seja atribuído a todas as camadas de símbolos**
>
> A configuração do campo de rotação definido por dados no nível mais alto da árvore de símbolos o propagará automaticamente para todas as camadas de símbolos, um pré-requisito para executar a rotação gráfica de símbolos com a ferramenta:guilabel:*Rotate Point Symbols*. De fato, se uma camada de símbolo não tiver o mesmo campo anexado à sua propriedade de rotação, a ferramenta não funcionará.

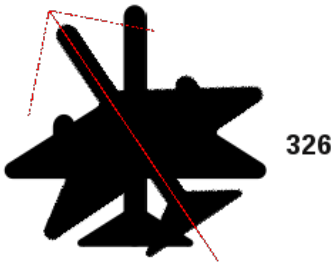


Figura12.91: Girando um símbolo de ponto

2. Then click on a point symbol in the map canvas with the Rotate Point Symbols tool
3. Move the mouse around. A red arrow with the rotation value will be visualized (see Fig.12.91). If you hold the `Ctrl` key while moving, the rotation will be done in 15 degree steps.
4. Quando você obtiver o valor do ângulo esperado, clique novamente. O símbolo é renderizado com esta nova rotação e o campo associado é atualizado de acordo.

   Você pode clicar com o botão direito do mouse para abortar a rotação do símbolo.

Offset Point Symbols

The Offset Point Symbols allows you to interactively change the rendered position of point symbols in the map canvas. This tool behaves like the Rotate Point Symbols tool except that it requires you to connect a field to the data-defined *Offset (X,Y)* property of each layer of the symbol. The field will then be populated with the offset coordinates for the features whose symbol is moved in the map canvas.

1. Associate a field to the data-defined widget of the *Offset (X,Y)* property of the symbol. If the symbol is made with many layers, you may want to assign the field to each of them
2. Select the Offset Point Symbols tool
3. Clique em um símbolo de ponto
4. Mover para um novo local
5. Clique novamente. O símbolo é movido para o novo local. Os valores de deslocamento da posição original são armazenados no campo vinculado.

   You can right-click to abort symbol offset.

Nota

The Offset Point Symbols tool doesn't move the point feature itself; you should use the Vertex Tool (Current Layer) or Move Feature tool for this purpose.

Recortar / estender o recurso

The Trim/Extend tool allows you to shorten or lengthen segments of a (multi)line or (multi)polygon geometry to converge with a selected segment (the cutting line). This results in a modified geometry with a vertex snapped to a reference segment or in its prolongation. Depending on how the selected geometries are placed in relation to each other, the tool will either:

- **Aparar**: remove partes do segmento de linha ou limite de polígono, além da linha de corte
- **Estender**: estende os limites dos polígonos ou segmentos de linha para que eles possam se encaixar na linha de corte.

A fim de aparar ou ampliar as geometrias existentes:

1. Select the Trim/Extend tool
2. Click the reference segment, i.e., the segment with respect to which you want to extend or trim another segment. It doesn't need to be on the active layer. A red dotted line appears, visually extending the reference segment across the map canvas according to the active layer's CRS.
3. Hover over the target segment, i.e., the one you want to trim or extend. It does not need to be the last segment of the geometry, but has to be on the active layer. QGIS displays a preview of the feature's geometry with the segment limited to its intersection with the highlighted red dotted extending line.
    1. If an extension of the segment is required, click anywhere on the segment.
    2. In the case of a trim, click the part that should remain.
    3. In either case, the feature's geometry is updated accordingly. When both segments are in 3D, the tool performs an interpolation on the limit segment to get the Z value.
4. If you need to use the same reference segment for trimming or extending many features:
    1. Press `Shift` while selecting the reference segment.
    2. Click consecutively on the segments to modify and each will be trimmed or extended accordingly.

Nota

Snapping is automatically enabled when this tool is activated. Your original snapping settings will be restored once the tool is deactivated.

Atenção

Pay attention to the modified geometry while using the Trim/Extend tool. Depending on the inputs, it can create invalid geometries, potentially resulting in failure at layer saving.

12.3.6 Digitalização de forma

The *Shape Digitizing* toolbar offers a set of tools to draw lines or polygons features of regular shape. It is synchronized with the Digitize Shape *geometry drawing method* you can select on the *Digitizing Toolbar*. To use it:

1. Display the toolbar: *View* ► *Toolbars* ► *Shape Digitizing*
2. Select a tool that creates or modifies the shape of a geometry, e.g. Add line feature, Add polygon feature, Add part, Add ring, Reshape Features, …

3. The Digitize with segment button on the *Digitizing Toolbar* is enabled. The first time, you may need to switch it to the Digitize Shape in order to enable tools on the *Shape Digitizing* toolbar.
4. Pick a shape digitizing tool and draw.

Circular string by radius

The Circular string by radius button allows to add line or polygon features with a circular geometry, given two nodes on the curve and a radius:

1. Left click twice to place the two points on the geometry.
2. A *Radius* widget in the top right corner of the map canvas displays current radius (corresponding to distance between the points). Edit that field to the value you want.
3. An overview of the arcs matching these constraints is displayed while moving around the cursor. Right-click to validate when the expected arc is shown.
4. Add a new point to start shaping another arc.

Nota

Curved geometries are stored as such only in compatible data provider

Although QGIS allows to digitize curved geometries within any editable data format, you need to be using a data provider (e.g. PostgreSQL, Oracle Spatial, memory layer, GML or WFS) that supports curves to have features stored as curved, otherwise QGIS segmentizes the circular arcs.

Desenhar círculos

Há um conjunto de ferramentas para desenhar círculos. As ferramentas são descritas abaixo.

Circles are converted into circular strings. Therefore, as explained in *Circular string by radius*, if allowed by the data provider, it will be saved as a curved geometry, if not, QGIS will segmentize the circular arcs.

- Circle from 2 points: The two points define the diameter and the orientation of the circle. (Left-click, right-click)
- Circle from 3 points: Draws a circle from three known points on the circle. (Left-click, left-click, right-click)
- Circle by a center point and another point: Draws a circle with a given center and a point on the circle (Left-click, right-click). When used with the *O painel Digitalização avançada* this tool can become a “Add circle from center and radius” tool by setting and locking the distance value after first click.
- Circle from 3 tangents: Draws a circle that is tangential to three segments. **Note that you must activate snapping to segments** (See *Setting the snapping tolerance and search radius*). Click on a segment to add a tangent. If two tangents are parallel, the coordinates of the click on the first parallel tangent are used to determine the positioning of the circle. If three tangents are parallel, an error message appears and the input is cleared. (Left-click, left-click, right-click)
- Circle from 2 tangents and a point: Similar to circle from 3 tangents, except that you have to select two tangents, enter a radius and select the desired center.

Desenhar Elipses

There is a set of tools for drawing ellipses. The tools are described below.

Ellipses cannot be converted as circular strings, so they will always be segmented.

- Ellipse from center and two points: Draws an ellipse with a given center, major axis and minor axis. (Left-click, left-click, right-click)
- Ellipse from center and a point: Draws an ellipse into a bounding box with the center and a corner. (Left-click, right-click)
- Ellipse from extent: Draws an ellipse into a bounding box with two opposite corners. (Left-click, right-click)
- Ellipse from foci: Draws an ellipse by 2 points for foci and a point on the ellipse. (Left-click, left-click, right-click)

Desenhar Retângulos

There is a set of tools for drawing rectangles. The tools are described below.

- Rectangle from center and a point: Draws a rectangle from the center and a corner. (Left-click, right-click)
- Rectangle from extent: Draws a rectangle from two opposite corners. (Left-click, right-click)
- Rectangle from 3 points (distance): Draws an oriented rectangle from three points. The first and second points determine the length and angle of the first edge. The third point determines the length of the other edge. One can use *O painel Digitalização avançada* to set the length of the edges. (Left-click, left-click, right-click)
- Rectangle from 3 points (projected): Same as the preceding tool, but the length of the second edge is computed from the projection of the third point on the first edge. (Left-click, left-click, right-click)

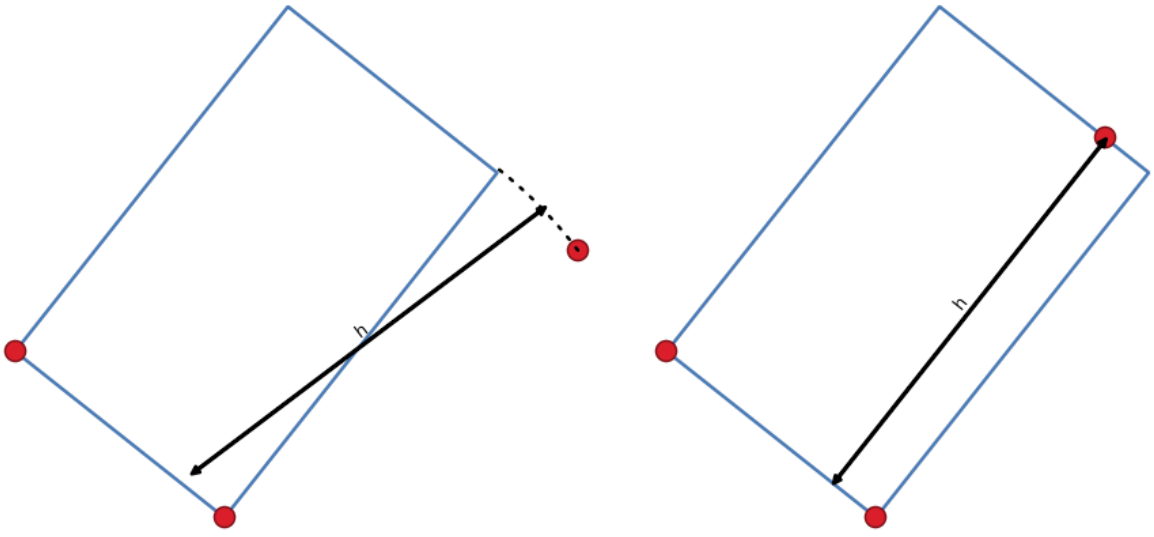


Figura12.92: Draw rectangle from 3 points using distance (right) and projected (left)

Desenhar Polígonos Regulares

There is a set of tools for drawing regular polygons. The tools are described below. Left-click to place the first point. A dialog appears, where you can set the number of polygon edges. Right-click to finish the regular polygon.

- Regular polygon from two points: Draws a regular polygon where the two points determine the length and angle of the first edge.
- Regular polygon from center and a point: Draws a regular polygon from the provided center point. The second point determines the angle and distance to the middle of an edge.
- Regular polygon from center and a corner: Same as the preceding tool, but the second point determines the angle and distante to a vertex.

12.3.7 O painel Digitalização avançada

When capturing, reshaping, splitting new or existing geometries you also have the possibility to use the Advanced Digitizing panel. You can digitize lines exactly parallel or perpendicular to a particular angle or lock lines to specific angles. Furthermore, you can make a precise definition of your new geometry by entering X and Y coordinates as well as Z for 3D features, or M values.

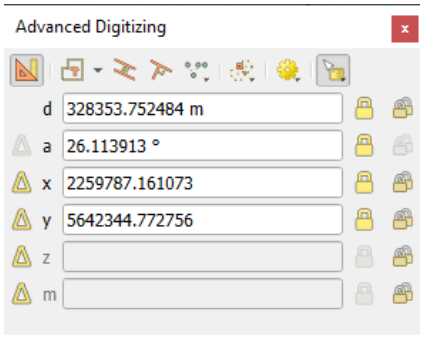


Figura12.93: O painel Digitalização avançada

The *Advanced Digitizing* panel can be opened either with a right-click on the toolbar, from *View* ► *Panels* ► menu or pressing `Ctrl+4`. Once the panel is visible, click the Enable advanced digitizing tools button to activate the set of tools.

Nota

As ferramentas não são habilitadas se a visão do mapa está em coordenadas geográficas.

The aim of the Advanced Digitizing tool is to lock coordinates, lengths, and angles when moving the mouse during the digitalizing in the map canvas.

You can also create constraints with relative or absolute reference. Relative reference means that the next vertex constraints' values will be relative to the previous vertex or segment.

The toolbar

At the top of the *Digitizing panel*, you find the following buttons:

- Enable advanced digitizing tools
- Construction mode: allows to capture the clicks' positions to reuse as reference points to lock distance, angle, X, Y, Z or M relative values. More details at *Modo de construção*.
- Parallel to draw a line parallel to an existing one (more at *Linhas paralelas e perpendiculares*)
- Perpendicular to draw a line perpendicular to an existing one (more at *Linhas paralelas e perpendiculares*)
- Construction Tools provides a couple of options that constrain the vertices placement based on extrapolated coordinates of existing elements:
  - *Line Extension*: hover over a segment and you get a purple dotted line extending the segment across the map canvas. You can snap the vertex anywhere on this virtual line.
  - *X/Y Point*: hover over a vertex and you get a purple dotted line along its X or Y coordinate, across the map canvas. You can snap the vertex anywhere on this virtual line. It is even possible to hover over two different vertices, generating virtual coordinate lines for both, and snap to their intersection.
- 2-circle Point Intersection: allows you to digitize a point or vertex at the intersection of two circles. (more at *2-circle point intersection*).
- Snap to common angles: when moving the cursor, displays a virtual line that you can snap to to add the next vertex. The snapping line is defined by the last added vertex and an (absolute or relative to previous segment) angle from a preset list (following steps of 0.1°, 0.5°, 1°, 5°, 10°, 15°, 18°, 22.5°, 30°, 45° or 90°). Choose *Do not snap to common angles* to disable this feature.

  Snapping to features can be used along with snapping to common angles for accurate digitizing. For a fine-grained control on how the target element to snap to is retained, you can indicate whether to prioritize snapping to features over common angles, and vice-versa under the *Snapping priority* entry. You can switch from one method to the other during the digitizing operation, and this avoids disabling any of the snapping options in the meantime. Press `N` (or `Shift+N`) during a digitizing operation to cycle through the angles list.
- Floater settings: if the *Show floater* item is checked, a contextual menu with digitizing information follows the cursor during digitizing. The values can be accessed using the *panel's shortcuts*, edited and Locked after validation (pressing `Enter`). The type of information to display can be selected in the bottom part of the menu:
  - *Show distance*
  - *Show angle*
  - *Show XY coordinates*
  - *Show Z value*
  - *Show M value*
  - *Show bearing/azimuth*
  - *Show common snapping angle*

Below the toolbar, you will find a number of text boxes whose value reflects by default the position or movement of the cursor in the map canvas. Editing these values helps you constrain the position of the items you edit:

- *d* for the distance from a reference position, usually the last edited vertex
- *a* for the angle (absolute or relative) from a reference position, usually the last edited segment
- *x* for the X coordinate of the pointer

- *y* for the Y coordinate of the pointer
- *z* for the default Z value or the Z coordinate of the vertex or segment under the pointer
- *m* for the default M value or the M value of the vertex or segment under the pointer

Atalhos do Teclado

To speed up the use of Advanced Digitizing Panel, there are a couple of keyboard shortcuts available:

Tabela12.5: Keyboard shortcuts of the Advanced Digitizing Panel tools

Chave	Simples	`Ctrl+` ou `Alt+`	`Shift+`
`D`	Definir distância	Lock distance	
`A`	Definir Ângulo	Lock angle	Toggle relative angle to last segment
`X`	Definir coordenada X	Lock X coordinate	Toggle relative X to last vertex
`Y`	Definir coordenada Y	Lock Y coordinate	Toggle relative Y to last vertex
`Z`	Definir coordenada Z	Lock Z coordinate	Toggle relative Z to last vertex
`M`	Definir valor M	Lock M value	Toggle relative M to last vertex
`C`	Alternar modo de construção		
`P`	Alternar entre modos perpendicular e paralelo		

Nota

As opções de coordenada Z e valor M estão disponíveis somente se forem compatíveis com a dimensão da geometria da camada.

Absolute reference digitizing

When drawing a new geometry from scratch, it is very useful to have the possibility to start digitizing vertexes at given coordinates.

For example, to add a new feature to a polygonal layer, click the button. You can enter the exact coordinates where you want to start editing the feature, i.e.:

1. Click the *x* text box (or use the `X` keyboard shortcut).
2. Type the X coordinate value you want and press `Enter` or click the button to their right to lock the mouse to the X axis on the map canvas.
3. Click the *y* text box (or use the `Y` keyboard shortcut).
4. Type the Y coordinate value you want and press `Enter` or click the button to their right to lock the mouse to the Y axis on the map canvas.
5. If the layer has Z coordinate or M values, the corresponding *z* or *m* widget is enabled and displays its default value, as set in *Settings* ► *Options* ► *Digitizing* tab.
    1. Click the *z* or *m* text box (or use respectively the `Z` or `M` keyboard shortcut).
    2. Type the coordinate value you want and press `Enter` or click the button to their right to lock the value in the widget.

> **Nota**
>
> Read *Rules of Z coordinate or M value assignment* for details on how Z coordinate and M values are automatically determined from existing features.

6. Two blue dotted lines and a green cross identify the exact coordinates you entered. Click on the map canvas to add a vertex at the green cross position.

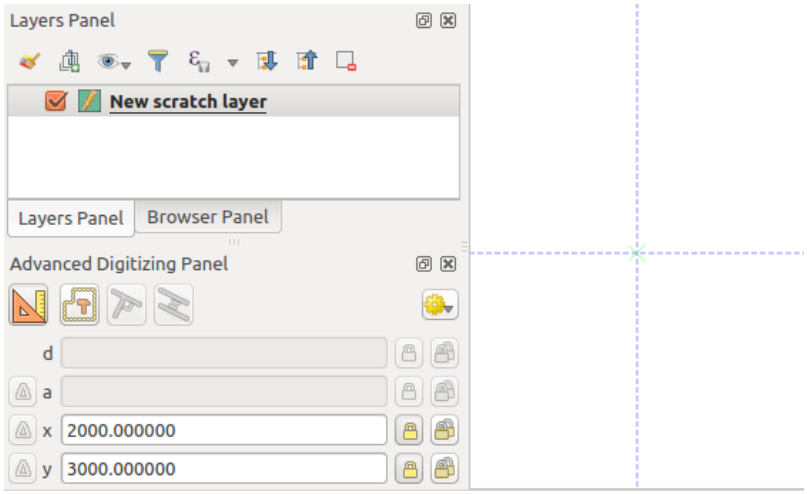


Figura12.94: Comece a desenhar nas coordenadas fornecidas

7. You can proceed as above, adding a new set of coordinates for the next vertex, or switch to another *mode of digitizing* (e.g. segment, curve or stream).
8. Se você quiser desenhar um segmento de um determinado comprimento:
    1. Click the *d (distance)* text box (keyboard shortcut `D`)
    2. Digite o valor da distância (em unidades do mapa)
    3. Press `Enter` or click the button on the right to lock the mouse in the map canvas to the length of the segment. In the map canvas, the latest vertex is surrounded by a circle whose radius is the value entered in the distance text box. A cross on the circle shows the position of the next vertex if you click.

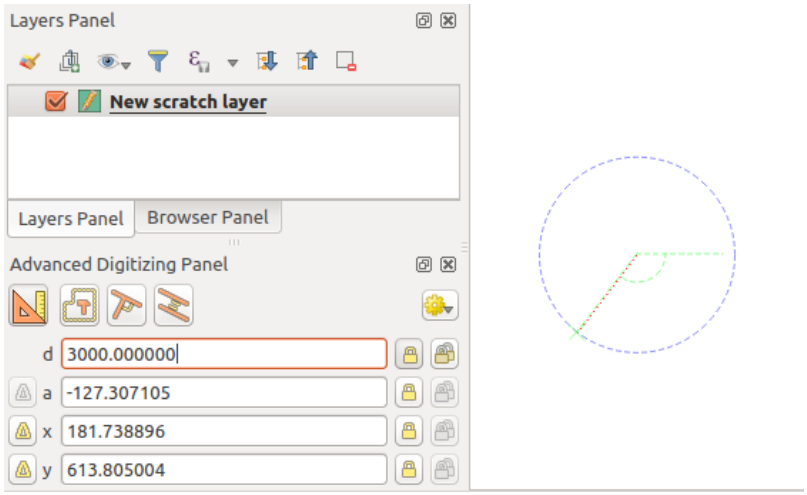


Figura12.95: Fixed length segment

9. Você também pode restringir a posição do vértice, definindo o ângulo do segmento. Conforme descrito anteriormente:
    1. Click the *a (angle)* text box (keyboard shortcut `A`)
    2. Digite o valor do ângulo (em graus)
    3. Press `Enter` or click the button on the right to lock it. A line going through the latest vertex and rotated based on the set angle appears in the map canvas and a cross on it shows the next vertex position if you click.

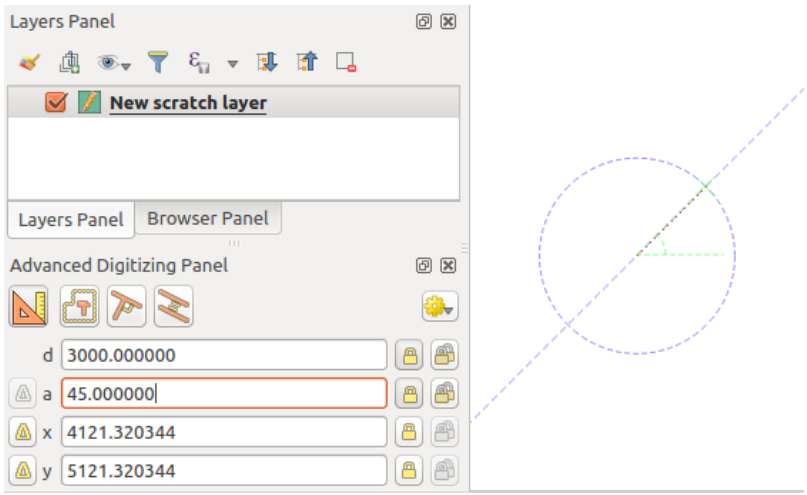


Figura12.96: Fixed angle segment

Dica

Pressing `Ctrl+<key>` or `Alt+<key>` automatically locks the target property and puts its value into edit. Modify, press `Enter` and you are done. Combined with the Toggle floater, this can be a real time saver, with keyboard digitizing.

Relative reference digitizing

Instead of using absolute values of angles or coordinates, you can also use values relative to the last digitized vertex or segment.

For angles, you can click the button on the left of the *a* text box (or press `Shift+A`) to toggle relative angles to the previous segment. With that option on, angles are measured between the last segment and the mouse pointer.

For coordinates, click the buttons to the left of the *x*, *y*, *z* or *m* text boxes (or press `Shift+<key>`) to toggle relative coordinates to the previous vertex. With these options on, coordinates measurement will consider the last vertex to be the origin of the set coordinates.

Continuous lock

Both in absolute or relative reference digitizing, angle, distance, X, Y, Z and M constraints can be locked continuously by clicking the *Continuous lock* buttons. Using continuous lock allows you to digitize several points or vertexes using the same constraints.

Modo de construção

You can enable and disable *construction mode* by clicking on the Construction mode icon or with the `C` keyboard shortcut. While in construction mode, clicking the map canvas won't add new vertexes, but will capture the clicks' positions so that you can use them as reference points to then lock distance, angle or X, Y, Z, M relative values.

In the Construction mode you will find a drop-down menu where you can choose to:

- *Record Construction Guides*: All construction steps are rendered as dashed lines. Each step that you perform is visually represented, allowing you to trace the construction process. The guides are displayed as long as *Advanced Digitizing* remains active.
- *Show Construction Guides*: Allows you to make construction guides visible or hidden on the canvas. When enabled, all active guides are displayed, offering better spatial orientation and precision during the digitizing process.
- *Snap to Visible Construction Guides*: The guides are snap-able, allowing you to start new construction steps from any point along the existing guides.
- Choose *Clear Construction Guides* to remove all the guides from the canvas.

Como exemplo, o modo de construção pode ser usado para desenhar algum ponto a uma distância exata de um ponto existente.

With an existing point in the map canvas and the snapping mode correctly activated, you can easily draw other points at given distances and angles from it. In addition to the button, you have to activate also the *construction mode* by clicking the Construction mode icon or with the `C` keyboard shortcut.

Click next to the point from which you want to calculate the distance and click on the *d* box (`D` shortcut) type the desired distance and press `Enter` to lock the mouse position in the map canvas:

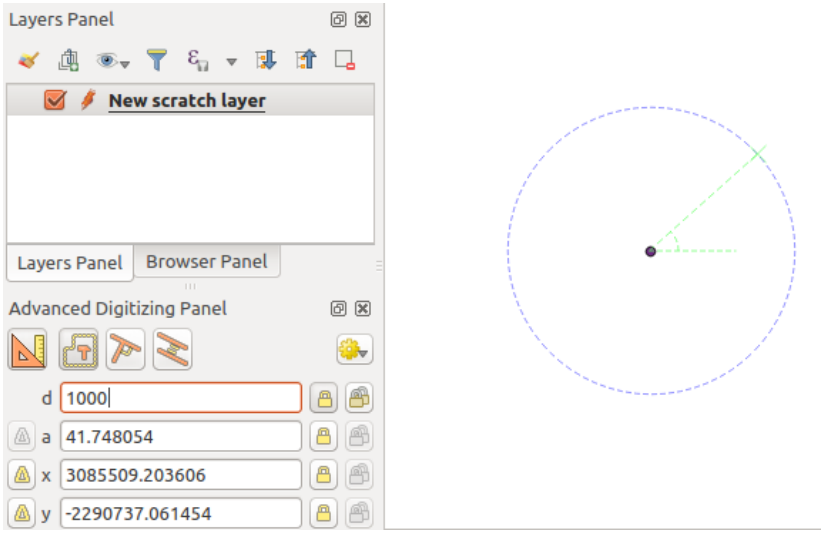


Figura12.97: Distância do ponto

Before adding the new point, press `C` to exit the construction mode. Now, you can click on the map canvas, and the point will be placed at the distance entered.

You can also use the angle constraint to, for example, create another point at the same distance of the original one, but at a particular angle from the newly added point. Click the Construction mode icon or with the `C` keyboard shortcut to enter construction mode. Click the recently added point, and then the other one to set a direction segment. Then, click on the *d* text box (`D` shortcut) type the desired distance and press `Enter`. Click the *a* text box (`A` shortcut) type the angle you want and press `Enter`. The mouse position will be locked both in distance and angle.

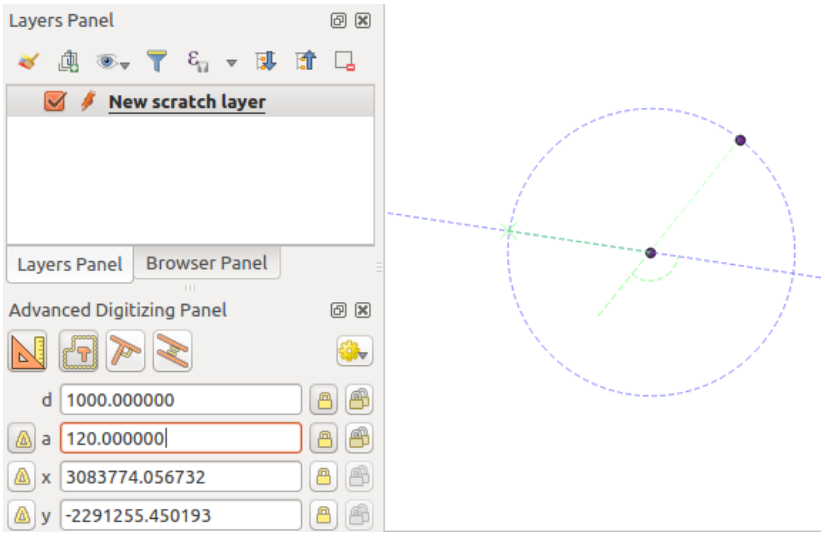


Figura12.98: Distância e ângulo dos pontos

Before adding the new point, press `C` to exit the construction mode. Now, you can click on the map canvas, and the point will be placed at the distance and angle entered. Repeating the process, several points can be added.

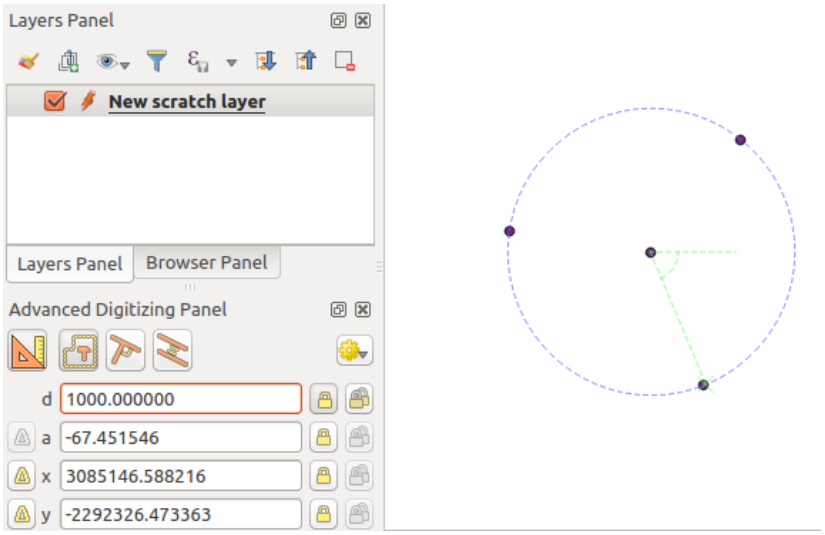


Figura12.99: Pontos a uma dada distância e ângulo

Linhas paralelas e perpendiculares

All the tools described above can be combined with the Perpendicular and Parallel tools. These two tools allow drawing segments perfectly perpendicular or parallel to another segment. The target segment can be on another layer, another feature within the layer or the feature being digitized (requires *self-snapping option*).

Para desenhar um segmento *perpendicular*:

1. Primeiro adicione um dos vértices do segmento.
2. Click the Perpendicular icon (keyboard shortcut `P`) to activate it.
3. Clique no segmento ao qual você deseja ser perpendicular.
4. A virtual dotted line perpendicular to the segment through the previous vertex appears. The angle property is locked, constraining the next vertex on that line and, a cross indicates the projected position of the cursor on the line. Click to place the new vertex.

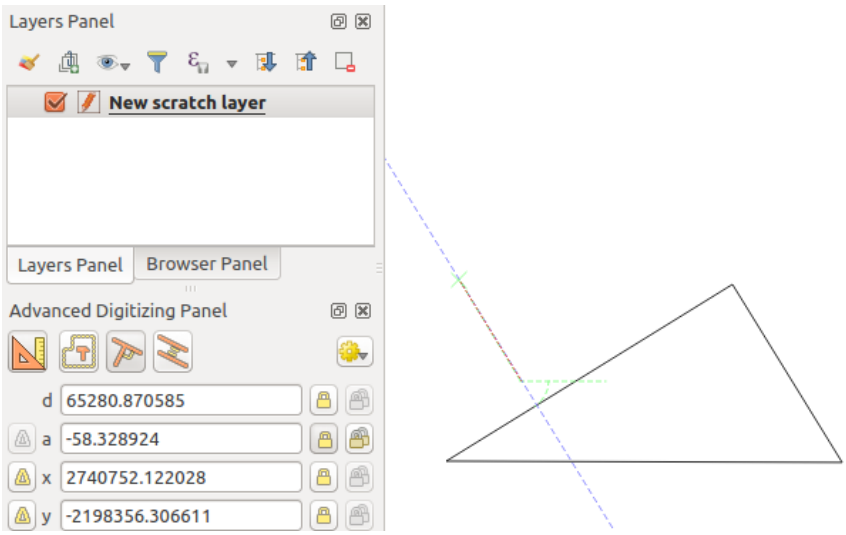


Figura12.100: Digitalização perpendicular

To draw a *parallel* segment, the steps are the same except that you need to click on the Parallel icon (keyboard shortcut `P` twice).

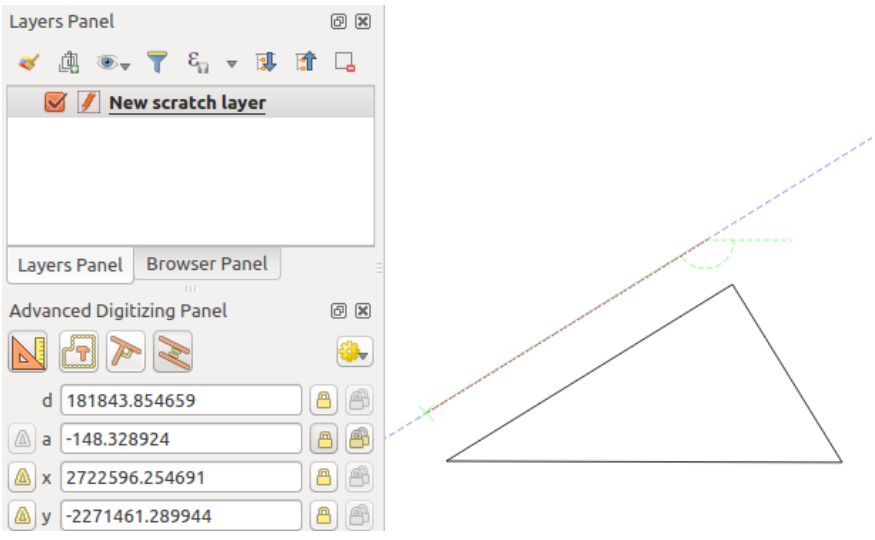


Figura12.101: Digitalização paralela

Essas duas ferramentas apenas encontram o ângulo certo do ângulo perpendicular e paralelo e bloqueiam esse parâmetro durante sua edição. Desbloqueie o parâmetro ângulo para cancelar seu uso no meio do processo.

2-circle point intersection

To add a vertex at the intersection of two circles, follow these steps:

1. Click the 2-circle Point Intersection icon.
2. A dialog will open where you can define the parameters for *Circle #1* and *Circle #2*.
3. Click on the map canvas, and the tool will automatically calculate the *X* and *Y* coordinates for the centers of both circles.
4. Enter the distance *d* for each circle.
5. The tool will calculate and display the two intersection points of the circles.
6. Click one of the intersection points to add the new vertex.

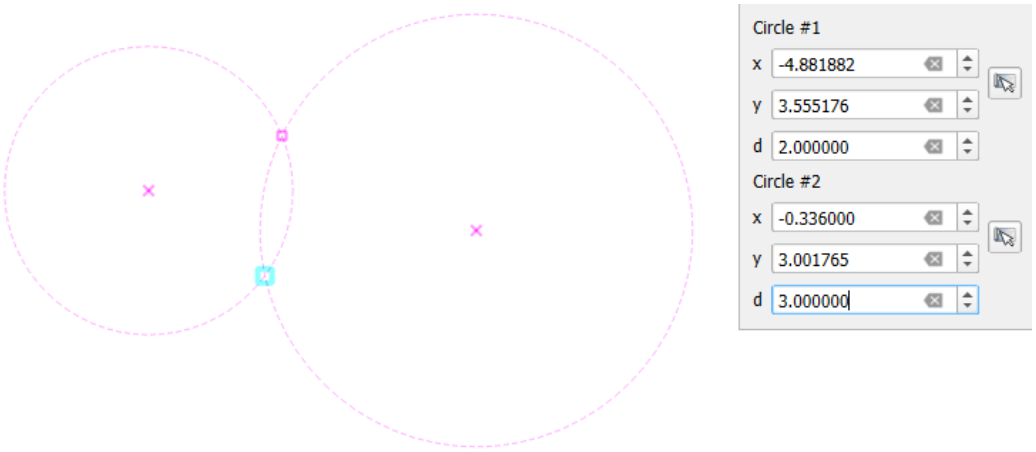


Figura12.102: 2-circle Point Intersection

12.3.8 The Processing in-place layer modifier

The *Processing menu* provides access to a large set of tools to analyze and create new features based on the properties of the input features or their relations with other features (within the same layer or not). While the common behavior is to create new layers as outputs, some algorithms also allow modifications to the input layer. This is a handy way to automate multiple features modification using advanced and complex operations.

Para editar feições no local:

1. Select the layer to edit in the *Layers* panel.
2. Selecione as feições em questão. Você pode pular esta etapa, nesse caso a modificação será aplicada a toda a camada.
3. Press the Edit Features In-Place button at the top of the *Processing toolbox*. The list of algorithms is filtered, showing only those compatible with in-place modifications, i.e.:
   - Eles funcionam na origem da feição e não no nível da camada.
   - Eles não alteram a estrutura da camada, por ex. adicionando ou removendo campos.

- Eles não alteram o tipo de geometria, por ex. de camada de linha para ponto.

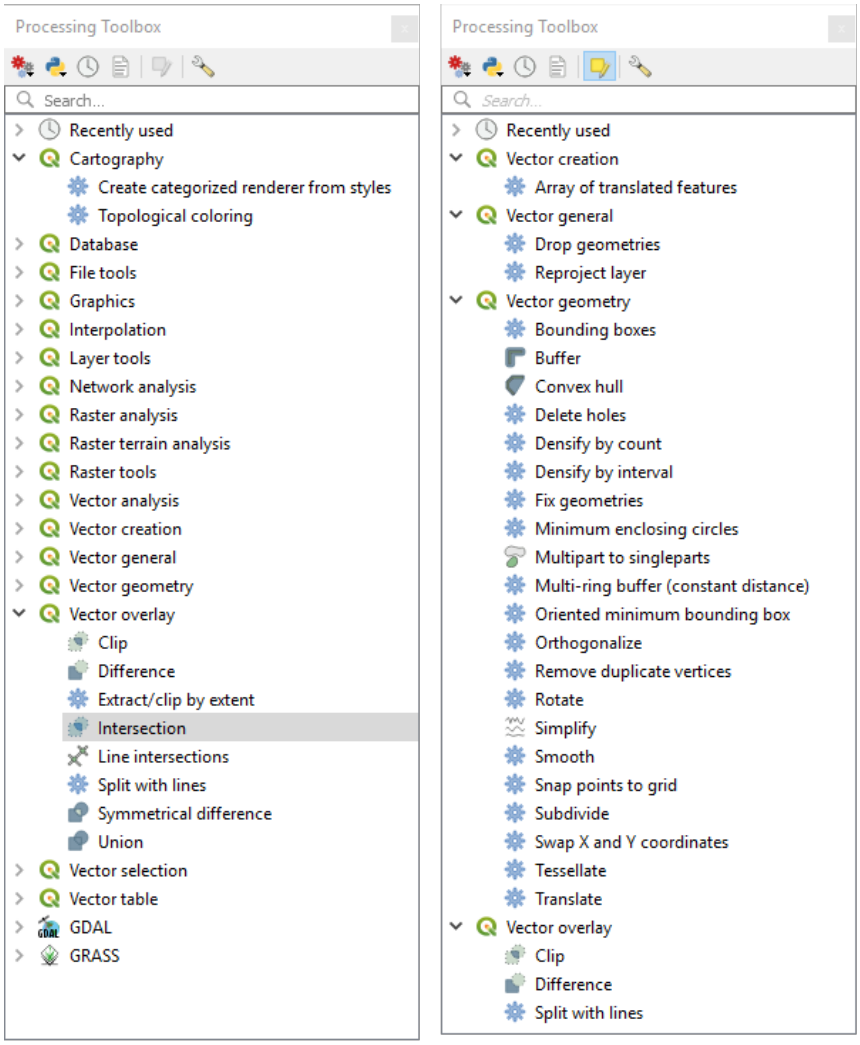


Figura12.103: Algoritmos de processamento: todos (esquerda) vs editores de polígonos no local (direita)

4. Encontre o algoritmo que deseja executar e clique duas vezes nele.

> **Nota**
>
> Se o algoritmo não precisar de nenhum parâmetro adicional definido pelo usuário (excluindo os parâmetros usuais da camada de entrada e saída), o algoritmo será executado imediatamente sem nenhum pop-up de diálogo.

   1. Se forem necessários parâmetros diferentes das camadas usuais de entrada ou saída, a caixa de diálogo do algoritmo será exibida. Preencha as informações necessárias.
   2. Click *Modify Selected Features* or *Modify All Features* depending on whether there's an active selection.

   Changes are applied to the layer and placed in the edit buffer: the layer is indeed toggled to editing mode with unsaved modification as indicated by the icon next to the layer name.

5. As usual, press Save layer edits to commit the changes in the layer. You can also press Undo to rollback the whole modification.

12.4 Conectando e Editando Dados Através de Camadas

Ability to connect data from different layers is one of the duties of a GIS software. Such a connection can be based on the spatial relationship between the features, or on their shared attributes. QGIS provides tools to handle any of these associations, such as:

- Processing algorithms that can create a new layer as a result of the connection, namely *Unir atributos por localização*, *Unir atributos por mais próximo*, *Unir atributos por valor de campo*, …
- SQL queries to create a new layer from the *DB Manager* or as a *virtual layer*
- *Joins properties* or *relations settings* that temporarily extend attributes of features in a given layer, with those of features in another layer based on some matching attribute(s).

  Joins and relations are technical concepts borrowed from databases to get the most out of data stored in tables by combining their contents. The idea is that features (rows) of different layers (tables) can be associated to each other. The number of rows which are matching each other can be of any value (zero, one, many).

12.4.1 Joining features between two layers

Joins in QGIS allow you to associate features of the current layer to features from another loaded vector layer. Whether they are spatially enabled and the type of geometry do not matter. The join is based on an attribute that is shared by the layers, in a one-to-one relationship.

To create a join on a layer (identified below as `target layer`):

1. Go to the layer *Properties* ► *Joins* tab
2. Clique no botão | symbologyAdd | :sup:botão ‘Adicionar nova junção`. A caixa de diálogo:guilabel:‘Adicionar junção de vetor` aparece.
3. Selecione o:guilabel:*camada de junção* que deseja conectar com a camada vetorial de destino
4. Specify the *Join field* (from the `join layer`) and the *Target field* (from the `target layer`). These are the fields that are used to find matching feature in both layers hence they should have values in common.
5. Pressione:guilabel:‘OK` e um resumo dos parâmetros selecionados é adicionado ao painel:guilabel:*Join*.

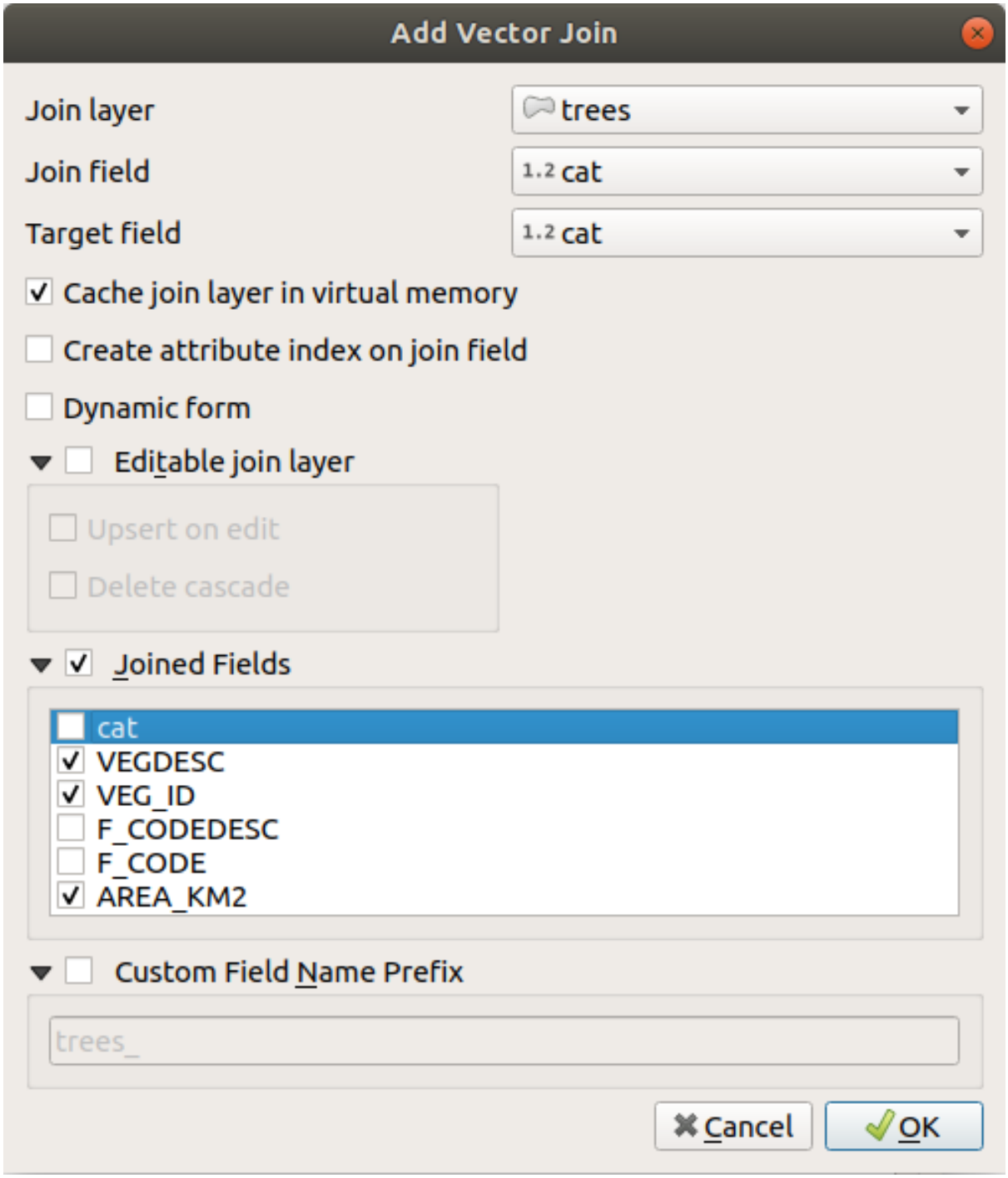


Figura12.104: Una uma tabela de atributos a uma camada vetorial existente

The steps above will create a join, where **ALL** the attributes of the **first matching feature** in the join layer is added to the target layer's feature. The following logic is used to pair features during a join process:

- All the features in the target layer are returned, regardless they have a match
- If the target field contains duplicate values, these features are assigned the same feature from the join layer.
- If the join field contains duplicate matching values, only the first fetched feature is picked.

Nota

Joins in QGIS are based on a single field matching so most of the times, you would want to make sure that values in the matchable fields are unique.

QGIS provides some more options to tweak the join:

- *Cache join layer in virtual memory*: allows you to cache values in memory (without geometries) from the joined layer in order to speed up lookups.
- *Create attribute index on the join field* to speed up lookups
- *Dynamic form*: helps to synchronize join fields on the fly, according to the *Target field*. This way, constraints for join fields are also correctly updated. Note that it's deactivated by default because it may be very time consuming if you have a lot of features or a myriad of joins.
- Caso a camada de destino seja editável, alguns ícones serão exibidos na tabela de atributos ao lado dos campos, para informar sobre o seu estado:
  - : the join layer is not configured to be editable. If you want to be able to edit join features from the target attribute table, then you have to check the option *Editable join layer*.
  - : the join layer is well configured to be editable, but its current status is read only.
  - : the join layer is editable, but synchronization mechanisms are not activated. If you want to automatically add a feature in the join layer when a feature is created in the target layer, then you have to check the option *Upsert on edit*. Symmetrically, the option *Delete cascade* may be activated if you want to automatically delete join features.
- *Joined fields*: instead of adding all the fields from the joined layer, you can specify a subset.
- *Custom field name prefix* for joined fields, in order to avoid name collision

12.4.2 Setting relations between multiple layers

Unlike joins that define a one-to-one link between features across two layers, relations help you build interconnections between multiple features across two or more layers. As such, relations are project level settings and are set in *Project* ► *Properties* ► *Relations* tab. From there, you can:

- *Add relation* whose type can be:
  - *one to many relation*
  - *many to many relation*
  - *polymorphic relation* that you can add or edit with the dedicated tools in the action drop-down menu.

> **Nota**
>
> There is no simple way yet to edit a non-polymorphic relation once it has been created. Only the name can be edited with a double-click. For any other parameters of such a relation you will have to remove and recreate it.

- *Discover relations*: QGIS is able to discover existing relations from supported database formats (PostgreSQL, GeoPackage, ESRI File Geodatabase, …). This can be a good way to ease the relations definition.
- *Remove relation*

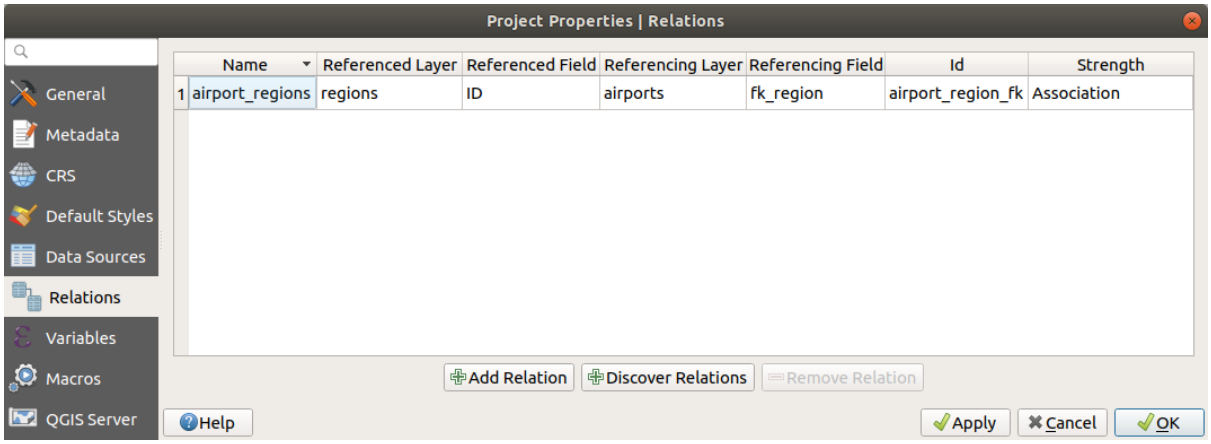


Figura12.105: Aba de relações

One to many (1-N) relations

As an example you have a layer with all regions of Alaska (polygon) which provides some attributes about its name and region type and a unique id (which acts as primary key).

Em seguida, você obtém outra camada ou tabela de pontos com informações sobre aeroportos localizados nas regiões e também deseja acompanhá-los. Se você deseja adicioná-los à camada de regiões, você precisa criar uma relação de um para muitos usando chaves estrangeiras, pois existem vários aeroportos na maioria das regiões.

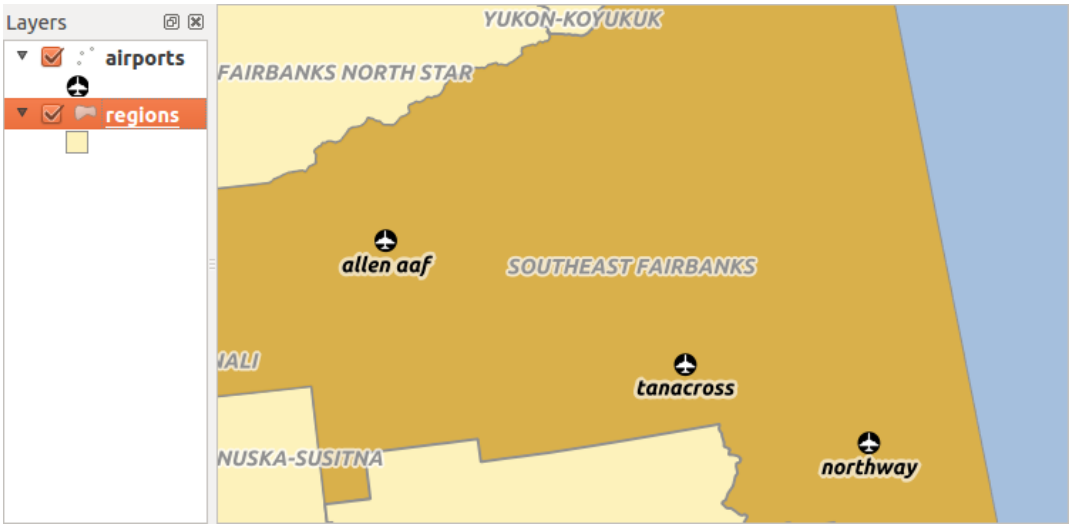


Figura12.106: Região do Alasca com aeroportos

Layers and keys

QGIS makes no difference between a table and a vector layer. Basically, a vector layer is a table with a geometry. So you can add your table as a vector layer. To demonstrate the 1-n relation, you can load the `regions` and `airports` layers in the sample dataset. In practice, each airport belongs to exactly one region while each region can have any number of airports (a typical one to many relation).

In addition to the attributes describing the airports, the aiports layer has another field `fk_region` which acts as a foreign key (if you have a database, you will probably want to define a constraint on it). This `fk_region` field will always contain an id of a region. It can be seen like a pointer to the region it belongs to.

All you have to do is to tell QGIS the relation between the layers so that you can design a custom edit form for editing and QGIS takes care of the setup. It works with different providers (so you can also use it with shape and csv files).

Definindo relações 1-N

The first thing we are going to do is to let QGIS know about the relations between the layers. This is done in *Project* ► *Properties…*. Open the *Relations* tab and click on *Add Relation*.

- **Name** is going to be used as a title. It should be a human readable string describing what the relation is used for. We will just call say **airport_relation** in this case.
- **Referenced Layer (Parent)** also considered as parent layer, is the one with the primary key, pointed to, so here it is the `regions` layer. You need to define the primary key of the referenced layer, so it is `ID`.
- **Referencing Layer (Child)** also considered as child layer, is the one with the foreign key field on it. In our case, this is the `airports` layer. For this layer you need to add a referencing field which points to the other layer, so this is `fk_region`.

> **Nota**
>
> Sometimes, you need more than a single field to uniquely identify features in a layer. Creating a relation with such a layer requires a **composite key**, i.e. more than a single pair of matching fields. Use the Add new field pair as part of a composite foreign key button to add as many pairs as necessary.

- **Id** will be used for internal purposes and has to be unique. You may need it to build *custom forms*. If you leave it empty, one will be generated for you but you can assign one yourself to get one that is easier to handle
- **Relationship strength** sets the strength of the relation between the parent and the child layer. The default *Association* type means that the parent layer is *simply* linked to the child one while the *Composition* type allows you to duplicate also the child features when duplicating the parent ones and on deleting a feature the children are deleted as well, resulting in cascade over all levels (means children of children of… are deleted as well).

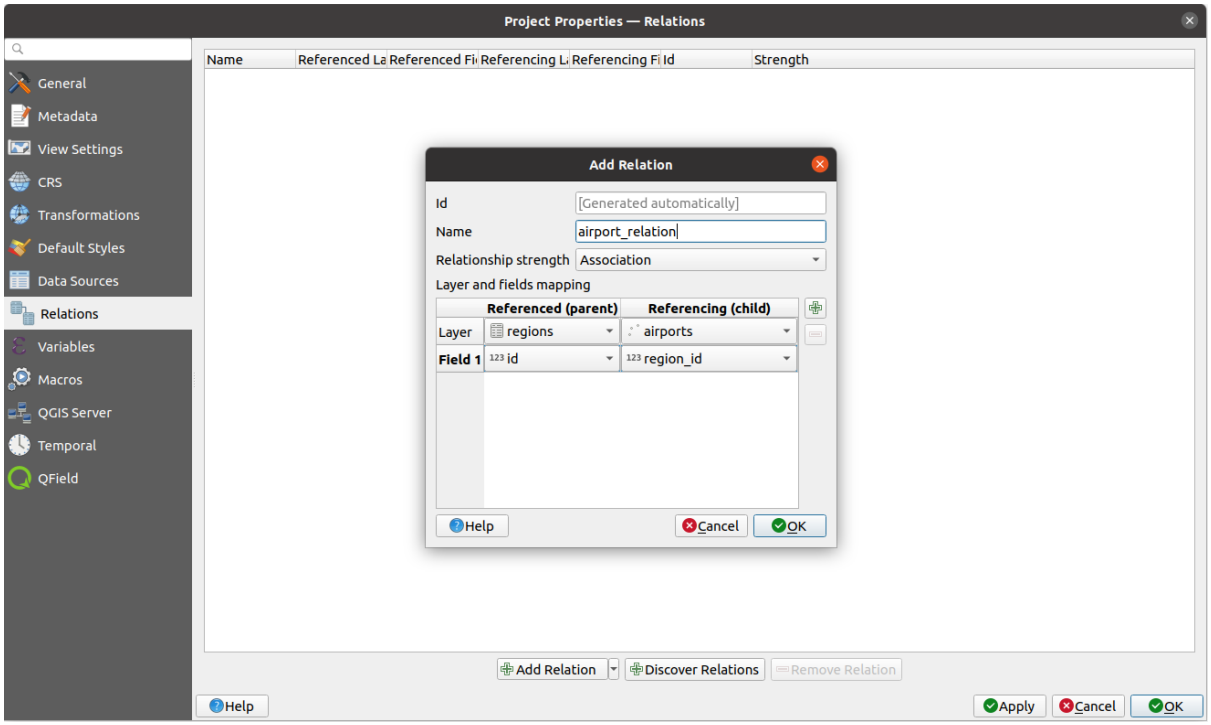


Figura12.107: Adicionando uma relação entre camadas de regiões e de aeroportos

From the *Relations* tab, you can also press the *Discover Relation* button to fetch the relations available from the providers of the loaded layers. This is possible for layers stored in data providers like PostgreSQL or SpatiaLite.

Formulários para relações 1-N

Now that QGIS knows about the relation, it will be used to improve the forms it generates. As we did not change the default form method (autogenerated), it will just add a new widget in our form. So let's select the layer region in the legend and use the identify tool. Depending on your settings, the form might open directly or you will have to choose to open it in the identification dialog under actions.

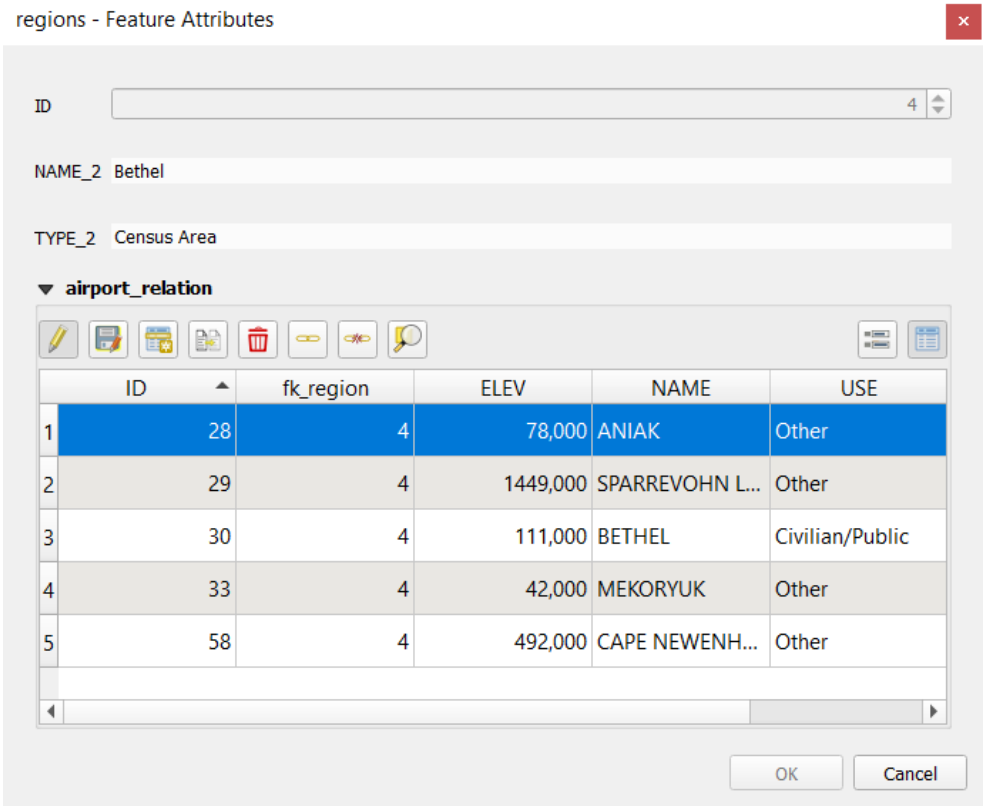


Figura12.108: Regiões de diálogo de identificação com relação a aeroportos

As you can see, the airports assigned to this particular region are all shown in a table. And there are also some buttons available. Let's review them shortly:

- The button is for toggling the edit mode. Be aware that it toggles the edit mode of the airport layer, although we are in the feature form of a feature from the region layer. But the table is representing features of the airport layer.
- O botão |Salvar Edições | é para salvar todas as edições na camada infantil (aeroporto).
- The button lets you digitize the airport geometry in the map canvas and assigns the new feature to the current region by default. Note that the icon will change according to the geometry type.
- The button adds a new record to the airport layer attribute table and assigns the new feature to the current region by default. The geometry can be drawn later with the *Add part* digitizing tool.
- The button allows you to copy and paste one or more child features within the child layer. They can later be assigned to a different parent feature or have their attributes modified.
- The button deletes the selected airport(s) permanently.
- The symbol opens a new dialog where you can select any existing airport which will then be assigned to the current region. This may be handy if you created the airport on the wrong region by accident.

- The symbol unlinks the selected airport(s) from the current region, leaving them unassigned (the foreign key is set to NULL) effectively.
- With the button you can zoom the map to the selected child features.
- The two buttons and to the right switch between the *table view and form view* of the related child features.

Se você utilizar a função Design de arrastar e soltar' para as regiões, você pode selecionar quais ferramentas estão disponíveis. Você pode até decidir se deseja abrir um novo formulário quando um novo recurso for adicionado utilizando a opção :guilabel:`Forçar a forma de ocultação no recurso adicionar. Esteja ciente de que esta opção implica que atributos não nulos devem ter um valor padrão válido para funcionar corretamente.

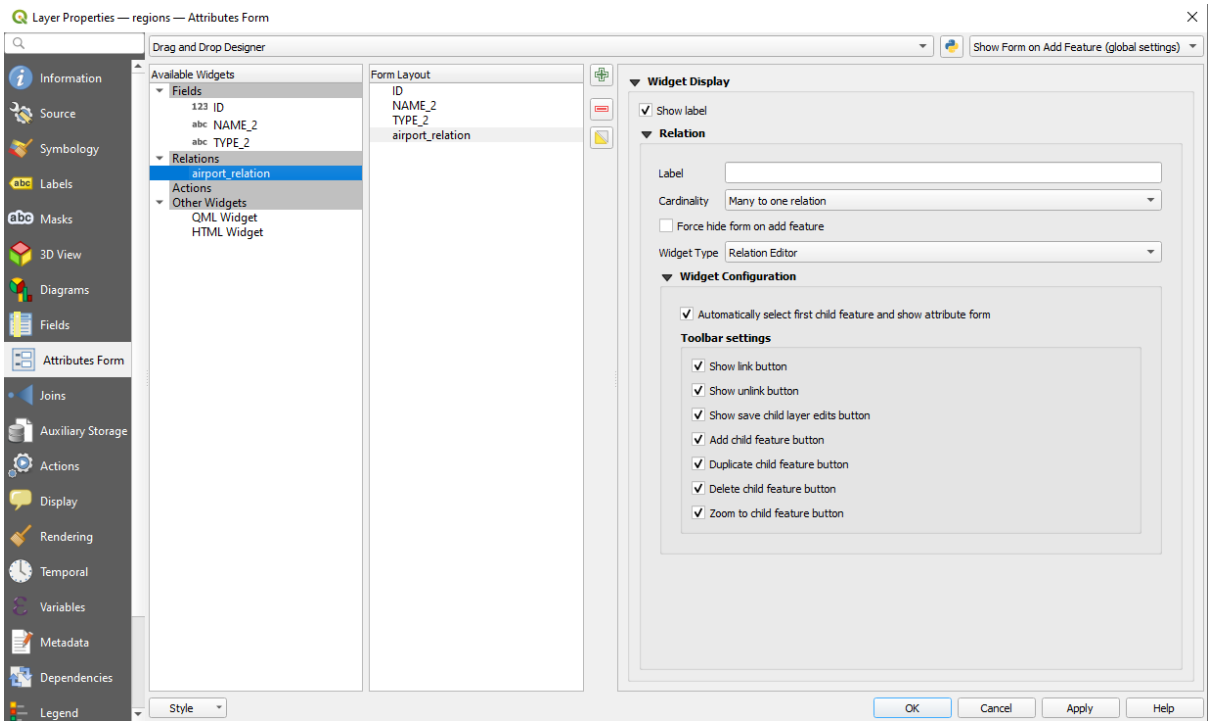


Figura12.109: Design de arrasta/solta para configuração de regiões - ferramentas de relação aeroportos

In the above example the referencing layer has geometries (so it isn't just an alphanumeric table) so the above steps will create an entry in the layer attribute table that has no corresponding geometric feature. To add the geometry:

1. Choose *Open Attribute Table* for the referencing layer.
2. Select the record that has been added previously within the feature form of the referenced layer.
3. Use the Add Part digitizing tool to attach a geometry to the selected attributes table record.

If you work on the airport table, the widget Relation Reference is automatically set up for the `fk_region` field (the one used to create the relation), see *Relation Reference widget*.

In the airport form you will see the button at the right side of the `fk_region` field: if you click on the button the form of the region layer will be opened. This widget allows you to easily and quickly open the forms of the linked parent features.

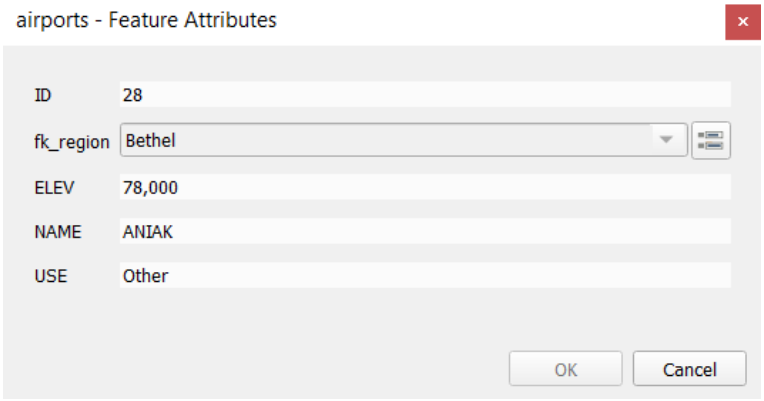
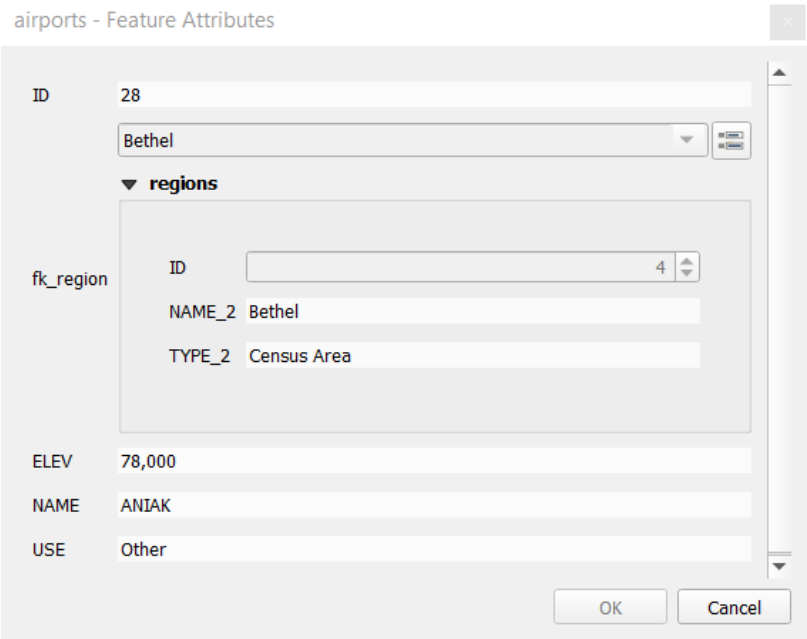


Figura12.110: Diálogo de identificação do aeroporto com relação às regiões

The Relation Reference widget has also an option to embed the form of the parent layer within the child one. It is available in the *Properties* ► *Attributes Form* menu of the airport layer: select the `fk_region` field and check the `Show embedded form` option.

If you look at the feature dialog now, you will see that the form of the region is embedded inside the airports form and will even have a combobox, which allows you to assign the current airport to another region.



Moreover if you toggle the editing mode of the airport layer, the `fk_region` field has also an autocompleter function: while typing you will see all the values of the `id` field of the region layer. Here it is possible to digitize a polygon for the region layer using the button if you chose the option `Allow adding new features` in the *Properties* ► *Attributes Form* menu of the airport layer.

The child layer can also be used in the *Selecionar Feições Por Valor* tool in order to select features of the parent layer based on attributes of their children.

In Fig.12.111, all the regions where the mean altitude of the airports is greater than 500 meters above sea level are selected.

Você verá que muitas funções de agregação diferentes estão disponíveis no formulário.

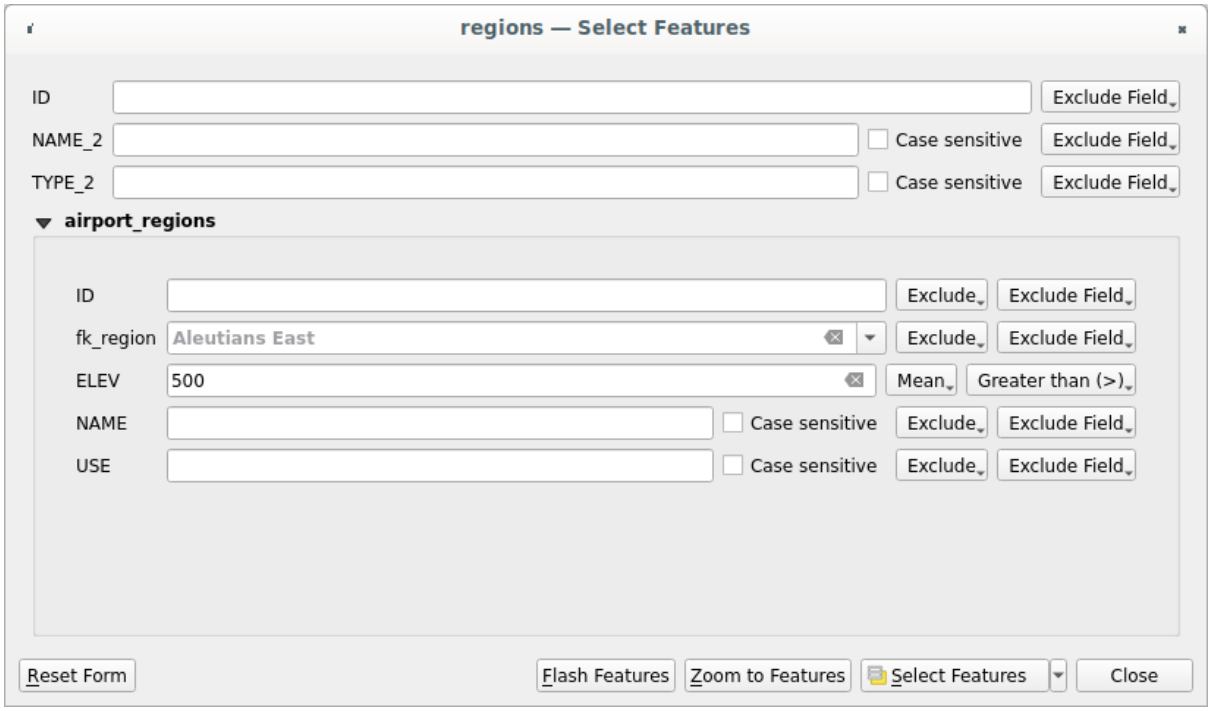


Figura12.111: Selecione as características dos pais com valores infantis

Many-to-many (N-M) relations

N-M relations are many-to-many relations between two tables. For instance, the `airports` and `airlines` layers: an airport receives several airline companies and an airline company flies to several airports.

This SQL code creates the three tables we need for an N-M relationship in a PostgreSQL schema named *locations*. You can run the code using the *Database* ► *DB Manager*... for PostgreSQL or external tools such as pgAdmin. The airports table stores the `airports` layer and the airline table stores the `airlines` layer. In both tables few fields are used for clarity. The *tricky* part is the `airports_airlines` table. We need it to list all airlines for all airports (or vice versa). This kind of table is known as a *pivot table*. The *constraints* in this table force that an airport can be associated with an airline only if both already exist in their layers.

```
CREATE SCHEMA locations;

CREATE TABLE locations.airports
(
   id serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   airport_name text NOT NULL,
   CONSTRAINT airports_pkey PRIMARY KEY (id)
);

CREATE INDEX airports_geom_idx ON locations.airports USING gist (geom);

CREATE TABLE locations.airlines
(
   id serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   airline_name text NOT NULL,
   CONSTRAINT airlines_pkey PRIMARY KEY (id)
);

CREATE INDEX airlines_geom_idx ON locations.airlines USING gist (geom);
```

(continua na próxima página)

(continuação da página anterior)

```
CREATE TABLE locations.airports_airlines
(
   id serial NOT NULL,
   airport_fk integer NOT NULL,
   airline_fk integer NOT NULL,
   CONSTRAINT airports_airlines_pkey PRIMARY KEY (id),
   CONSTRAINT airports_airlines_airport_fk_fkey FOREIGN KEY (airport_fk)
      REFERENCES locations.airports (id)
      ON DELETE CASCADE
      ON UPDATE CASCADE
      DEFERRABLE INITIALLY DEFERRED,
   CONSTRAINT airports_airlines_airline_fk_fkey FOREIGN KEY (airline_fk)
      REFERENCES locations.airlines (id)
      ON DELETE CASCADE
      ON UPDATE CASCADE
      DEFERRABLE INITIALLY DEFERRED
 );
```

Ao invés do PostgreSQL, você também pode usar o GeoPackage. Neste caso, as três tabelas podem ser criadas manualmente utilizando o *Base de Dados* ► *Gerenciar BD…*. No GeoPackage não há esquemas, portanto o prefixo *localizações* não é necessário.

Foreign key constraints in `airports_airlines` table can´t be created using *Table* ► *Create Table…* or *Table* ► *Edit Table…* so they should be created using *Database* ► *SQL Window…*. GeoPackage doesn't support *ADD CONSTRAINT* statements so the `airports_airlines` table should be created in two steps:

1. Configure a tabela somente com o campo "ID", utilizando *Tabela* ► *Criar Tabela…*
2. Utilizando *Base de dados* ► *SQL Window…*, digite e execute este código SQL:

```
ALTER TABLE airports_airlines
   ADD COLUMN airport_fk INTEGER
   REFERENCES airports (id)
   ON DELETE CASCADE
   ON UPDATE CASCADE
   DEFERRABLE INITIALLY DEFERRED;

ALTER TABLE airports_airlines
   ADD COLUMN airline_fk INTEGER
   REFERENCES airlines (id)
   ON DELETE CASCADE
   ON UPDATE CASCADE
   DEFERRABLE INITIALLY DEFERRED;
```

Então, no QGIS, você deve estabelecer duas: ref:*uma-para-muitas relações <one_to_many_relation>* como explicado acima:

- uma relação entre a "tabela de linhas aéreas" e a tabela dinâmica;
- e uma segunda entre a mesa "aeroportos" e a tabela dinâmica.

Uma maneira mais fácil de fazê-lo (somente para o PostgreSQL) é utilizar o :guilabel:`Relações de Descoberta' em :menuselection:`Projeto –> Propriedades –> Relações'. QGIS lerá automaticamente todas as relações em seu banco de dados e você só terá que selecionar as duas que você precisa. Lembre-se de carregar primeiro as três tabelas do projeto QGIS.

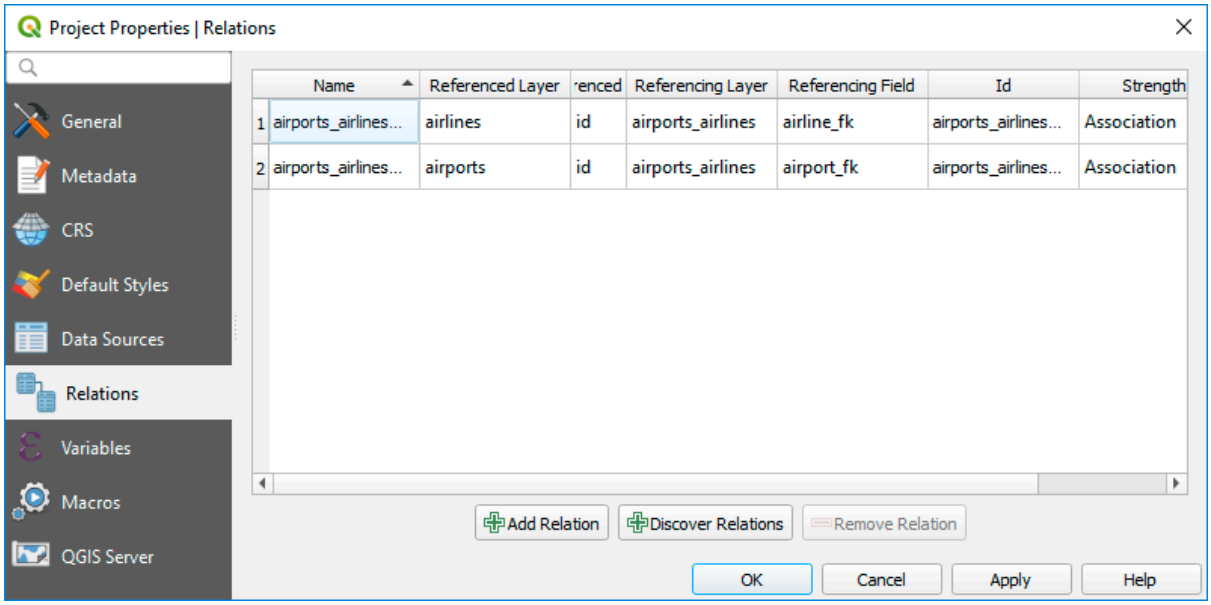


Figura12.112: Relações e auto-descobrimento

Caso você queira remover um “aeroporto” ou uma “linha aérea”, o QGIS não removerá o(s) registro(s) associado(s) na tabela “airports_airlines”. Esta tarefa será feita pelo banco de dados se especificarmos as *restrições* corretas na criação da tabela pivô, como no exemplo atual.

Nota

Combinação da relação N-M com grupo de transação automática

Você deve ativar o modo de transação em *Propriedades do projeto* ► *Fontes de dados* ► quando trabalhar em tal contexto. QGIS deve ser capaz de adicionar ou atualizar linha(s) em todas as tabelas (companhias aéreas, aeroportos e as tabelas pivot).

Finally we have to select the right cardinality in the *Layer Properties* ► *Attributes Form* for the `airports` and `airlines` layers. For the first one we should choose the **airlines (id)** option and for the second one the **airports (id)** option.

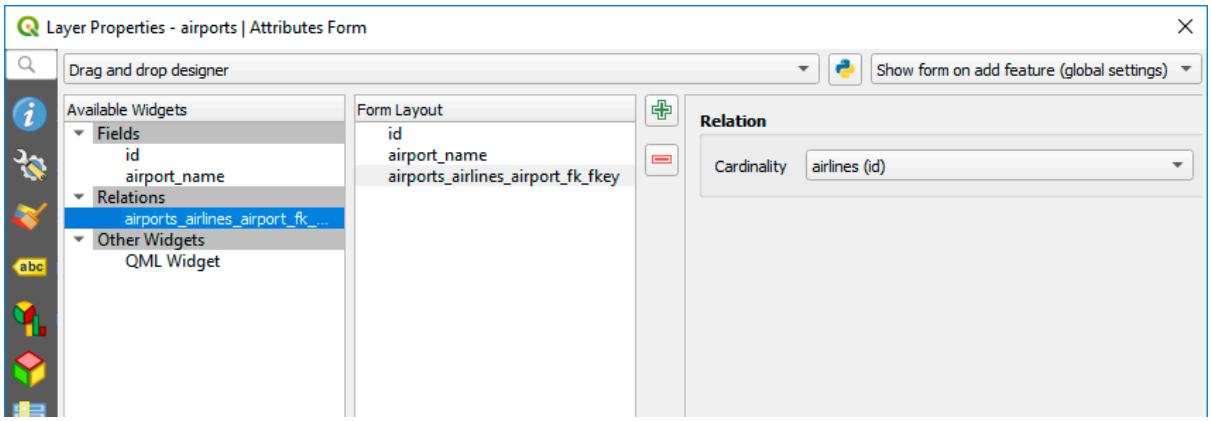


Figura12.113: Definir a cardinalidade da relação

Agora você pode associar um aeroporto com uma companhia aérea (ou uma companhia aérea com um aeroporto) utilizando *Adicionar feição filha* ou *Link existente de feição filha* nas subformulários. Um registro será automaticamente inserido na tabela `aeroportos_companhiaAerea`.

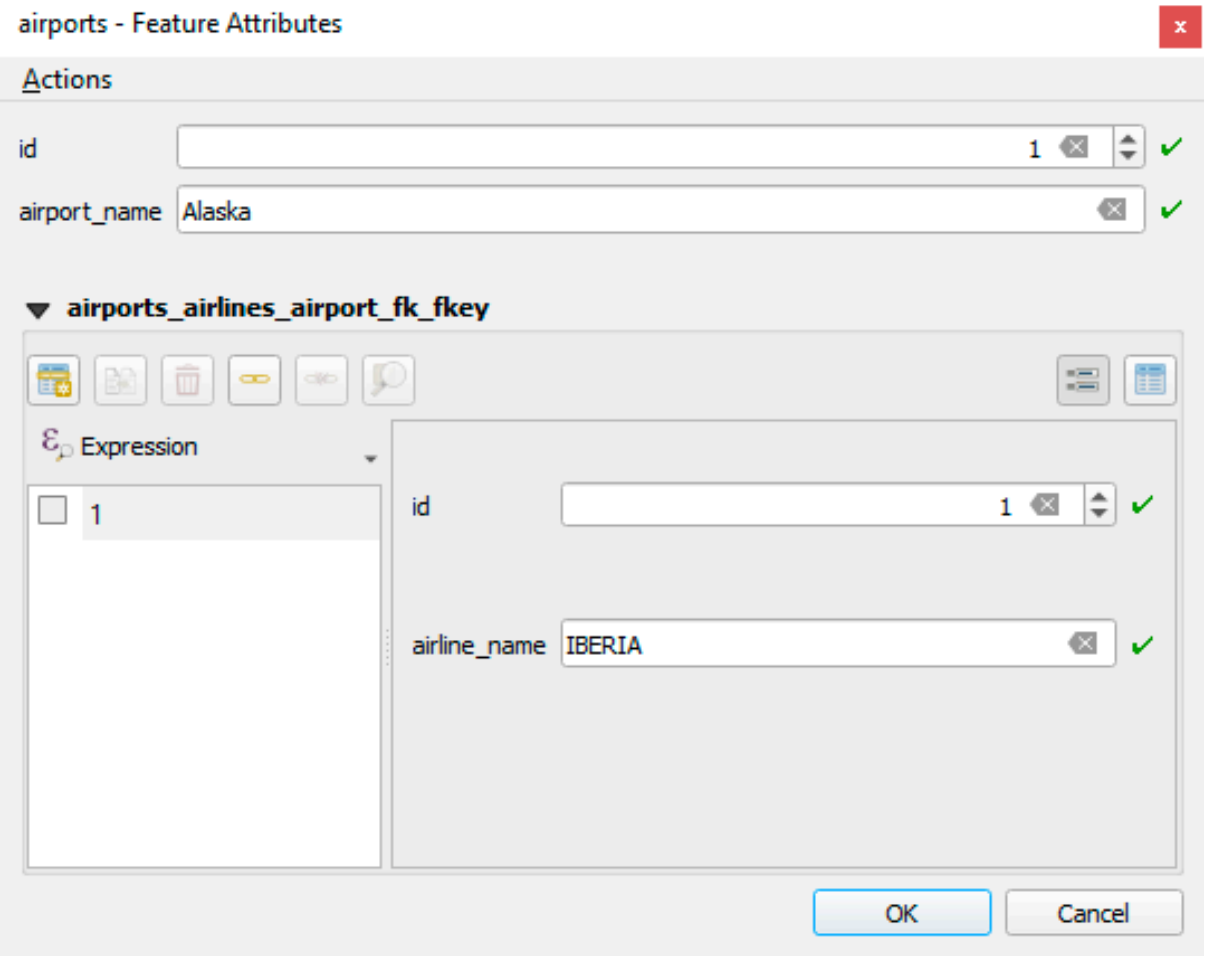


Figura12.114: Relação N-M entre aeroportos e companhias aéreas

Nota

Usando cardinalidade **relação muitos para um**

Sometimes hiding the pivot table in an N-M relationship is not desirable. Mainly because there are attributes in the relationship that can only have values when a relationship is established. If your tables have a geometry field, it could be interesting to activate the *On map identification* option (*Layer Properties* ► *Attributes Form* ► *Available widgets* ► *Fields*) for the foreign key fields in the pivot table.

Nota

Chave primária da tabela dinâmica

Avoid using multiple fields in the primary key in a pivot table. QGIS assumes a single primary key so a constraint like `constraint airports_airlines_pkey primary key (airport_fk, airline_fk)` will not work.

Polymorphic relations

The purpose

Polymorphic relations are special case of 1-N relations, where a single referencing (document) layer contains the features for multiple referenced layers. This differs from normal relations which require different referencing layer for each referenced layer. A single referencing (document) layer is achieved by adding an adiditonal `layer_field` column in the referencing (document) layer that stores information to identify the referenced layer. In its most simple form, the referencing (document) layer will just insert the layer name of the referenced layer into this field.

Para ser mais preciso, uma relação polimórfica é um conjunto de relações normais com a mesma camada de referência, mas com a camada referenciada dinamicamente definida. O ajuste polimórfico da camada é resolvido usando uma expressão que tem que combinar algumas propriedades da camada referenciada como o nome da tabela, id da camada, nome da camada.

Imagine we are going to the park and want to take pictures of different species of `plants` and `animals` we see there. Each plant or animal has multiple pictures associated with it, so if we use the normal 1:N relations to store pictures, we would need two separate tables, `animal_images` and `plant_images`. This might not be a problem for 2 tables, but imagine if we want to take separate pictures for mushrooms, birds etc.

Polymorphic relations solve this problem as all the referencing features are stored in the same table `documents`. For each feature the referenced layer is stored in the `referenced_layer` field and the referenced feature id in the `referenced_id` field.

Definindo relações polimórficas

First, let QGIS know about the polymorphic relations between the layers. This is done in *Project* ► *Properties*….

Open the *Relations* tab and click on the little down arrow next to the *Add Relation* button, so you can select the *Add Polymorphic Relation* option from the newly appeared dropdown.

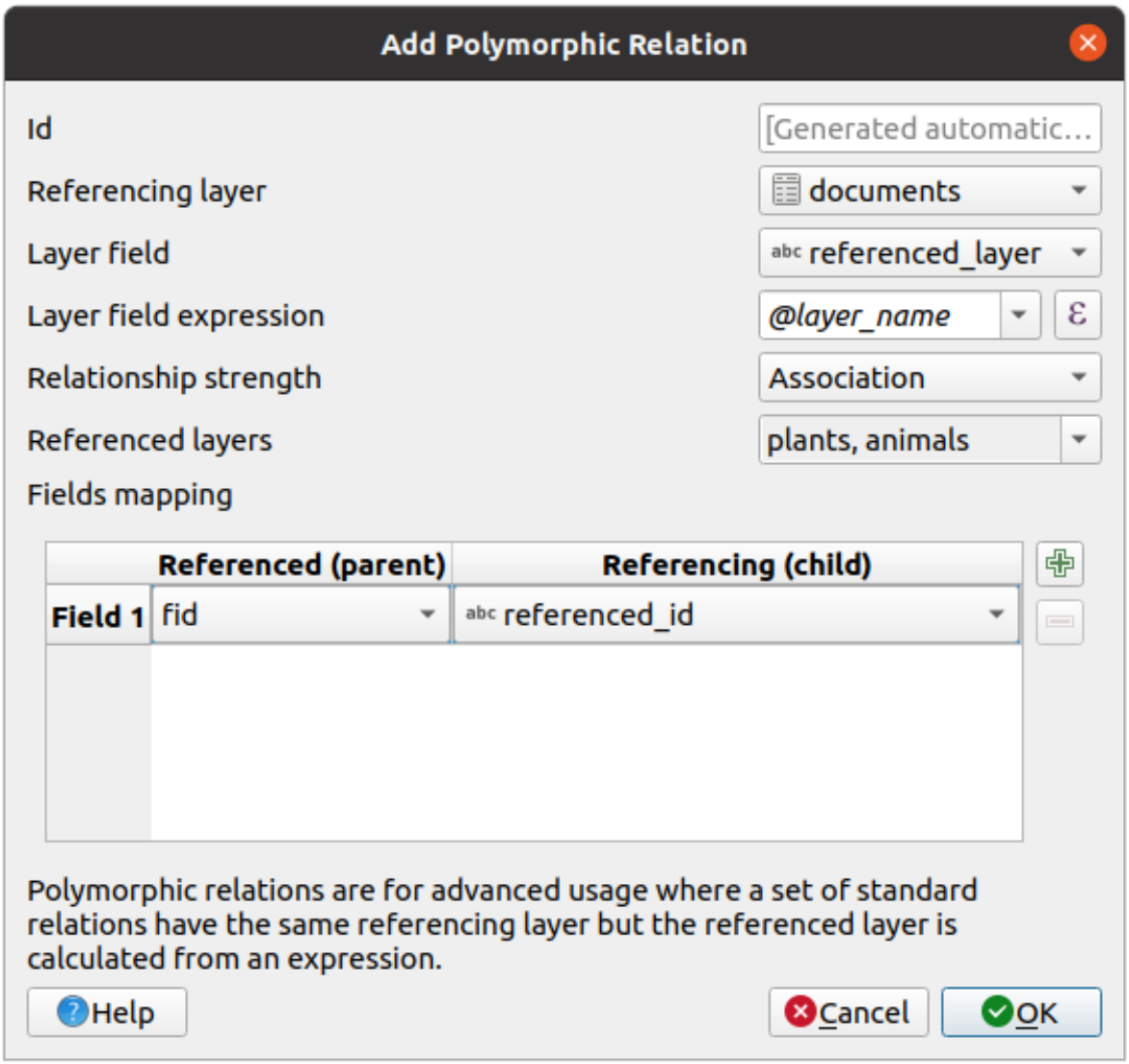


Figura12.115: Adicionando uma relação polimórfica utilizando “documentos” como camadas de referência e “animais” e “plantas” como camadas de referência.

- **Id** will be used for internal purposes and has to be unique. You may need it to build *custom forms*. If you leave it empty, one will be generated for you but you can assign one yourself to get one that is easier to handle
- **Referencing Layer (Child)** also considered as child layer, is the one with the foreign key field on it. In our case, this is the `documents` layer. For this layer you need to add a referencing field which points to the other layer, so this is `referenced_id`.

> **Nota**
>
> Sometimes, you need more than a single field to uniquely identify features in a layer. Creating a relation with such a layer requires a **composite key**, i.e. more than a single pair of matching fields. Use the Add new field pair as part of a composite foreign key button to add as many pairs as necessary.

- **Layer Field** is the field in the referencing table that stores the result of the evaluated layer expression which is the referencing table that this feature belongs to. In our example, this would be the `referenced_layer` field.
- **Layer expression** evaluates to a unique identifier of the layer. This can be the layer name `@layer_name`,

the layer id `@layer_id`, the layer's table name `decode_uri(@layer, 'table')` or anything that can uniquely identifies a layer.

- **Relationship strength** sets the strength of the generated relations between the parent and the child layer. The default *Association* type means that the parent layer is *simply* linked to the child one while the *Composition* type allows you to duplicate also the child features when duplicating the parent ones and on deleting a feature the children are deleted as well, resulting in cascade over all levels (means children of children of… are deleted as well).
- **Referenced Layers** also considered as parent layers, are those with the primary key, pointed to, so here they would be `plants` and `animals` layers. You need to define the primary key of the referenced layers from the dropdown, so it is `fid`. Note that the definition of a valid primary key requires all the referenced layers to have a field with that name. If there is no such field you cannot save a polymorphic relation.

Uma vez adicionada, a relação polimórfica pode ser editada através da entrada do menu *Editar relação polimórfica*.

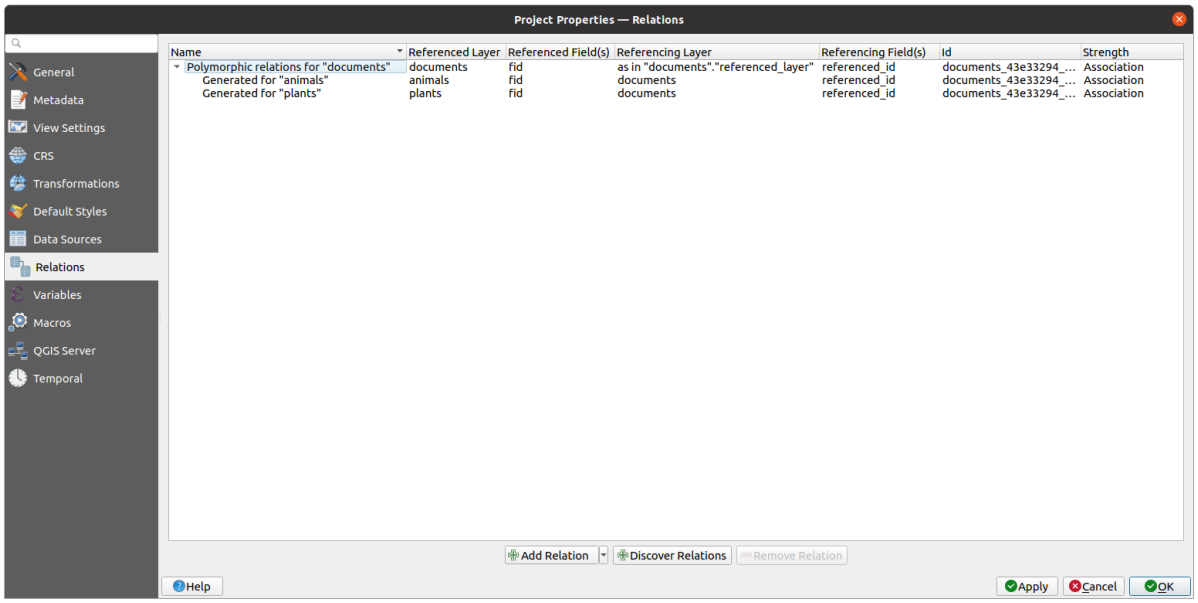


Figura12.116: Preview of the newly created polymorphic relation and its child relations for animals and plants.

O exemplo acima utiliza o seguinte esquema de banco de dados:

```
CREATE SCHEMA park;

CREATE TABLE park.animals
(
   fid serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   animal_species text NOT NULL,
   CONSTRAINT animals_pkey PRIMARY KEY (fid)
);

CREATE INDEX animals_geom_idx ON park.animals USING gist (geom);

CREATE TABLE park.plants
(
   fid serial NOT NULL,
   geom geometry(Point, 4326) NOT NULL,
   plant_species text NOT NULL,
   CONSTRAINT plants_pkey PRIMARY KEY (fid)
);
```

(continua na próxima página)

(continuação da página anterior)

```
CREATE INDEX plants_geom_idx ON park.plants USING gist (geom);

CREATE TABLE park.documents
(
   fid serial NOT NULL,
   referenced_layer text NOT NULL,
   referenced_id integer NOT NULL,
   image_filename text NOT NULL,
   CONSTRAINT documents_pkey PRIMARY KEY (fid)
);
```

Once relations are established, the *Identify Features* result panel will display related features, allowing you to view associated data directly.

Dataset-stored Relationships

Other than the above mentioned relationships that are created in the project properties and apply only in the active project, QGIS can also read database-level relations, i.e., connections defined between layers in the dataset. This is supported for database providers such as PostgreSQL, SpatiaLite, GeoPackage, ESRI FileGeodatabase. When such parent and child layers are loaded in a project, QGIS can automatically detect the relation settings and fill appropriately the *Project* ► *Project properties* ► *Relations* tab. This can also be triggered using the *Discover relations* button.

For some data types (e.g., GeoPackage, SpatiaLite or ESRI FileGeodatabase), QGIS also allows to create and edit the relations between the tables in the database. This can be managed through the *Browser Panel*. From the *Browser panel*, right-click on the database entry, select *New relationship…*, and in the new dialog, fill the following options:

- *Name*: set the name of the new relationship.
- *Cardinality*: define the cardinality of the relationship.
- *Strength*: specify the strength of the relationship.
- Under the *Tables* group, set your *Parent* and *Child* tables.
- Under the *Fields* group, set your *Parent field* and *Child field*.
- Under the *Advanced* section, depending on the database type, you can set:
  - *Related table type*:
    - *features*: for standard feature tables.
    - *media*: for tables storing media files.
    - *simple_attributes*: for tables with simple attribute data.
    - *attributes*: for tables with complex attribute data.
    - *tiles*: for tables storing tile data.
  - *Forward label*: a name to uniquely identify the relationship when navigating from the parent table to the child table.
  - *Reverse label*: a name to uniquely identify the relationship when navigating from the child table to the parent table.

Applying the dialog will add a *Relationships* node under the database entry, with the new relationship as child item.

Editing or deleting existing relationships can be done by right-clicking on the relationship in the Browser Panel.

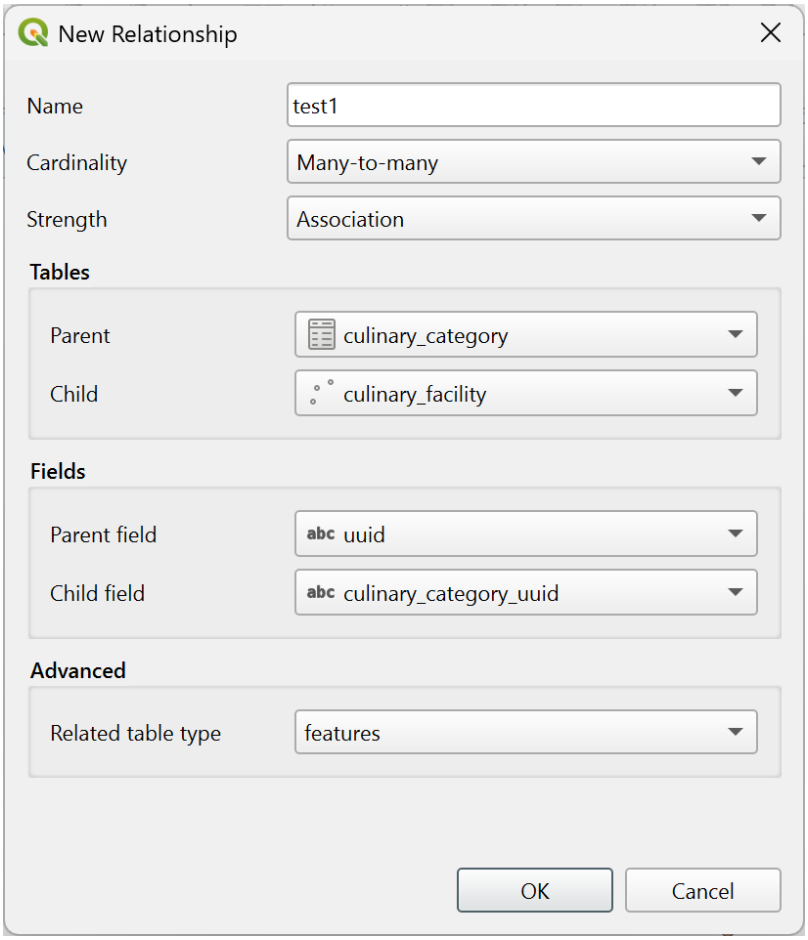


Figura12.117: Managing dataset-stored relationships in the Browser Panel

12.4.3 Storing and fetching an external resource

A field may target a resource stored on an external storage system. Attribute forms can be configured so they act as a client to an external storage system in order to store and fetch those resources, on users demand, directly from the forms.

Configuring an external storage

In order to setup an external storage, you have to first configure it from the vector *attribute form properties* and select the *Attachment* widget.

Figura12.118: Editing a WebDAV external storage for a given field

From the *Attachment* widget, you have to first select the *Storage type*:

- *Select Existing File*: The target URL already exists. When you select a resource, no store operation is achieved, the attribute is simply updated with the URL.
- *Simple Copy*: Stores a copy of the resource on a file disk destination (which could be a local or network shared file system) and the attribute is updated with the path to the copy.
- *WebDAV Storage*: The resource is pushed to a HTTP server supporting the WebDAV protocol and the attribute is updated with its URL. Nextcloud, Pydio or other file hosting software support this protocol.
- *AWS S3*: The resource is pushed to a server supporting AWS Simple Storage Service protocol and the attribute is updated with its URL. Amazon Web Service and MinIO hosting software support this protocol.

Then, you have to set up the *Store URL* parameter, which provides the URL to be used when a new resource needs to be stored. It's possible to set up an expression using the *data defined override widget* in order to have specific values according to feature attributes.

The variable **@selected_file_path** could be used in that expression and represent the absolute file path of the user selected file (using the file selector or drag'n drop).

Nota

Using the **WebDAV** or **AWS S3** external storage, if the URL ends with a "/", it is considered as a folder and the selected file name will be appended to get the final URL.

If the external storage system needs to, it's possible to configure an *authentication*.

Nota

To use the **AWS S3** external storage, you must use an **AWS S3** authentication type.

Using an external storage

Once configured, you can select a local file using the button … when editing a feature's attribute. Depending on the configured *storage type*, the file will be stored on the external storage system (except if *Select existing file* has been selected) and the field will be updated with the new resource URL.

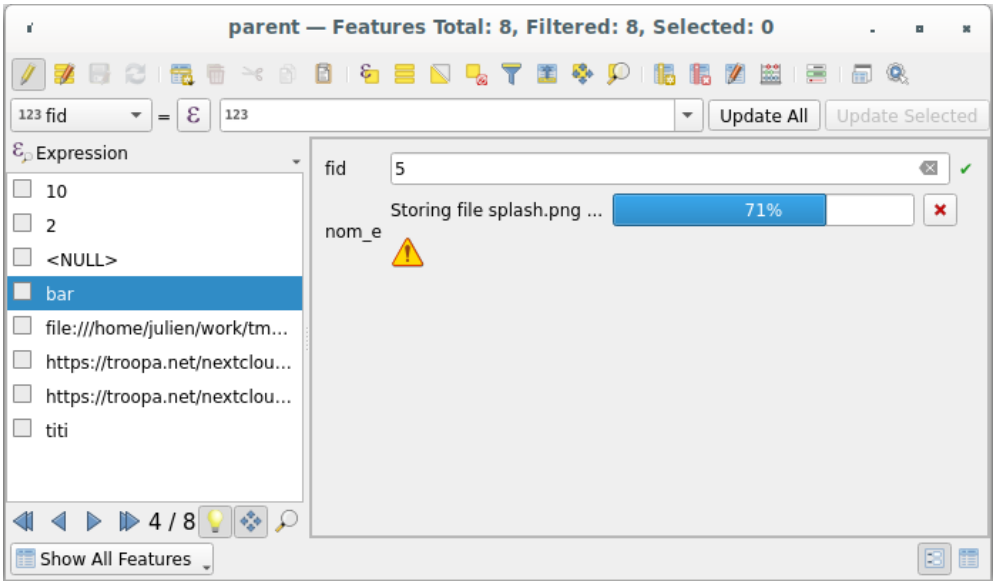


Figura12.119: Storing a file to a WebDAV external storage

Nota

User can also achieve the same result if he drags and drops a file on the whole attachment widget.

Use the Cancel button to abort the storing process. It's possible to configure a viewer using the *Integrated document viewer* so the resource will be automatically fetched from the external storage system and displayed directly below the URL. The above icon indicates that the resource cannot be fetched from the external storage system. In that case, more details might appear in the *Log Messages Panel*.

CAPÍTULO 13

Trabalhando com Dados Raster

13.1 Caixa de Diálogo de Propriedades do Raster

Raster data is made up of pixels (or cells), and each pixel has a value. It is commonly used to store various types of data, including:

- Imagery, such as satellite images, digital aerial photographs, scanned maps
- Elevation data, such as digital elevation models (DEMs), digital terrain models (DTMs)
- Other types of data, such as land cover, soil types, rainfall and many others.

Raster data can be stored in several supported formats, including GeoTIFF, ERDAS Imagine, ArcInfo ASCII GRID, PostgreSQL Raster and others. See more at *Abrir Dados*.

Para visualizar e definir as propriedades de uma camada raster, clique duas vezes no nome da camada na legenda do mapa ou clique com o botão direito do mouse no nome da camada e escolha *Propriedades* no menu de contexto. Isso abrirá a caixa de diálogo *Propriedades da Camada Raster*.

Existem várias guias na caixa de diálogo:

Information	*Source*	*Symbology* ([1])
Transparency ([1])	*Labels* ([1])	*Histogram* ([1])
Rendering	*Temporal*	*Pyramids*
Elevation	*Metadata*	*Legend*
Display	*Attribute Tables*	*QGIS Server*
External plugins ([2]) tabs		

[1] Also available in the *Layer styling panel*
[2] *External plugins* you install can optionally add tabs to this dialog. Those are not presented in this document. Refer to their documentation.

> **Dica**
>
> **Renderização de atualização ao vivo**
>
> O *Painel de Estilo de Camada* fornece alguns dos recursos comuns da caixa de diálogo Propriedades da Camada e é um bom widget sem modelo que você pode usar para acelerar a configuração dos estilos de camada e visualizar suas alterações na tela do mapa.

> **Nota**
>
> Como as propriedades (simbologia, rótulo, ações, valores padrão, formulários…) das camadas incorporadas (veja *Incorporando camadas de projetos externos*) são extraídas do arquivo original do projeto, e para evitar alterações que possam quebrar esse comportamento, a caixa de diaĺogo das propriedades da camada fica indisponível para essas camadas.

13.1.1 Propriedades da informação

A guia *Informações* é somente leitura e representa um local interessante para obter rapidamente informações e metadados resumidos da camada atual. As informações fornecidas são:

- general such as name in the project, source path, list of auxiliary files, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties include *Identify/format*, which influences how the results from using the *Identify features* tool over a raster layer are formatted. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- based on the provider of the layer: extent, width and height, data type, GDAL driver, bands statistics
- o Sistema de Referência de Coordenadas: nome, unidades, método, precisão, referência (ou seja, se é estático ou dinâmico)
- read from layer properties: data type, extent, width/height, compression, pixel size, statistics on bands, number of columns, rows and no-data values of the raster…
- picked from the *filled metadata*: access, extents, links, contacts, history…

13.1.2 Propriedades de origem

A guia *Fonte* exibe informações básicas sobre o raster selecionado, incluindo:

- o *Nome da Camada* para ser mostrado no *Painel de Camadas*;
- the *Coordinate Reference System*: Displays the layer's *Coordinate Reference System (CRS)*. You can change the layer's CRS, by selecting a recently used one in the drop-down list or clicking on the Select CRS button (see *Seletor do Sistemas de Referência de Coordenadas*). Use this process only if the layer CRS is a wrong or not specified. If you wish to reproject your data, use a reprojection algorithm from Processing or *Save it as new dataset*.
- Depending on the data provider, a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer:
  - When the layer is stored as file on disk, edit the path shown in the text box or press … Browse to select another file on the disk
  - When the layer is provided by an ArcGIS MapServer service, it is possible to modify its *authentication settings*, while keeping unchanged the details for connecting to the service.

– For an XYZ remote layer, it is possible to modify at the layer level any of its *connection details* (URL, authentication settings, zoom levels, resolution, …). Modifications are applied to the layer without altering the original connection settings.

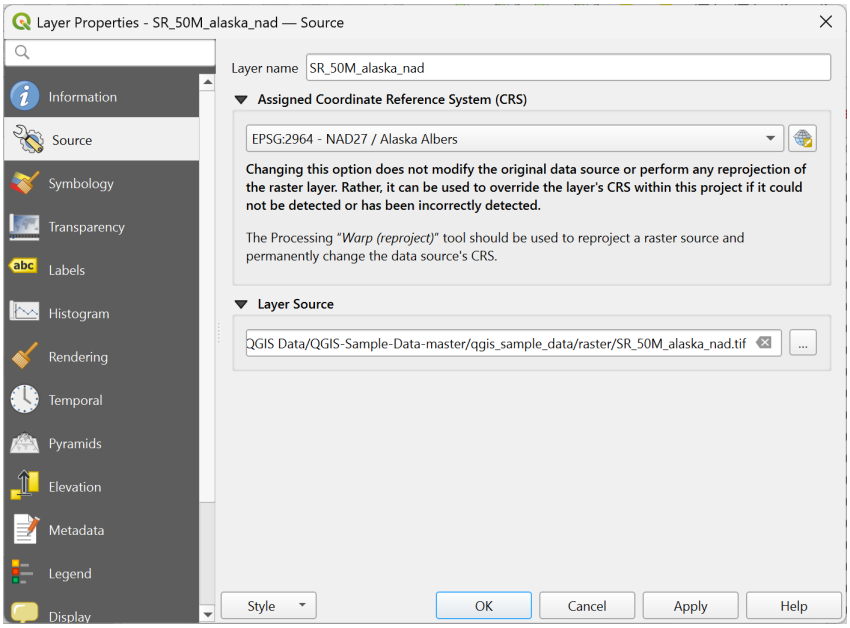


Figura13.1: Raster Layer Properties - Source Dialog

13.1.3 Propriedades da simbologia

A aba de simbologia da camada raster é composta de três seções diferentes:

- The *Band rendering* where you can control the renderer type to use
- The *Layer rendering* to apply effects on rendered data
- The *Resampling* methods to optimize rendering on map

Representar a banda

QGIS offers many different *Render types*. The choice of renderer depends on the data type and the information you'd like to highlight.

1. *Multiband color* - if the file comes with several bands (e.g. a satellite image with several bands).
2. *Paletted/Unique values* - for single band files that come with an indexed palette (e.g. a digital topographic map) or for general use of palettes for rendering raster layers.
3. *Singleband gray* - (one band of) the image will be rendered as gray. QGIS will choose this renderer if the file is neither multiband nor paletted (e.g. a shaded relief map).
4. *Singleband pseudocolor* - this renderer can be used for files with a continuous palette or color map (e.g. an elevation map).
5. *Single color* - the raster layer will be rendered with a single color.
6. *Hillshade* - Creates hillshade from a band.
7. *Contours* - Generates contours on the fly for a source raster band.

Cor multibanda

With the multiband color renderer, three selected bands from the image will be used as the red, green or blue component of the color image. QGIS automatically fetches *Min* and *Max* values for each band of the raster and scales the coloring accordingly. You can control the value ranges in the *Min/Max Value Settings* section.

A *Contrast enhancement* method can be applied to the values: 'No enhancement', 'Stretch to MinMax', 'Stretch and clip to MinMax' and 'Clip to min max'.

Nota

Melhora do contraste

When adding GRASS rasters, the option *Contrast enhancement* will always be set automatically to *stretch to min max*, even if this is set to another value in the QGIS general options.

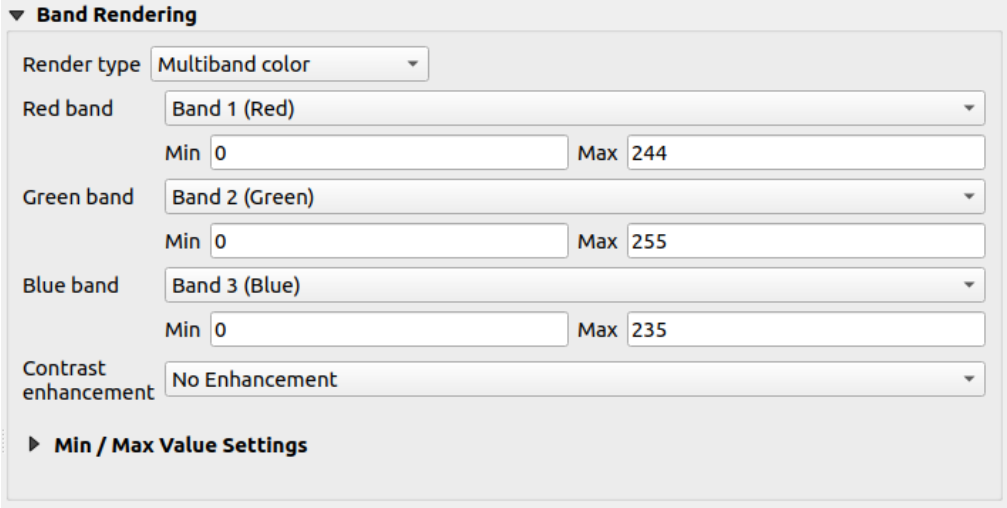


Figura13.2: Raster Symbology - Multiband color rendering

Dica

Visualizando uma única banda do Raster Multibanda

If you want to view a single band of a multiband image (for example, Red), you might think you would set the Green and Blue bands to *Not Set*. But the preferred way of doing this is to set the image type to *Singleband gray*, and then select Red as the *Gray band* to use.

Paletted/Unique values

This is the standard render option for singleband files that include a color table, where a certain color is assigned to each pixel value. In that case, the palette is rendered automatically.

Pode ser usado para todos os tipos de bandas matriciais, atribuindo uma cor a cada valor matricial exclusivo.

If you want to change a color, just double-click on the color and the *Select color* dialog appears.

Também é possível atribuir etiquetas às cores. A etiqueta aparecerá então na legenda da camada matriz.

Right-clicking over selected rows in the color table shows a contextual menu to:

- *Change Color…* for the selection

- *Change Opacity…* for the selection
- *Change Label…* for the selection

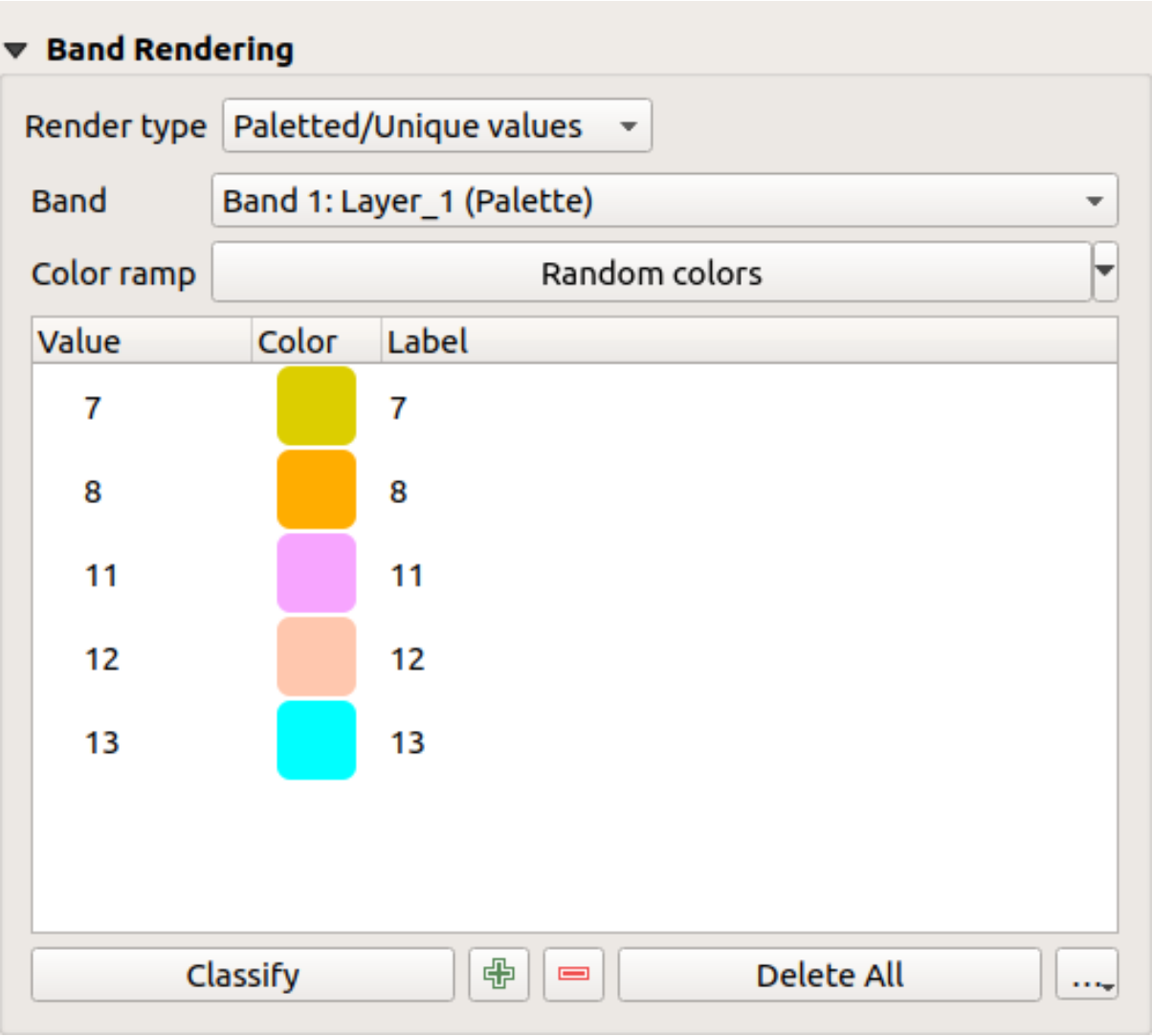


Figura13.3: Raster Symbology - Paletted unique value rendering

The pulldown menu, that opens when clicking the … (Advanced options) button below the color map to the right, offers color map loading (*Load Color Map from File…*) and exporting (*Export Color Map to File…*), and loading of classes (*Load Classes from Layer*).

Singleband gray

This renderer allows you to render a layer using only one band with a *Color gradient*: 'Black to white' or 'White to black'. You can change the range of values to color (*Min* and *Max*) in the *Min/Max Value Settings*.

A *Contrast enhancement* method can be applied to the values: 'No enhancement', 'Stretch to MinMax', 'Stretch and clip to MinMax' and 'Clip to min max'.

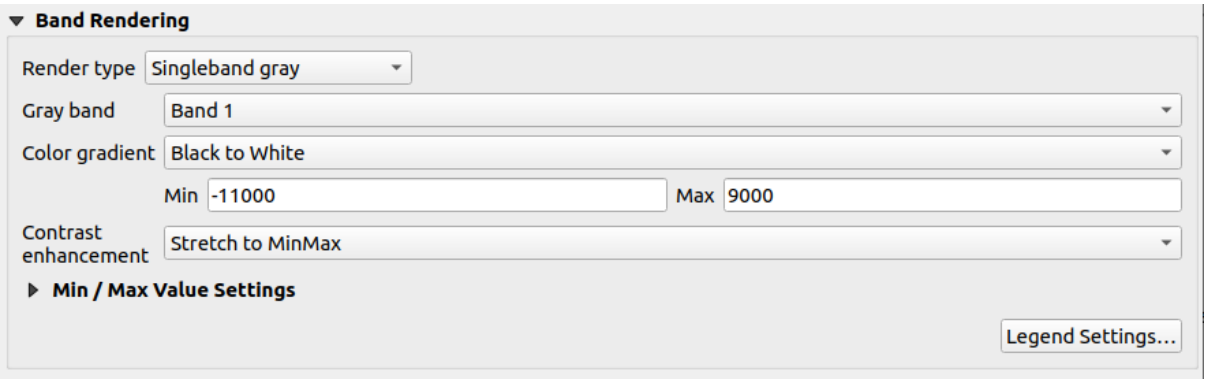


Figura13.4: Raster Symbology - Singleband gray rendering

Pixels are assigned a color based on the selected color gradient and the layer's legend (in the *Layers* panel and the layout *legend item*) is displayed using a continuous color ramp. Press *Legend settings...* if you wish to tweak the settings. More details at *Customize raster legend*.

Singleband pseudocolor

Esta é uma opção de renderização para arquivos de banda única que incluem uma paleta contínua. Você também pode criar mapas de cores para uma banda de uma matriz multibanda.

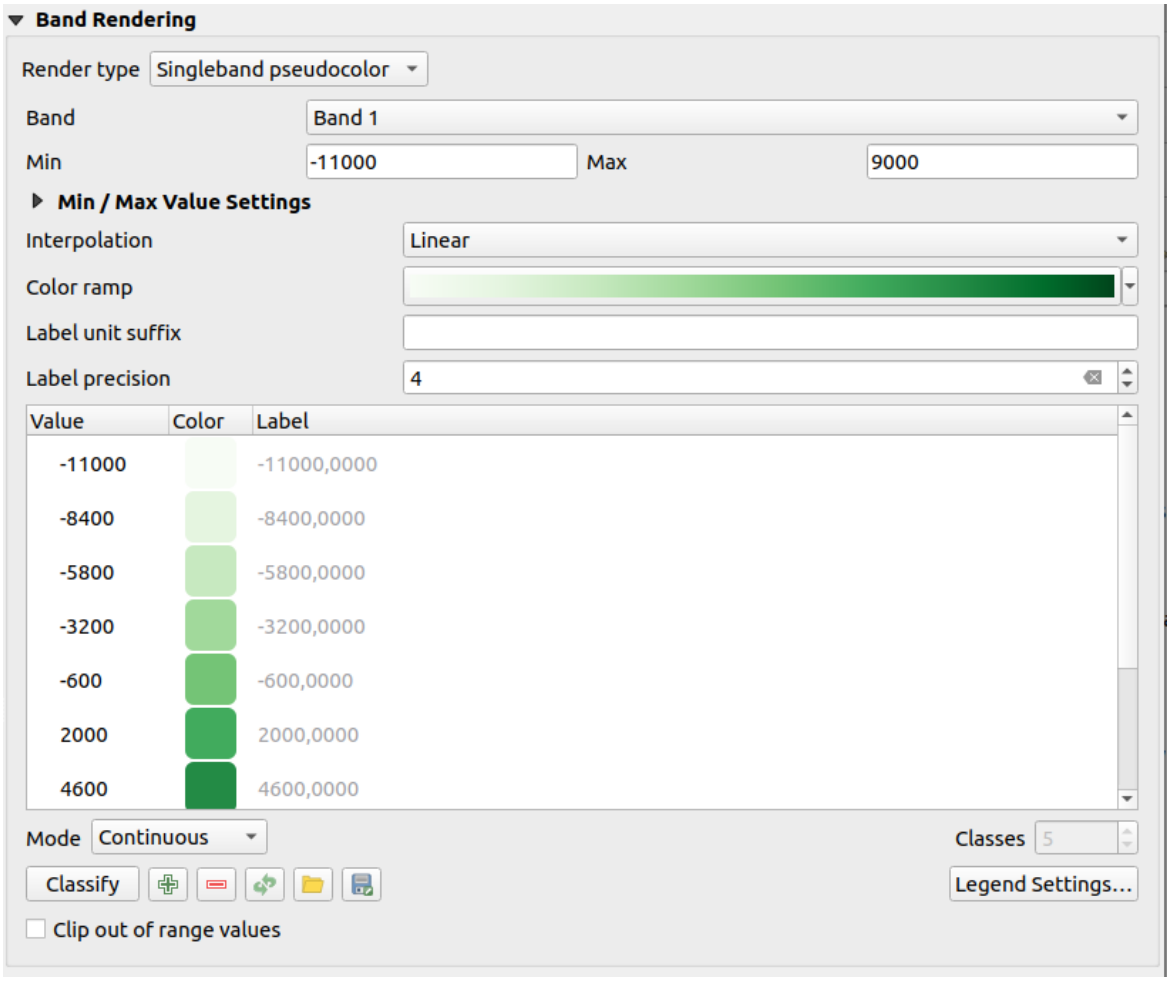


Figura13.5: Raster Symbology - Singleband pseudocolor rendering

Using a *Band* of the layer and a *values range*, you can now interpolate and assign representation color to pixels within classes. More at *Color ramp shader classification*.

Pixels are assigned a color based on the selected color ramp and the layer's legend (in the *Layers* panel and the layout *legend item*) is displayed using a continuous color ramp. Press *Legend settings…* if you wish to tweak the settings or instead use a legend with separated classes (and colors). More details at *Customize raster legend*.

Single color

This renderer allows you to render a raster layer using *Single color*. This type of renderer is useful when you want to display a raster layer uniformly, without any variation in color based on pixel values.

The single color renderer can be used with both single-band and multiband raster layers. When used with multiband rasters, you can select which band to apply the single color to, effectively displaying that specific band uniformly across the entire layer.

Figura13.6: Raster Symbology - Single color rendering

Sombreamento

Render a band of the raster layer using hillshading.

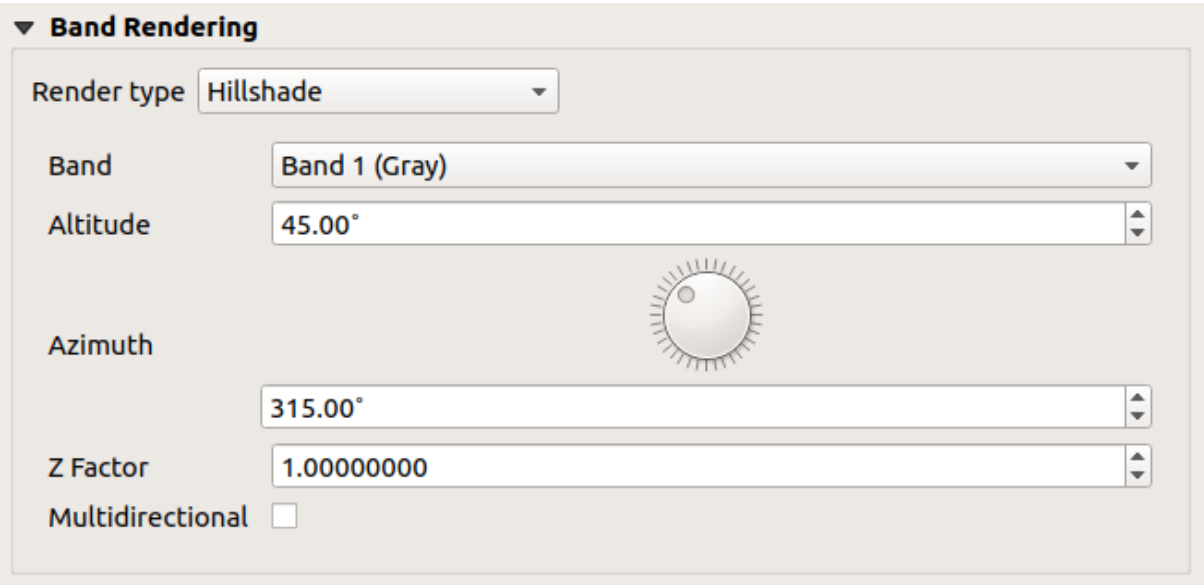


Figura13.7: Raster Symbology - Hillshade rendering

Opções:

- *Band*: The raster band to use.
- *Altitude*: The elevation angle of the light source (default is `45°`).

- *Azimuth*: The azimuth of the light source (default is `315°`).
- *Z Factor*: Scaling factor for the values of the raster band (default is `1`).
- *Multidirectional*: Specify if multidirectional hillshading is to be used (default is `off`).

Contornos

This renderer draws contour lines that are calculated on the fly from the source raster band.

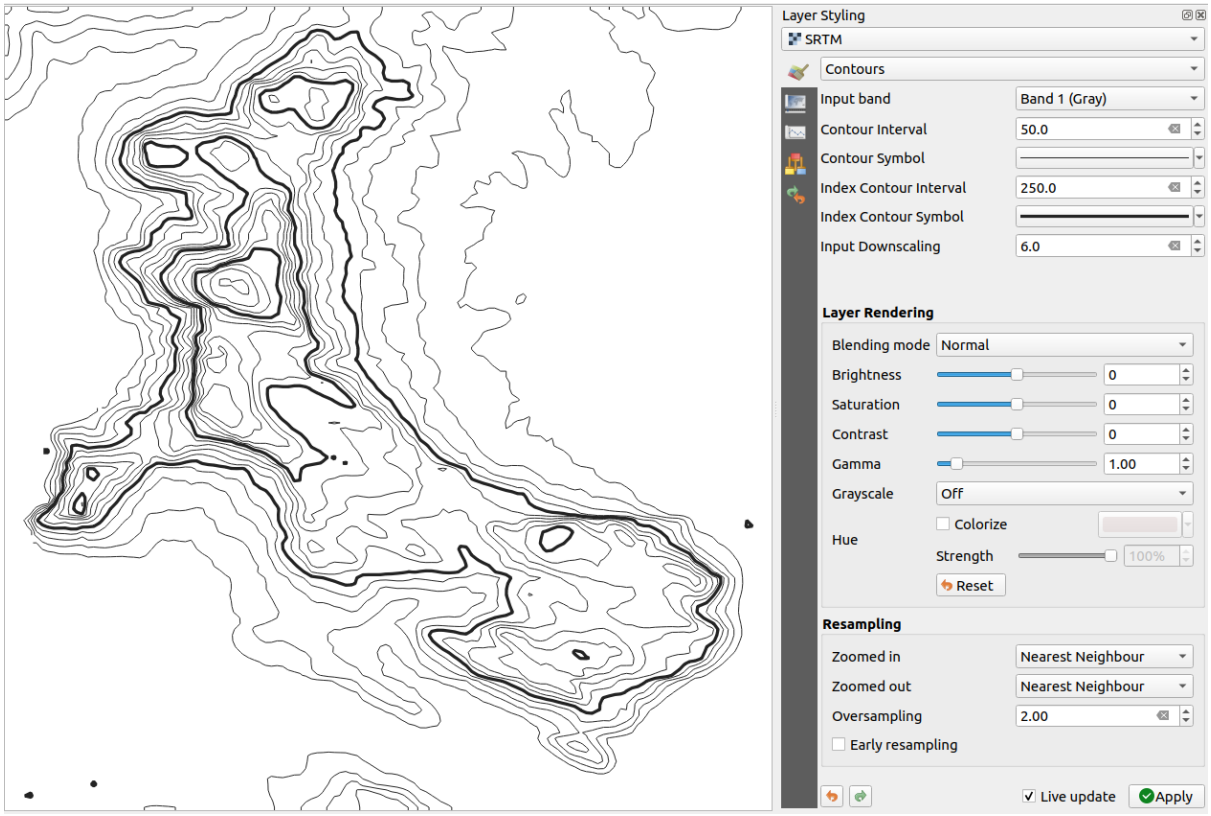


Figura13.8: Raster Symbology - Contours rendering

Opções:

- *Input band*: the raster band to use.
- *Contour interval*: the distance between two consecutive contour lines
- *Contour symbol*: the *symbol* to apply to the common contour lines.
- *Index contour interval*: the distance between two consecutive **index contours**, that is the lines shown in a distinctive manner for ease of identification, being commonly printed more heavily than other contour lines and generally labeled with a value along its course.
- *Index contour symbol*: the symbol to apply to the index contour lines
- *Input downscaling*: Indicates by how much the renderer will scale down the request to the data provider (default is `4.0`).

  For example, if you generate contour lines on input raster block with the same size as the output raster block, the generated lines would contain too much detail. This detail can be reduced by the "downscale" factor, requesting lower resolution of the source raster. For a raster block 1000x500 with downscale 10, the renderer will request raster 100x50 from provider. Higher downscale makes contour lines more simplified (at the expense of losing some detail).

Configuração dos valores mínimo e máximo

By default, QGIS reports the *Min* and *Max* values of the band(s) of the raster. A few very low and/or high values can have a negative impact on the rendering of the raster. The *Min/Max Value Settings* frame helps you control the rendering.

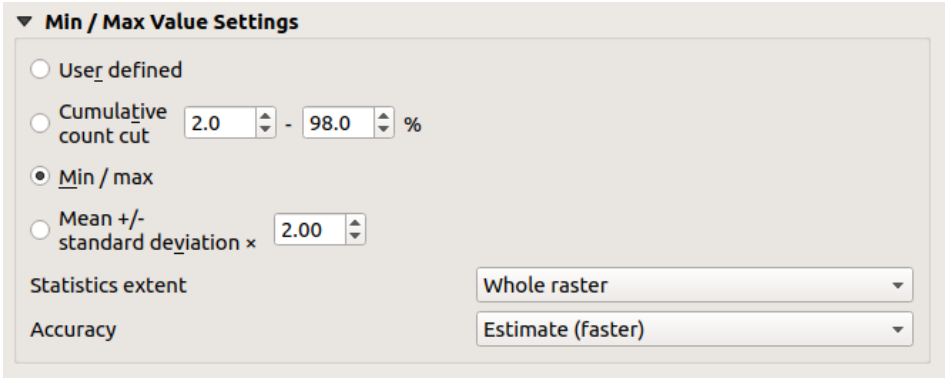


Figura13.9: Simbologia Raster - Configurações de Valor Mínimo e Máximo

As opções disponíveis são:

- *User defined*: The default *Min* and *Max* values of the band(s) can be overridden
- *Cumulative count cut*: Removes outliers. The standard range of values is `2%` to `98%`, but it can be adapted manually.
- *Min / max*: Uses the whole range of values in the image band.
- *Mean +/- standard deviation x*: Creates a color table that only considers values within the standard deviation or within multiple standard deviations. This is useful when you have one or two cells with abnormally high values in a raster layer that impact the rendering of the raster negatively.

Os cálculos dos valores mínimo e máximo das bandas são feitos com base em:

- *Statistics extent*: it can be *Whole raster*, *Current canvas* or *Updated canvas*. *Updated canvas* means that min/max values used for the rendering will change with the canvas extent (dynamic stretching).
- *Accuracy*, which can be either *Estimate (faster)* or *Actual (slower)*.

Nota

For some settings, you may need to press the *Apply* button of the layer properties dialog in order to display the actual min and max values in the widgets.

Color ramp shader classification

This method can be used to classify and represent scalar dataset (raster or mesh contour) based on their values. Given a *color ramp* and a number of classes, it generates intermediate color map entries for class limits. Each color is mapped with a value interpolated from a range of values and according to a classification mode. The scalar dataset elements are then assigned their color based on their class.

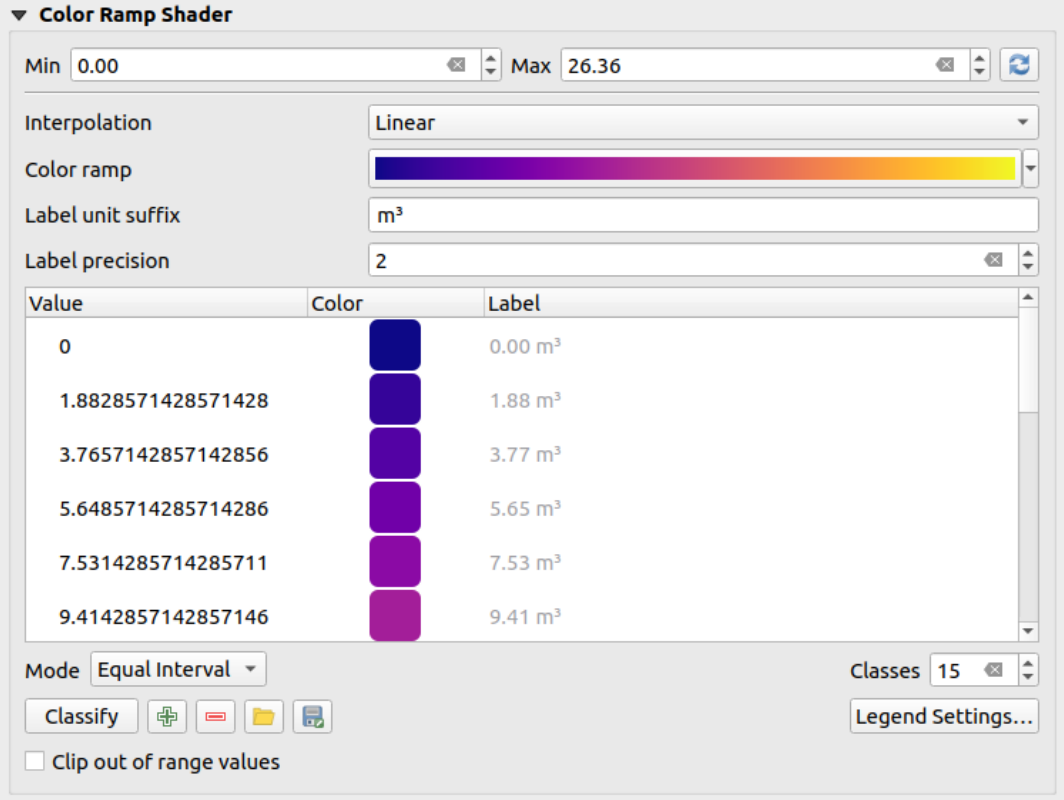


Figura13.10: Classifying a dataset with a color ramp shader

1. A *Min* and *Max* values must be defined and used to interpolate classes bounds. By default QGIS detects them from the dataset but they can be modified.
2. The *Interpolation* entry defines how scalar elements are assigned their color :
   - *Discrete* (a <= symbol appears in the header of the *Value* column): The color is taken from the closest color map entry with equal or higher value
   - *Linear*: The color is linearly interpolated from the color map entries above and below the pixel value, meaning that to each dataset value corresponds a unique color
   - *Exact* (a = symbol appears in the header of the *Value* column): Only pixels with value equal to a color map entry are applied a color; others are not rendered.
3. The *Color ramp* widget helps you select the color ramp to assign to the dataset. As usual with *this widget*, you can create a new one and edit or save the currently selected one. The name of the color ramp will be saved in the configuration.
4. The *Label unit suffix* adds a label after the value in the legend, and the *Label precision* controls the number of decimals to display.
5. The classification *Mode* helps you define how values are distributed across the classes:
   - *Equal interval*: Provided the *Number of classes*, limits values are defined so that the classes all have the same magnitude.
   - *Continuous*: Classes number and color are fetched from the color ramp stops; limits values are set following stops distribution in the color ramp.
   - *Quantile*: Provided the *Number of classes*, limits values are defined so that the classes have the same number of elements. Not available with *mesh layers*.
6. You can then *Classify* or tweak the classes:
   - The button Add values manually adds a value to the table.

- The button Remove selected row deletes selected values from the table.
- Um duplo clique na coluna *Valor* permite modificar o valor da classe.
- Double clicking in the *Color* column opens the dialog *Change color*, where you can select a color to apply for that value.
- Double clicking in the *Label* column to modify the label of the class, but this value won't be displayed when you use the identify feature tool.
- Right-clicking over selected rows in the color table shows a contextual menu to *Change Color…* and *Change Opacity…* for the selection.

You can use the buttons Load color map from file or Export color map to file to load an existing color table or to save the color table for later use.

7. Com *Interpolação* linear, você também pode configurar:

   - *Clip out of range values*: By default, the linear method assigns the first class (respectively the last class) color to values in the dataset that are lower than the set *Min* (respectively greater than the set *Max*) value. Check this setting if you do not want to render those values.
   - *Legend settings*, for display in the *Layers* panel and the layout *legend item*. More details at *Customize raster legend*.

Customize raster legend

When applying a color ramp to a raster or a mesh layer, you may want to display a legend showing the classification. By default, QGIS displays a continuous color ramp with min and max values in the *Layers* panel and the layout *legend item*. This can be customized using the *Legend settings* button in the classification widget.

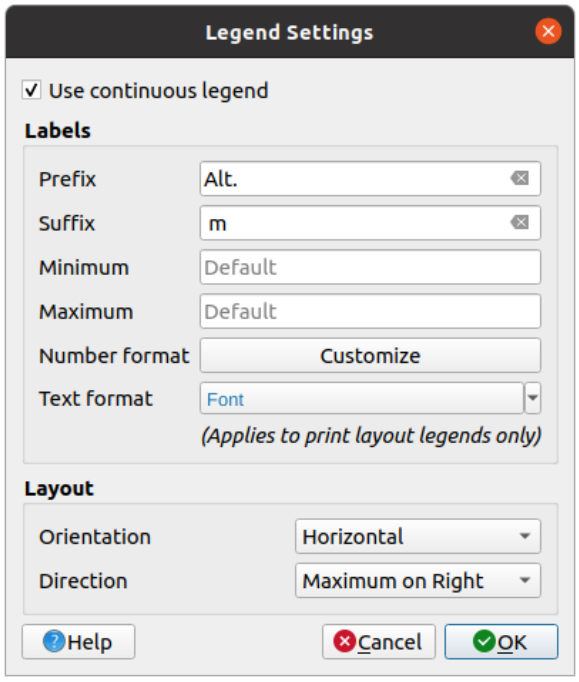


Figura13.11: Modificando uma legenda matricial

In this dialog, you can set whether to *Use continuous legend*: if unchecked, the legend displays separated colors corresponding to the different classes applied. This option is not available for raster *singleband gray* symbology.

Checking the *Use continuous legend* allows you to configure both the labels and layout properties of the legend.

Etiquetas

- Add a *Prefix* and a *Suffix* to the labels
- Modify the *Minimum* and a *Maximum* values to show in the legend
- *Customize* the *Number format*
- *Customize* the *Text format* to use in the print layout legend.

Layout

- Control the *Orientation* of the legend color ramp; it can be **Vertical** or **Horizontal**
- Control the *Direction* of the values depending on the orientation:
  - If vertical, you can display the **Maximum on top** or the **Minimum on top**
  - If horizontal, you can display the **Maximum on right** or the **Minimum on right**

Renderização da camada

Over the symbology type applied to the layer band(s), you can achieve special rendering effects for the whole raster file(s):

- Use one of the blending modes (see *Modos de Mistura*)
- Set custom *Brightness*, *Saturation*, *Gamma* and *Contrast* to colors.
- With the *Invert colors*, the layer is rendered with opposite colors. Handy, for example, to switch out-of-the box OpenStreetMap tiles to dark mode.
- Turn the layer to *Grayscale* option either 'By lightness', 'By luminosity' or 'By average'.
- *Colorize* and adjust the *Strength* of *Hue* in the color table

Press *Reset* to remove any custom changes to the layer rendering.

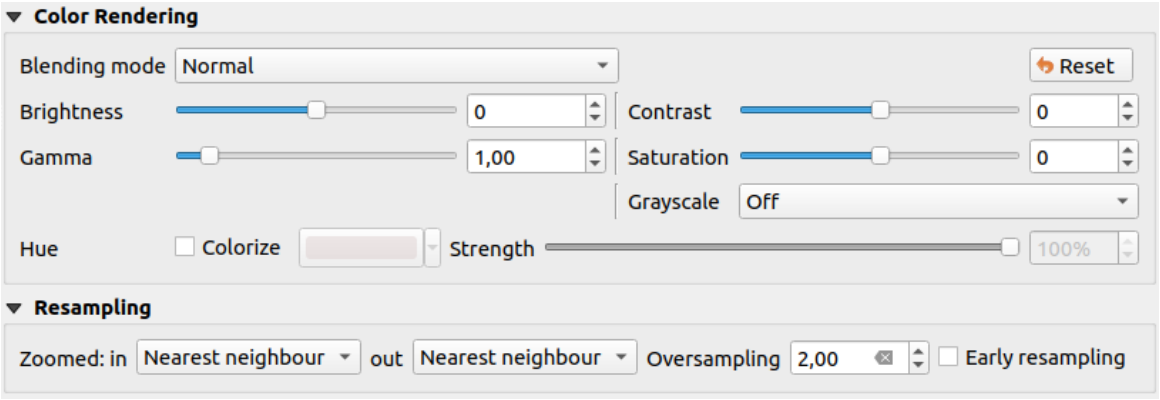


Figura13.12: Raster Symbology - Layer rendering and Resampling settings

Reamostragem

The *Resampling* option has effect when you zoom in and out of an image. Resampling modes can optimize the appearance of the map. They calculate a new gray value matrix through a geometric transformation.

When applying the 'Nearest neighbour' method, the map can get a pixelated structure when zooming in. This appearance can be improved by using the 'Bilinear (2x2 kernel)' or 'Cubic (4x4 kernel)' method, which cause sharp edges to be blurred. The effect is a smoother image. This method can be applied to for instance digital topographic raster maps.

Early resampling: allows to calculate the raster rendering at the provider level where the resolution of the source is known, and ensures a better zoom in rendering with QGIS custom styling. Really convenient for tile rasters loaded using an *interpretation method*.

13.1.4 Propriedades de Transparência

QGIS provides capabilities to set the *Transparency* level of a raster layer.

Use the *Global opacity* slider to set to what extent the underlying layers (if any) should be visible through the current raster layer. This is very useful if you overlay raster layers (e.g., a shaded relief map overlayed by a classified raster map). This will make the look of the map more three dimensional. The opacity of the raster can be data-defined, and vary e.g. depending on the visibility of another layer, by temporal variables, on different pages of an atlas, …

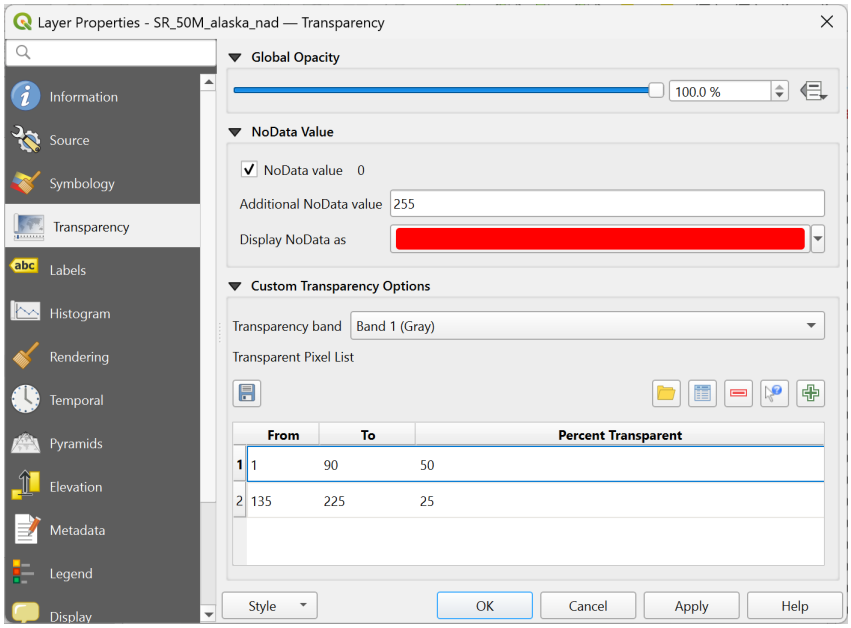


Figura13.13: Transparência do Raster

With *No data value* QGIS reports the original source no data value (if defined) which you can consider as is in the rendering. Additionally, you can enter a raster value that should be treated as an *Additional no data value*. The *Display no data as* color selector allows you to apply a custom color to no data pixels, instead of the default transparent rendering.

An even more flexible way to customize the transparency is available in the *Custom transparency options* section:

- Use *Transparency band* to apply transparency for an entire band.
- Fornecer uma lista de pixels para tornar transparente com os níveis correspondentes de transparência:

  1. Click the Add values manually button. A new row will appear in the pixel list.

2. For single-band based symbology (e.g. DEMs), enter the **From** and **To** values and adjust the **Percent Transparent** to apply.
3. For multiband based symbology (e.g. RGB images) enter the **Red**, **Green** and **Blue** values of the pixel and adjust the **Percent Transparent** to apply. QGIS supports **Tolerance** for pixel values, when defining transparency. This means that pixels with color close to the specified RGB values can also be made transparent. Note that this feature applies only to multiband rasters.
4. Alternatively, you can fetch the pixel values directly from the raster using the Add values from display button. Then enter the transparency value.
5. Repetir as etapas para ajustar mais valores com transparência personalizada.
6. Press the *Apply* button and have a look at the map.

As you can see, it is quite easy to set custom transparency, but it can be quite a lot of work. Therefore, you can use the button Export to file to save your transparency list to a file. The button Import from file loads your transparency settings and applies them to the current raster layer.

13.1.5 Labels Properties

The *Labels* properties provides you with all the needed and appropriate capabilities to configure smart labeling on raster layers. This dialog can also be accessed from the *Layer Styling* panel.

At the top of the dialog, you have:

- a combobox for selecting the appropriate labeling method for the active layer
- the Configure project labeling rules button: helps you control interactions between labels and features across the layers in the project. More details at *Configuring project labeling rules*.
- the Automated placement settings (applies to all layers) button: configure general properties on label placement and conflicts resolution. More details at *Configurando o mecanismo de posicionamento automatizado*.

The first step is to choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer
- *Label with pixel values*: Show labels on the map using a band.

Using the *Label with pixel values* option, the following dialog opens.

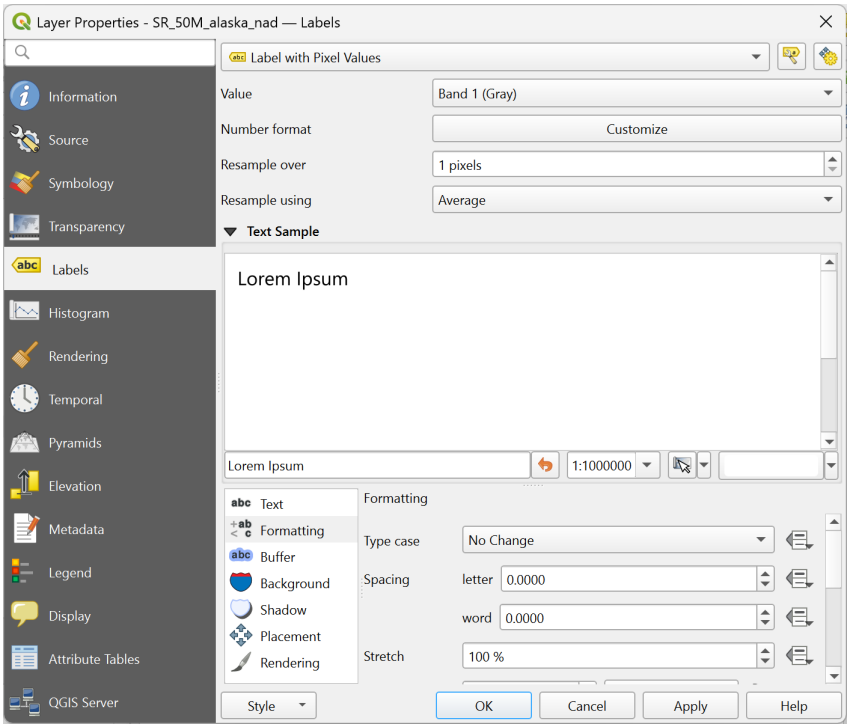


Figura13.14: Raster layer labeling settings

At the top of the dialog:

- A *Value* drop-down list allows you to select the band to take the values from
- Press *Customize* to configure a *proper number formatting*
- By default, the displayed value represents individual pixel band value. With *Resample over*, you can compute the value from the neighbouring pixels (setting `2` means `2*2=4` pixels) using a statistical method set in the *Resample using* widget.

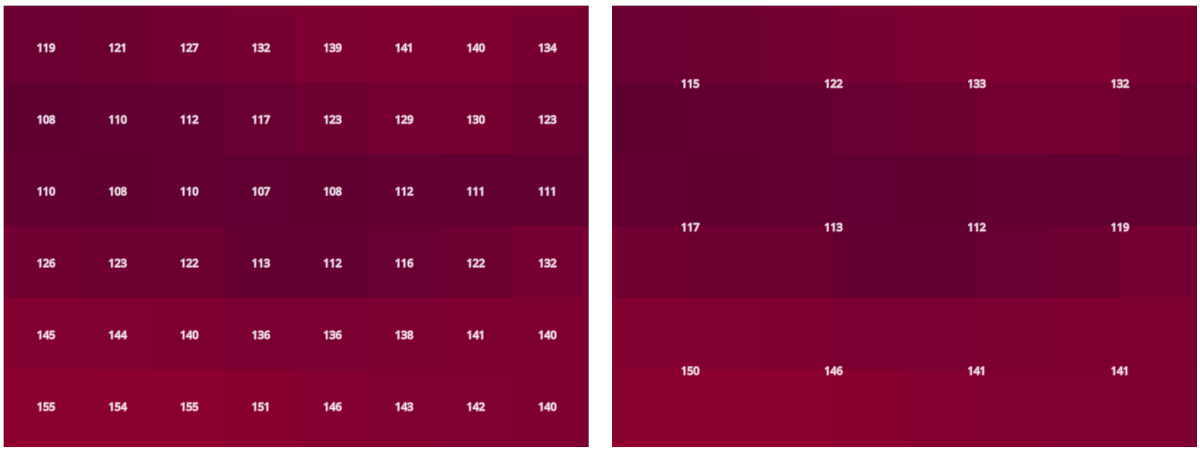


Figura13.15: Pixels labeled using various resampling options

Below are displayed options to customize the labels, under various tabs:

- abc *Text*
- +ab < c *Formatting*
- abc *Buffer*

- *Background*
- *Shadow*
- *Placement*
- *Rendering*

Description of how to set each property is exposed at *Configurando um rótulo*.

13.1.6 Propriedades do Histograma

The *Histogram* tab allows you to view the distribution of the values in your raster. The histogram is generated when you press the *Compute Histogram* button. All existing bands will be displayed together. You can save the histogram as an image with the button.

At the bottom of the histogram, you can select a raster band in the drop-down menu and *Set min/max style for* it. The *Prefs/Actions* drop-down menu gives you advanced options to customize the histogram:

- With the *Visibility* option, you can display histograms for individual bands. You will need to select the option *Show selected band*.
- The *Min/max options* allow you to 'Always show min/max markers', to 'Zoom to min/max' and to 'Update style to min/max'.
- The *Actions* option allows you to 'Reset' or 'Recompute histogram' after you have changed the min or max values of the band(s).

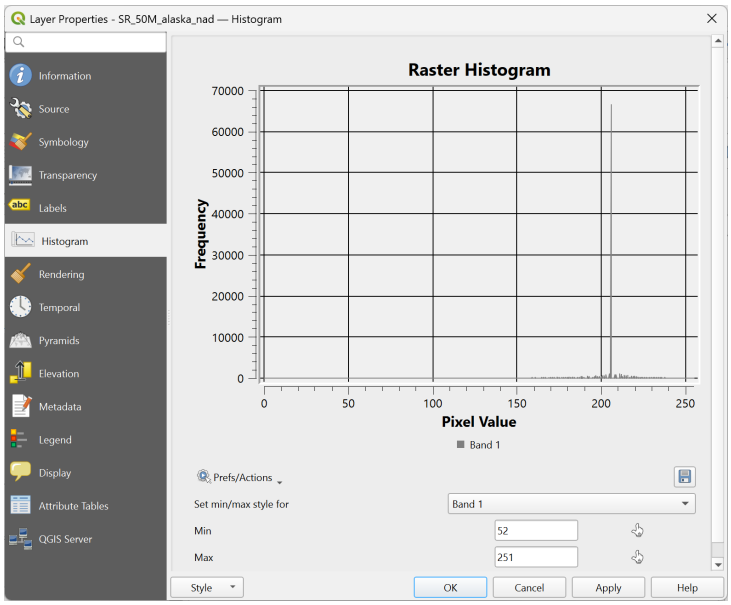


Figura13.16: Histograma do Raster

13.1.7 Propriedades de Renderização

In the *Rendering* tab, it's possible to:

- set *Scale dependent visibility* for the layer: You can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which the layer will be visible. It will be hidden outside this range. The Set to current canvas scale button helps you use the current map canvas scale as a boundary. See *Visibility Scale Selector* for more information.

  > **Nota**
  >
  > You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

- *Refresh layer at interval*: controls whether and how regular a layer can be refreshed. Available *Configuration* options are:
  - *Reload data*: the layer will be completely refreshed. Any cached data will be discarded and refetched from the provider. This mode may result in slower map refreshes.
  - *Redraw layer only*: this mode is useful for animation or when the layer's style will be updated at regular intervals. Canvas updates are deferred in order to avoid refreshing multiple times if more than one layer has an auto update interval set.

  It is also possible to set the *Interval (seconds)* between consecutive refreshments.

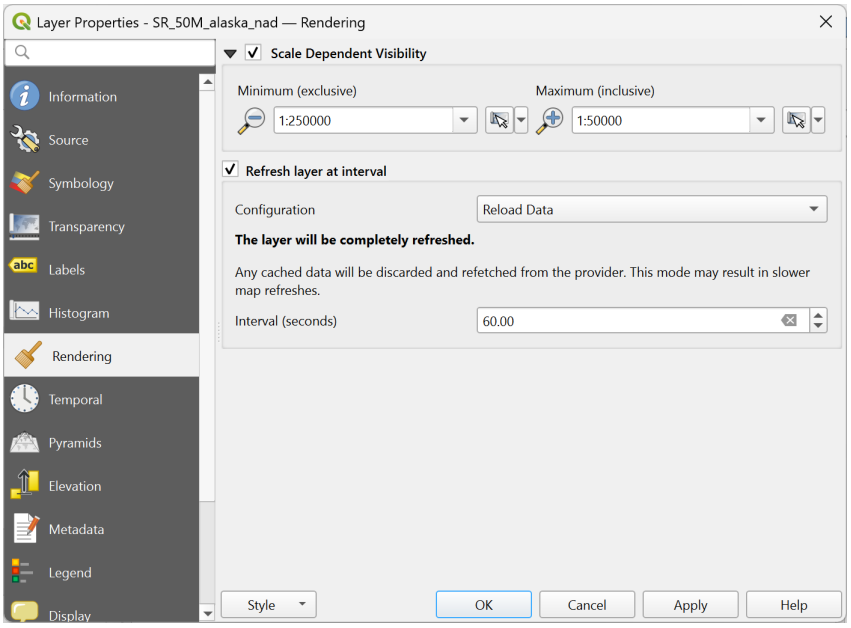


Figura13.17: Propriedades de Renderização do Raster

13.1.8 Propriedades Temporais

The *Temporal* tab provides options to control the rendering of the layer over time. Such dynamic rendering requires the *temporal navigation* to be enabled over the map canvas.

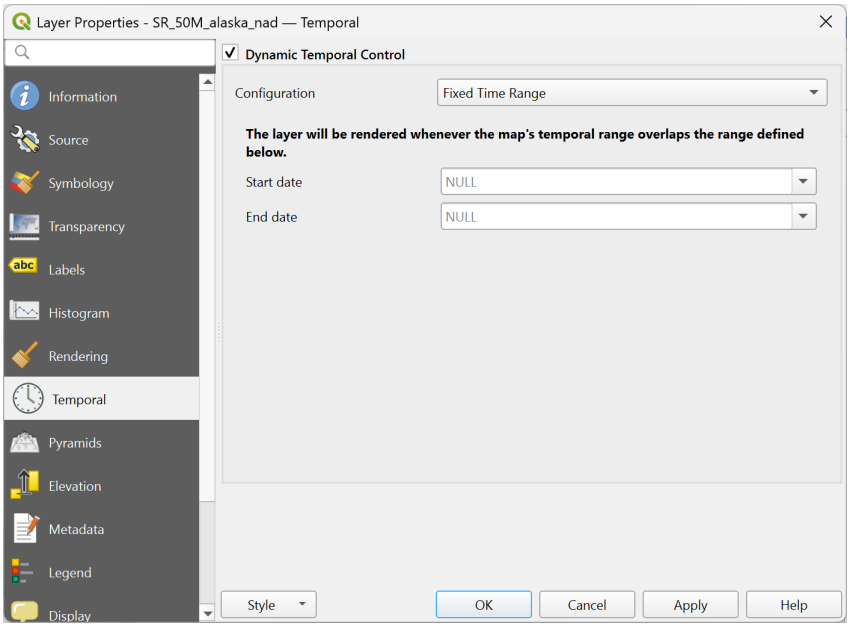


Figura13.18: Propriedades Temporais do Raster

Check the *Dynamic Temporal Control* option and set whether the layer redraw should be:

- *Automatic*: the rendering is controlled by the underlying data provider if it suppports temporal data handling. E.g. this can be used with WMS-T layers or PostgreSQL rasters.
- *Fixed Date/Time*: only show the raster layer at a single, specific date/time. This avoids having to enter the same value for both the start and end of the temporal range.
- *Fixed time range*: only show the raster layer if the animation time is within a *Start date* and *End date* range.
- *Fixed Time Range Per Band*: only shows a band when the current animation time is between its *Begin* and *End* date range. This option allows you to either manually set these time ranges for each band or use the button to automatically generate datetime values, enabling detailed temporal analysis and visualization. This mode is particularly useful for working with raster layers where each band corresponds to a specific time period, such as NetCDF files.
- *Represents Temporal Values*: interprets each pixel in the raster layer as a datetime value. When this temporal mode is active, pixels that do not fall within the temporal range specified in the render context will be hidden, ensuring that only temporally relevant data is displayed. This mode is effective for:
  - Analyzing land use changes, like observing deforestation patterns.
  - Studying flooding by comparing water coverage across different times.
  - Evaluating movement costs in terrain analysis, for example, using GRASS GIS's r.walk tool to calculate travel costs across a landscape.
- *Redraw layer only*: the layer is redrawn at each new animation frame. It's useful when the layer uses time-based expression values for renderer settings (e.g. data-defined renderer opacity, to fade in/out a raster layer).

13.1.9 Propriedades das Pirâmides

High resolution raster layers can slow navigation in QGIS. By creating lower resolution copies of the data (pyramids), performance can be considerably improved, as QGIS selects the most suitable resolution to use depending on the zoom level.

Você deve pode ter direito de gravação no diretório onde os dados originais são armazenados para construir pirâmides.

From the *Resolutions* list, select resolutions at which you want to create pyramid levels by clicking on them.

If you choose **Internal (if possible)** from the *Overview format* drop-down menu, QGIS tries to build pyramids internally.

> **Nota**
>
> Please note that building pyramids may alter the original data file, and once created they cannot be removed. If you wish to preserve a 'non-pyramided' version of your raster, make a backup copy prior to pyramid building.

If you choose **External** and **External (Erdas Imagine)** the pyramids will be created in a file next to the original raster with the same name and a `.ovr` extension.

Several *Resampling methods* can be used for pyramid calculation:

- Vizinho mais próximo
- Média
- Gauss
- Cúbico
- Cubic Spline
- Laczos
- Modo
- Nenhum

Finally, click *Build Pyramids* to start the process.

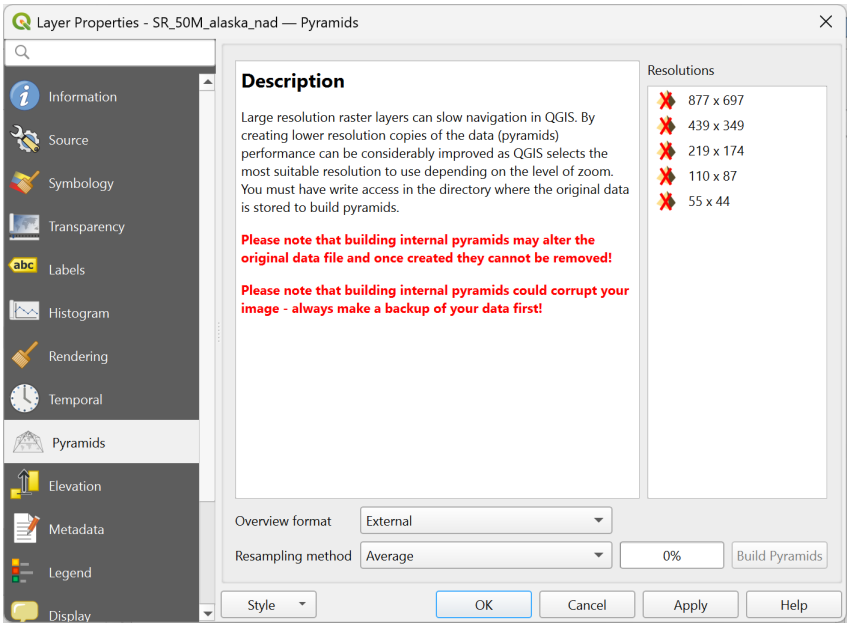


Figura13.19: Raster Pyramids

13.1.10 Elevation Properties

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and its appearance in the *profile tool charts*. Specifically, you can choose to *Disable* this configuration if the layer does not contain elevation data or you can set:

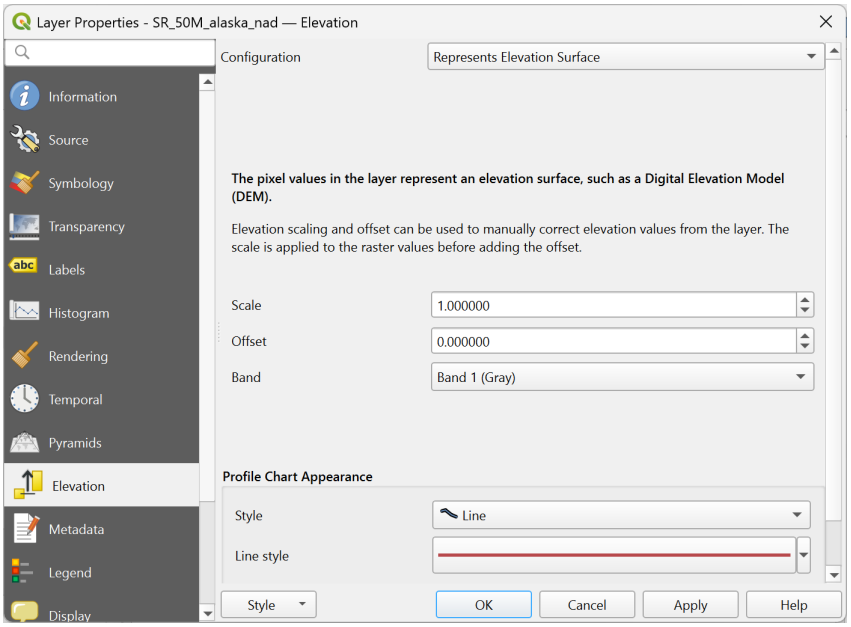


Figura13.20: Raster Elevation Properties

- *Represents Elevation Surface*: whether the raster layer represents a height surface (e.g DEM) and the pixel values should be interpreted as elevations. Choose this option if you want to display a raster in an *elevation profile view*. You will also need to fill in the *Band* to pick values from and can apply a *Scale* factor and an *Offset*.
- *Fixed Elevation Range*: The raster layer (or selected raster band) is associated with a fixed elevation range. This mode can be used when a layer has a single fixed elevation or a range (slice) of elevation values. If a range is specified, pixels will be extruded over this range. You can set the *Lower* and *Upper* elevation range values for the layer, and specify whether the lower or upper *Limits* are inclusive or exclusive.
- *Fixed Elevation Range Per Band*: Each band in the raster can have a fixed elevation range associated with it. This is designed for data sources that expose elevation-related data in bands, such as NetCDF files. For example, a raster with temperature data at different ocean depths. When rendering, the uppermost matching band will be selected and used for the layer's data. This feature is exposed as a user-editable table for raster bands with lower and upper values. Users can either populate the lower and upper values manually or use an *Expression* to auto-fill all band values based on expression. The expression-based fill allows you to design expressions that extract useful information from band names. For example, extracting the depth value from a band name like "Band 001: depth=-5500 (meters)".
- *Dynamic Elevation Range Per Band*: This mode calculates elevation ranges for raster bands dynamically using QGIS expressions. It's ideal for datasets where elevation values follow a consistent pattern across bands (like equally spaced vertical layers), eliminating the need to manually assign fixed elevations. Instead of entering individual values for each band, you define expressions for the *Lower* and *Upper* elevation bounds. These expressions can use variables like `@band`, `@band_name`, or `@band_description` automatically computing the elevation range based on each band's properties.
- *Profile Chart Appearance*: controls the rendering of the raster elevation data in the profile chart. The profile *Style* can be set as:
  - a *Line* with a specific *Line style*

  - an elevation surface rendered using a fill symbol either above (*Fill above*) or below (*Fill below*) the elevation curve line. The surface symbology is represented using:
    * a *Fill style*
    * and a *Limit*: the maximum (respectively minimum) altitude determining how high the fill surface will be

13.1.11 Propriedades dos Metadados

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metadados* for more information.

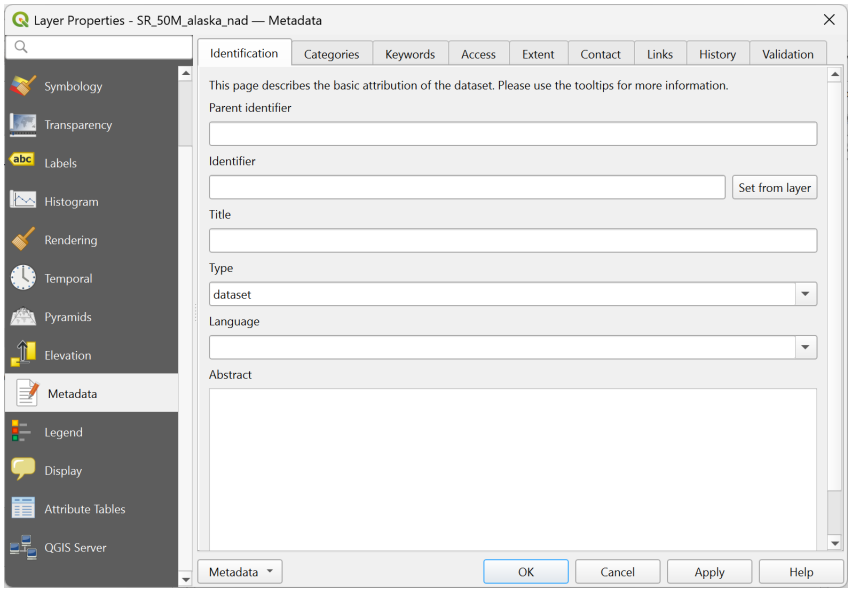


Figura13.21: Raster Metadata

13.1.12 Propriedades da legenda

The *Legend* tab provides you with advanced settings for the *Layers panel* and/or the *print layout legend*. These options include:

- Depending on the symbology applied to the layer, you may end up with several entries in the legend, not necessarily readable/useful to display. The *Legend placeholder image* helps you *select an image* for replacement, displayed both in the *Layers* panel and the print layout legend.
- The *Embedded widgets in Legend* provides you with a list of widgets you can embed within the layer tree in the Layers panel. The idea is to have a way to quickly access some actions that are often used with the layer (setup transparency, filtering, selection, style or other stuff…).

  By default, QGIS provides a transparency widget but this can be extended by plugins that register their own widgets and assign custom actions to layers they manage.

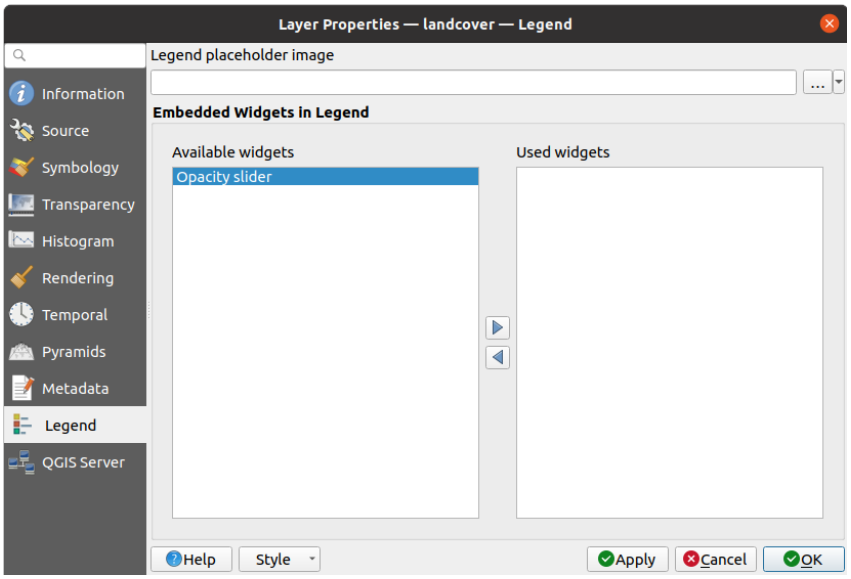


Figura13.22: Legenda do Raster

13.1.13 Display Properties

The *Display* tab helps you configure HTML map tips to use for pixels identification:

- *Enable Map Tips* controls whether to display map tips for the layer
- The *HTML Map Tip* provides a complex and full HTML text editor for map tips, mixing QGIS expressions and html styles and tags (multiline, fonts, images, hyperlink, tables, …). You can check the result of your code sample in the *Preview* frame. You can also select and edit existing expressions using the *Insert/Edit Expression* button.

  You might look for expressions located in *Rasters* group or the `@layer_cursor_point` variable in the *Expressions* dialog.

> **Nota**
>
> **Understanding the *Insert/Edit Expression* button behavior**
>
> If you select some text within an expression (between "[%" and "%]"), or if no text is selected but the cursor is inside an expression, the whole expression will be automatically selected for editing. If the cursor or a selected text is outside an expression, the dialog opens with the selection.

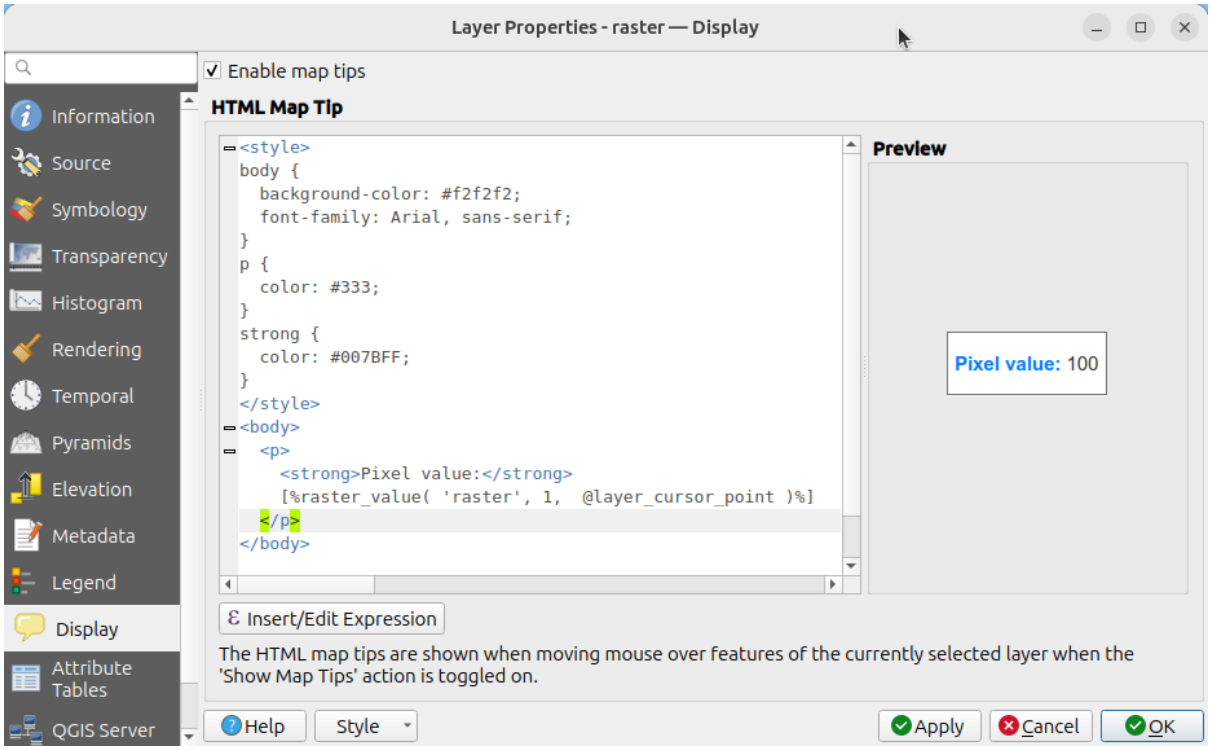


Figura13.23: Map tips with raster layer

To display map tips:

1. Select the menu option *View* ► *Show Map Tips* or click on the Show Map Tips icon of the *Attributes Toolbar*.
2. Make sure that the layer you target is active and has the *Enable Map Tips* property checked.
3. Move over a pixel, and the corresponding information will be displayed over.

Map tip is a cross-layer feature meaning that once activated, it stays on and applies to any map tip enabled layer in the project until it is toggled off.

13.1.14 Propriedades do Servidor QGIS

The *QGIS Server* tab helps you configure settings of the data when published by QGIS Server. The configuration concerns:

- *Description*: provides information to describe the data, such as *Short name*, *Title*, *Abstract*, a *Keyword list*, and a *Data URL* whose *Format* can be in `text/html`, `text/plain` or `application/pdf`.
- *Attribution*: a *Title* and *URL* to identify who provides the data
- *Metadata URL*: a list of *URL* for the metadata that can be of `FGDC` or `TC211` *Type*, and in `text/plain` or `text/xml` *Format*
- *Legend URL*: a *URL* for the legend, in either `image/png` or `image/jpeg` *Format*

Nota

When the raster layer you want to publish is already provided by a web service, further *properties* are available for setting.

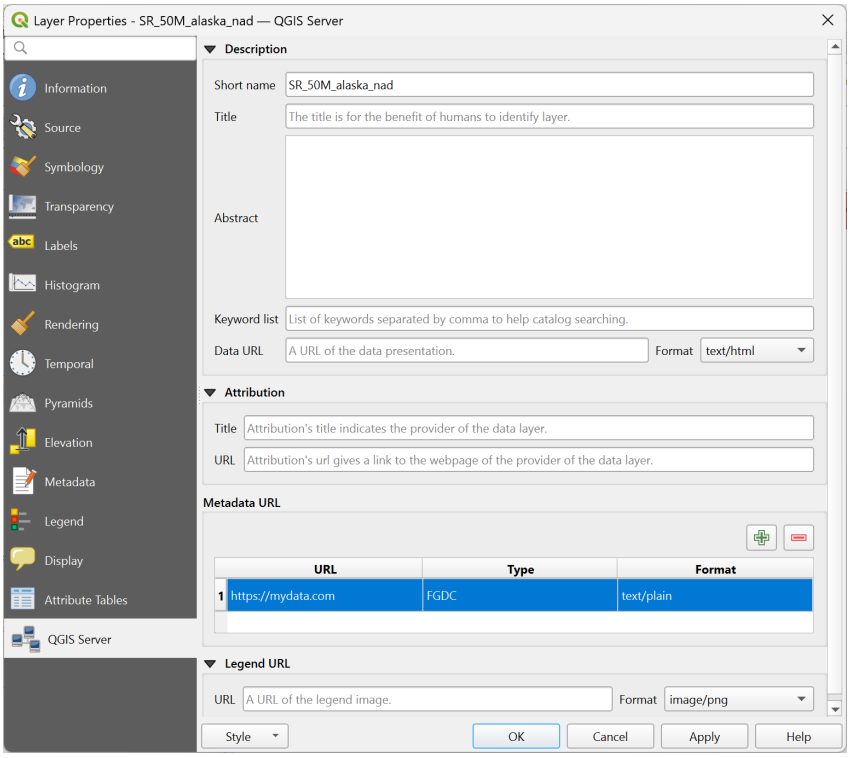


Figura13.24: QGIS Server in Raster Properties

13.1.15 Attribute Tables

Similar to vector layers, raster layers in QGIS can have associated Raster Attribute Tables (RATs) that store additional information about raster pixel values.

A Raster Attribute Table links raster values (or value ranges) to descriptive attributes such as class names, pixel counts, and colors. RATs are commonly used for classified rasters, color tables, and histogram information, helping QGIS interpret how raster values should be displayed and classified.

The *Attribute Tables* tab allows you to view and edit raster attribute data. To access the attribute table of a raster layer, right-click the layer in the layers panel, open *Properties…* and select *Attribute Tables*.

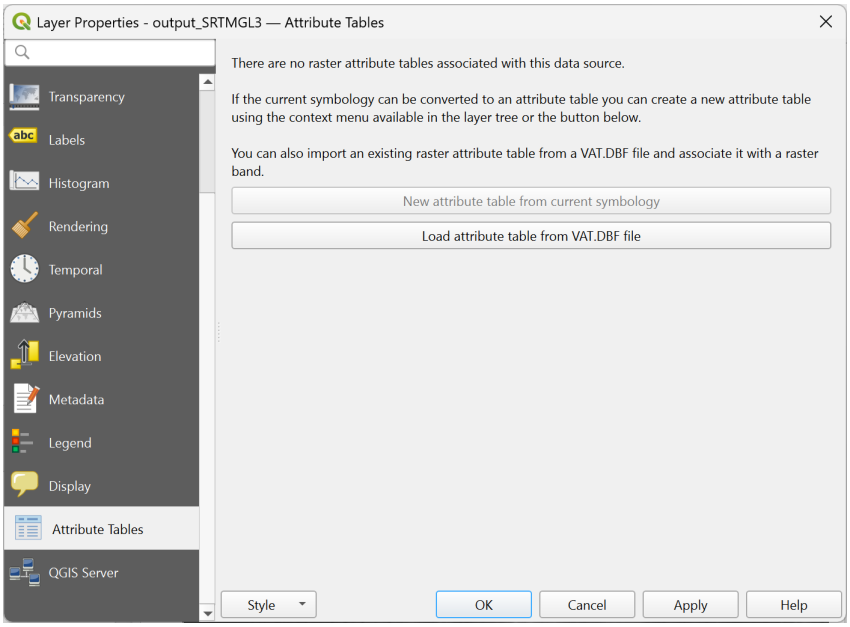


Figura13.25: Accessing the raster attribute table

When opening the *Attribute Tables* for a raster layer that does not have an associated Raster Attribute Table, QGIS displays a notice informing that no attribute table is currently linked to the data source. In this state, the interface provides two options to define and associate a new attribute table:

- *New attribute table from current symbology*: will appear if your raster layer has a classified symbology applied. Otherwise, this option is disabled. Applying classified symbology also triggers an option in the layer context menu to *Create Raster Attribute Table*. Click this button and set the following options in the *New Raster Attribute Table* dialog.
  - *Managed by the data provider*: the attribute table will be saved and managed by the data provider (if supported), overwriting any existing table for the raster band used by the current style. Depending on the data provider, the attribute table will be embedded in the main raster file or saved into a sidecar file managed by the data provider.
  - *Sidecar VAT.DBF file*: saves the attribute table into a sidecar VAT.DBF file. The resulting file will not be associated with any particular band.

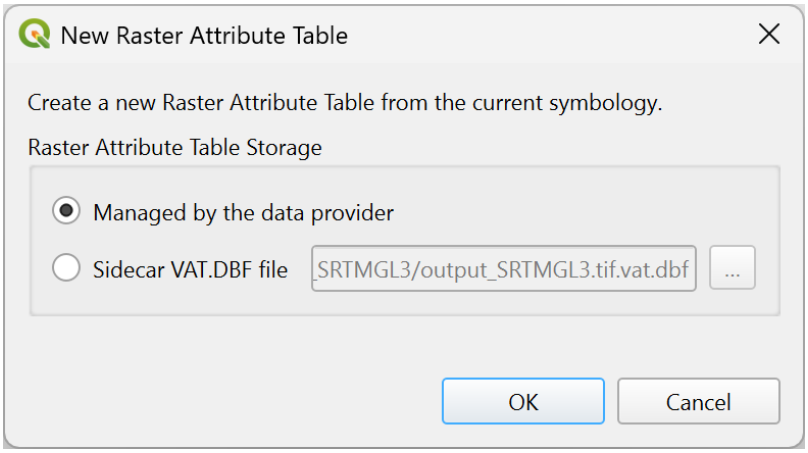


Figura13.26: Create new attribute table from current symbology

- *Load attribute table from VAT.DBF file*: retrieves the attribute table from an external DBF file. *Load Raster Attribute Table* dialog will appear to let you browse and select the DBF file and associate it with the raster

band. The action also can be done by right-clicking the raster layer in the layers panel and selecting *Load Raster Attribute Table from VAT.DBF*. *Open the newly loaded attribute table* allows you to directly open the attribute table editor after loading the table.

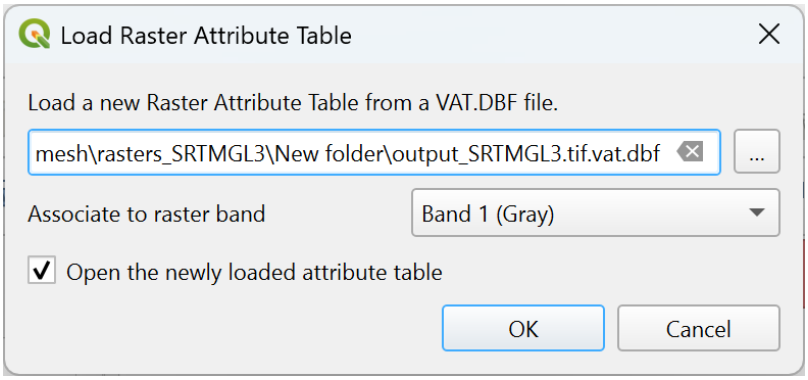


Figura13.27: Load attribute table from VAT.DBF file

Editing raster attribute tables

When set, a raster attribute table can be accessed from the *Attribute tables* tab of the layer properties dialog or by selecting *Open raster attribute table* from the layer contextual menu in *Layers* panel. This dialog shows a table whose rows correspond to the different pixel values or value ranges classified in a given band of the raster layer. Depending on the type of classification, a row is described by:

- for an exact values classification (e.g., paletted/unique values): the band name and the applied *Color*. Additional fields of the *Red*, *Green*, *Blue* and *Opacity* of the color are displayed.
- for range values classification (e.g., singleband pseudocolor): the range of values defined by their *Min*, *Max*, the associated label (*Class*) and color ramp interpolated over the values (*Color*). Additional fields of the minimum and maximum of *Red*, *Green*, *Blue* and *Opacity* of the color ramp are displayed.

You can also customize the raster attribute table by adding new fields for classification and group different value ranges accordingly, modifying value ranges and their associated colors or labels, adding new rows, … The new attribute table can be used to update the current layer symbology.

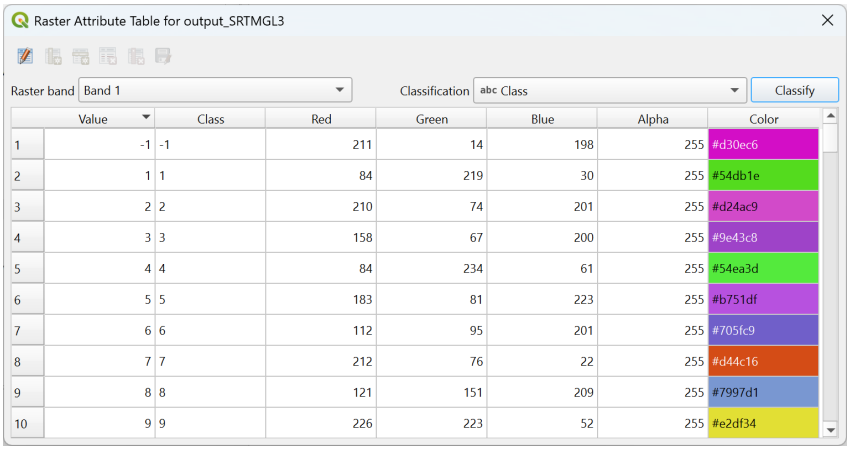


Figura13.28: Raster Attribute Table

At the top of the dialog, a set of tools allows to edit and save changes to the attribute table:

- Edit Attribute Table turns the table into edit mode. If no specific row or field is selected, only the Add Column… option is available.

- Add Column… opens the *Add Column* dialog to add a new column to the raster attribute table. The dialog lets you define:
  - *Column Type*, which can be:
    - *Standard column*
    - *Color RGBA*
    - *Color ramp (RGBA minimum and RGBA maximum)*
  - *Column Definition*, including:
    - *Name* of the column (mandatory)
    - *Usage*, which can be:
      - *Pixel Count*: stores the number of pixels belonging to a given value or value range.
      - *Generic*: general-purpose column with no predefined meaning.
      - *Name*: stores the class name or label associated with a raster value or range.
    - *Data type*, which can be:
      - *String*
      - *Integer*
      - *Long Integer*
      - *Double*
  - *Insertion point*, defining whether the new column is inserted *Before* or *After* a selected existing column.

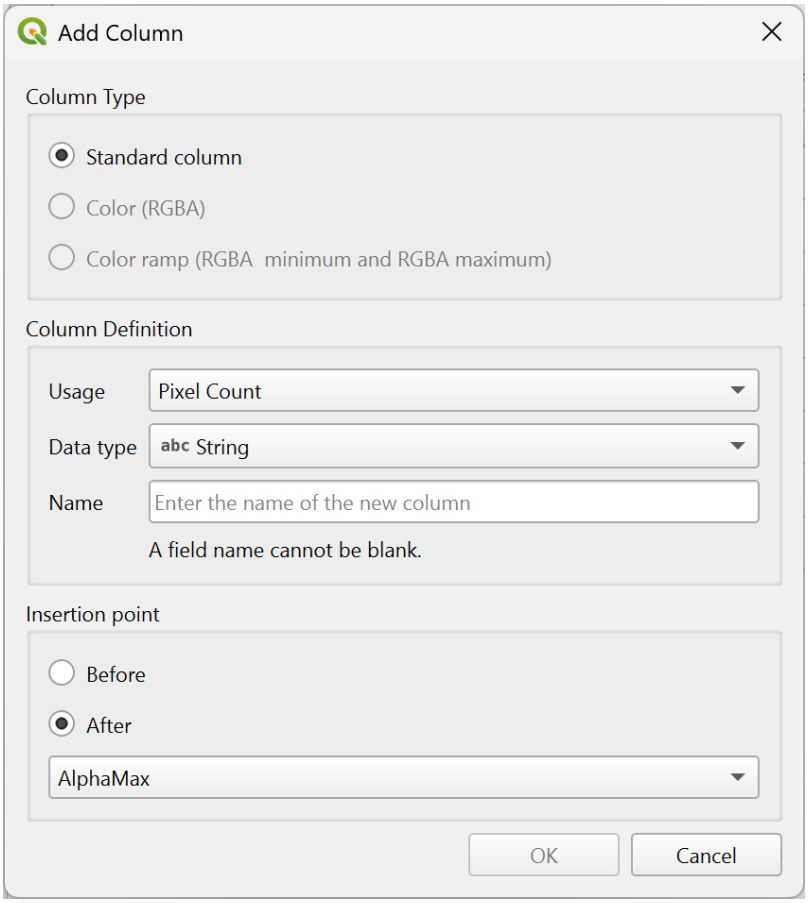


Figura13.29: Add Column dialog

- *Add Row…* opens the *Add Row* dialog to add a new row to the raster attribute table. Choose the *Insertion point* as *Before current row* or *After current row*.
- Remove Row removes the selected row from the raster attribute table.
- Remove Column removes the selected column from the raster attribute table.
- Save Changes saves any modifications made to the raster attribute table.

The next row provides options for controlling the displayed information in the table:

- *Raster band*: allows selecting which raster band the attribute table applies to. Raster layers may contain multiple bands (e.g., multispectral data). Each band can have its own Raster Attribute Table.
- *Classification*: allows selecting the fields used for classifying the raster layer. Keep in mind that the existing symbology for the raster will be replaced by a new symbology from the attribute table and any unsaved changes to the current symbology will be lost. Make sure to save your symbology style if needed before applying a classification.

13.1.16 Identify raster cells

The *identify features* tool allows you to get information about specific points in a raster layer.

To use the *Identify features* tool:

1. Select the raster layer in the Layers panel.
2. Click on the *Identify features* tool in the toolbar or press `Ctrl+Shift+I`.
3. Click on the point in the raster layer that you want to identify. The pixel will get highlighted.

The Identify Results panel will open in its default `Tree` view and display information about the clicked point. Formatting of the results vary depending on the provider of the layer. For example:

- For a local raster layer: below the name of the layer, you have on the left the band(s) of the clicked pixel, and on the right their respective value.
- For a remote layer such as WMS, a *Format* menu allows you to select whether the information should be displayed as *HTML*, *Feature* or *Text*.

These values can also be rendered (from the *View* menu located at the bottom of the panel) in:

- a `Table` view - organizes the information about the identified features and their values in a table.
- a `Graph` view - organizes the information about the identified features and their values in a graph.

Under the pixel attributes, you will find the *Derived* information, such as:

- Valores das coordenadas `X` e `Y` do ponto clicado
- Column and row of the point clicked (pixel) when compatible

13.2 Análise raster

13.2.1 Calculadora raster

The *Raster Calculator* in the *Raster* menu allows you to perform calculations on the basis of existing raster pixel values (see Fig.13.30). The results are written to a new raster layer in a GDAL-supported format.

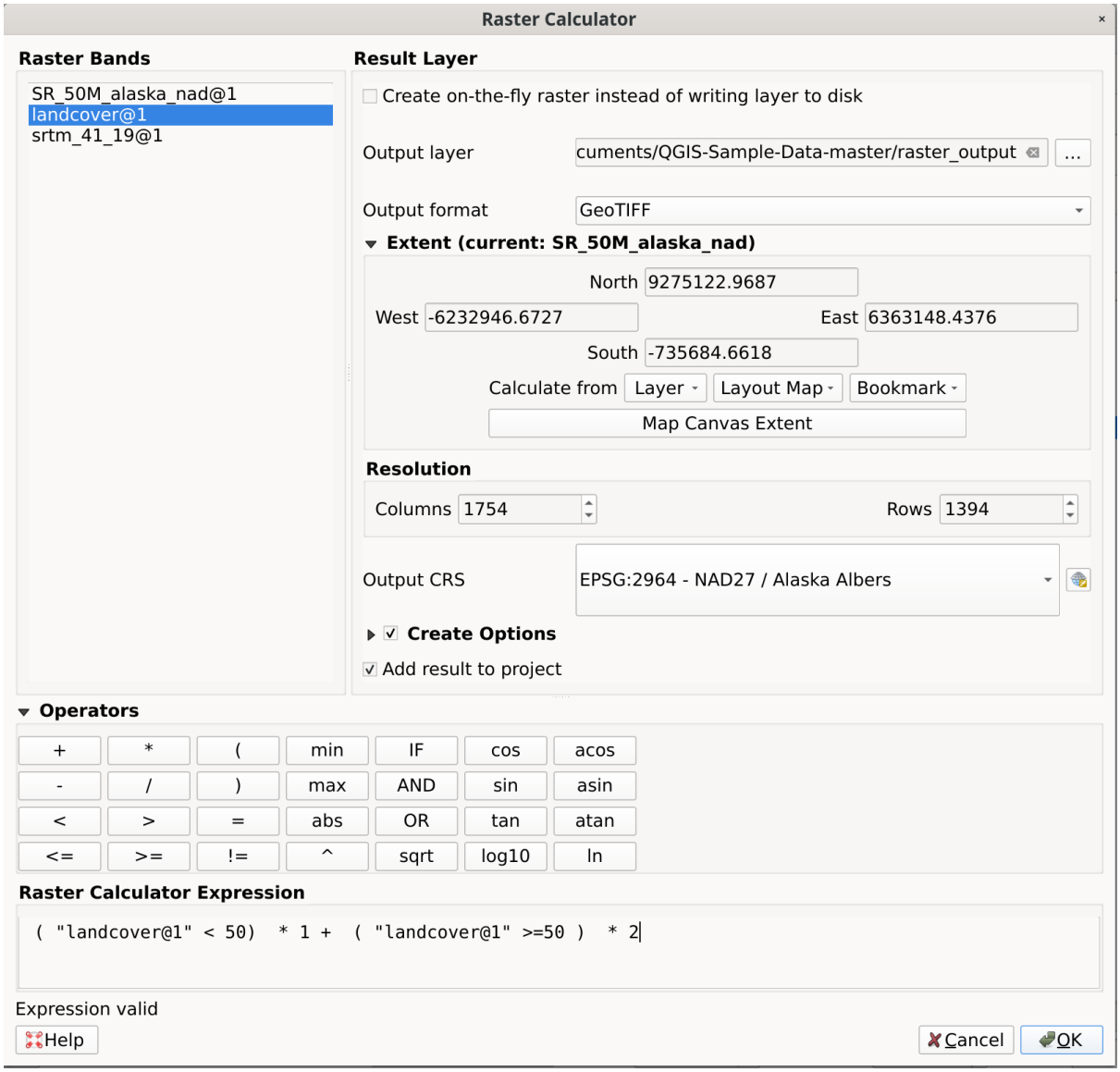


Figura13.30: Calculadora raster

The *Raster bands* list contains all loaded raster layers that can be used. To add a raster to the raster calculator expression field, double click its name in the Fields list. You can then use the operators to construct calculation expressions, or you can just type them into the box.

Na seção :guilabel:'camada de resultado', você precisará definir uma camada de saída. Você pode:

- *Create on-the-fly raster instead of writing layer to disk*:
  - If unchecked, the output is stored on the disk as a new plain file. An *Output layer* path and an *Output format* are required.
  - If checked, a virtual raster layer, i.e. a raster layer defined by its URI and whose pixels are calculated on-the-fly, is created. It's not a new file on disk; the virtual layer is still connected to the rasters used in the calculation meaning that deleting or moving these rasters would break it. A *Layer name* can be provided, otherwise the calculation expression is used as such. Removing the virtual layer from the project deletes it, and it can be made persistent in file using the layer *Export* ► *Save as…* contextual menu.
- Define the *Spatial extent* of the calculation using the *extent selector* widget.
- Set the *Resolution* of the layer using columns and rows number. If the input layer has a different resolution, the values will be resampled with the nearest neighbor algorithm.

- Enable the *Create Options* to define additional raster creation options. These options let you control how the output file is structured and compressed, including format, compression method, and other driver-specific settings. See more at *Raster driver options*.
- With the *Add result to project* checkbox, the result layer will automatically be added to the legend area and can be visualized. Checked by default for virtual rasters.

The *Operators* section contains all available operators. To add an operator to the raster calculator expression box, click the appropriate button. Mathematical calculations (+, −, *, …) and trigonometric functions (`sin`, `cos`, `tan`, …) are available. Conditional expressions (=, !=, <, >=, …) return either 0 for false or 1 for true, and therefore can be used with other operators and functions.

Ver também

Calculadora Raster and *Raster calculator (virtual)* algorithms

Raster calculator expression

The dialog

The *Raster calculator expression* dialog provides means to write expressions for pixels calculations between a set of raster layers.

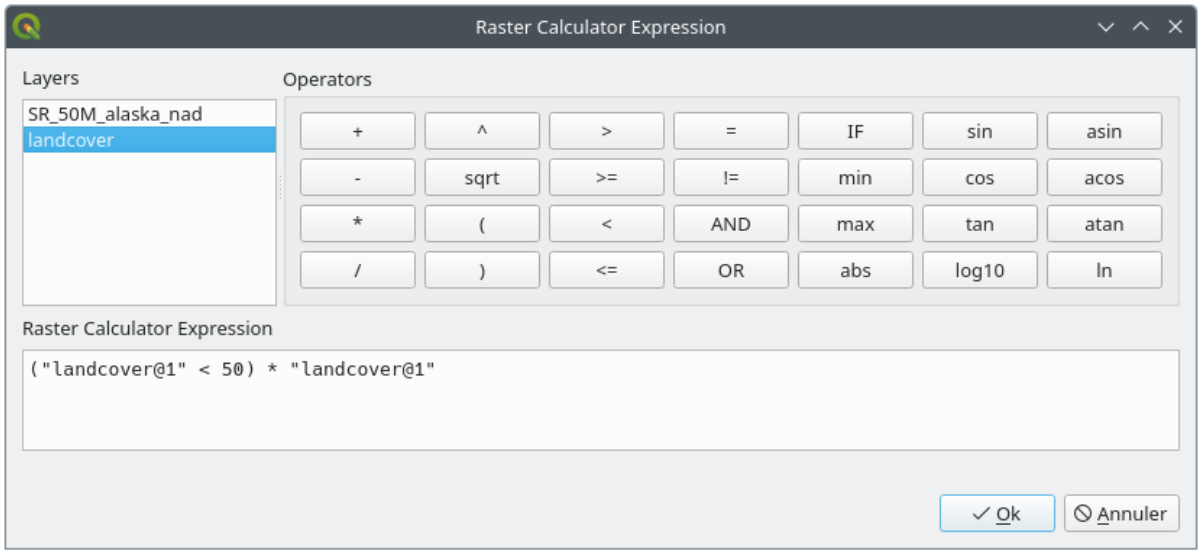


Figura13.31: Raster Expression Calculator

- *Layers*: Shows the list of all raster layers loaded in the legend. These can be used to fill the expression box (double click to add). Raster layers are referred by their name and the number of the band: `layer_name@band_number`. For instance, the first band from a layer named `DEM` will be referred as `DEM@1`.
- *Operators*: contains a number of calculation operators for pixels manipulation:
  - Arithmetical: +, −, *, `sqrt`, `abs`, `ln`, …
  - Trigonometric: `sin`, `cos`, `tan`, …
  - Comparison: =, !=, <, >=, …
  - Logical: `IF`, `AND`, `OR`, (,)
  - Statistical: `min`, `max`

To add an operator to the raster calculator expression box, click the appropriate button.

- *Raster calculator expression* is the area in which the expression is composed

Exemplos

Converter valores de elevação de metros para pés

Para criar um raster de elevação em pés a partir de um raster em metros, você precisa usar o fator de conversão de metros para pés: 3,28. A expressão é:

```
"elevation@1" * 3.28
```

Usando uma máscara

Se você deseja mascarar partes de um raster – digamos, por exemplo, porque você só está interessado em elevações acima de 0 metros – você pode usar a seguinte expressão para criar uma máscara e aplicar o resultado a um raster em um passo.

```
("elevation@1" >= 0) * "elevation@1"
```

Em outras palavras, para cada célula maior ou igual a 0, a expressão condicional é avaliada como 1, o que mantém o valor original multiplicando-o por 1. Caso contrário, a expressão condicional é avaliada como 0, que define o valor raster como 0. Isso cria a máscara em vôo.

Classificar um Raster

Se você deseja classificar um raster – por exemplo, em duas classes de elevação, é possível usar a expressão a seguir para criar um raster com dois valores 1 e 2 em um passo.

```
("elevation@1" < 50) * 1 + ("elevation@1" >= 50) * 2
```

Em outras palavras, para cada célula menor que 50, defina seu valor como 1. Para cada célula maior ou igual a 50, defina seu valor como 2.

Ou você pode utilizar o operador `SE`.

```
if ( elevation@1 < 50 , 1 , 2 )
```

CAPÍTULO 14

Working with Mesh Data

14.1 O que é uma malha?

A mesh is an unstructured grid usually with temporal and other components. The spatial component contains a collection of vertices, edges and/or faces, in 2D or 3D space:

- **vértices** - pontos XY(Z) (no sistema de referência de coordenadas da camada)
- **arestas** - conecta pares de vértices
- **faces** - a face is a set of edges forming a closed shape - typically a triangle or a quadrilateral (quad), rarely polygons with more vertices

Relying on the above, mesh layers can thus have different types of structure:

- 1D Meshes: consist of vertices and edges. An edge connects two vertices and can have assigned data (scalars or vectors) on it. The 1D mesh network can be for example used for modelling of an urban drainage system.
- 2D meshes: consist of faces with triangles, regular or unstructured quads.
- 3D layered meshes: consist of multiple stacked 2D unstructured meshes each extruded in the vertical direction (levels) by means of a vertical coordinate. The vertices and faces have the same topology in each vertical level. The mesh definition (vertical level extrusion) could in general change in time. The data is usually defined in volume centres or by some parametric function.

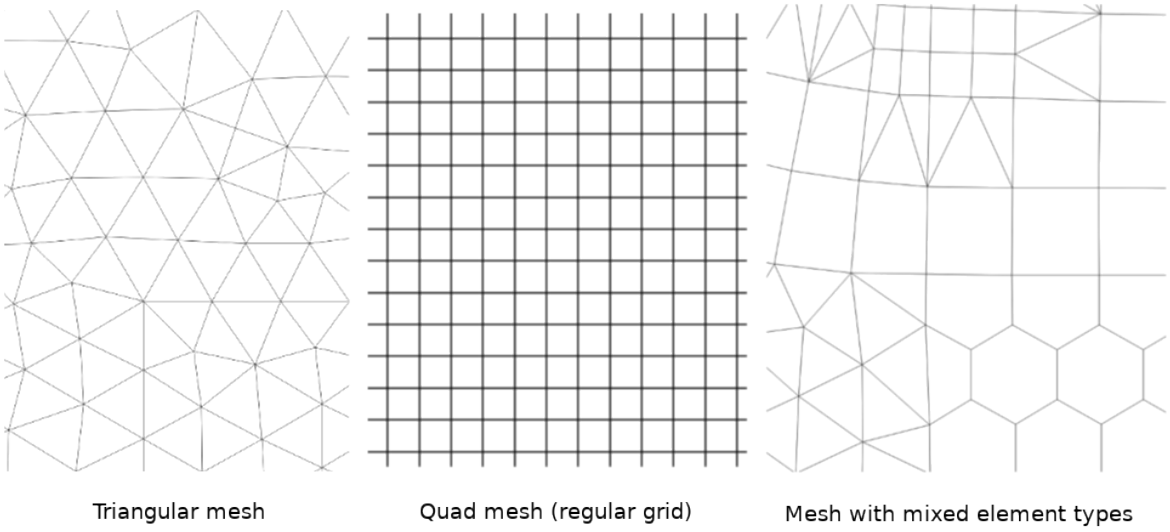


Figura14.1: Diferentes tipos de malha

Mesh provides information about the spatial structure. In addition, the mesh can have datasets (groups) that assign a value to every vertex. For example, having a triangular mesh with numbered vertices as shown in the image below:

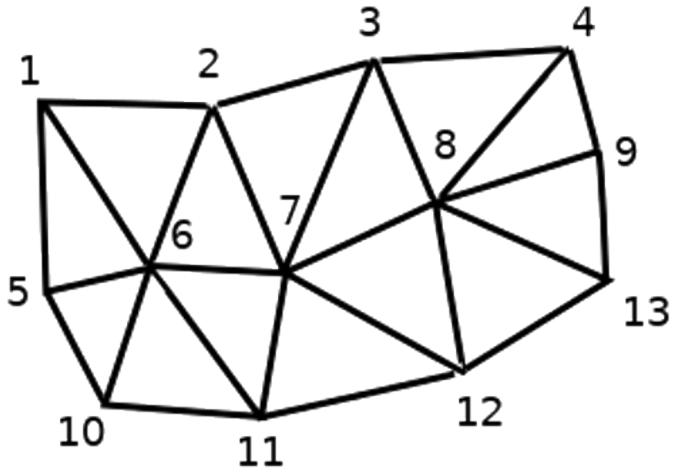


Figura14.2: Triangular grid with numbered vertices

Each vertex can store different datasets (typically multiple quantities), and those datasets can also have a temporal dimension. Thus, a single file may contain multiple datasets.

The following table gives an idea about the information that can be stored in mesh datasets. Table columns represent indices of mesh vertices, each row represents one dataset. Datasets can have different datatypes. In this case, it stores wind velocity at 10m at a particular moments in time (t1, t2, t3).

In a similar way, the mesh dataset can also store vector values for each vertex. For example, wind direction vector at the given time stamps:

Tabela14.1: Example of mesh dataset

10 metros de vento	1	2	3	...
10 metros de velocidade no tempo=t1	17251	24918	32858	…
10 metros de velocidade no tempo=t2	19168	23001	36418	…
10 metros de velocidade no tempo=t3	21085	30668	17251	…
…	…	…	…	…
10m wind direction time=t1	[20,2]	[20,3]	[20,4.5]	…
10m wind direction time=t2	[21,3]	[21,4]	[21,5.5]	…
10m wind direction time=t3	[22,4]	[22,5]	[22,6.5]	…
…	…	…	…	…

We can visualize the data by assigning colors to values (similarly to how it is done with *Singleband pseudocolor* raster rendering) and interpolating data between vertices according to the mesh topology. It is common that some quantities are 2D vectors rather than being simple scalar values (e.g. wind direction). For such quantities it is desirable to display arrows indicating the directions.

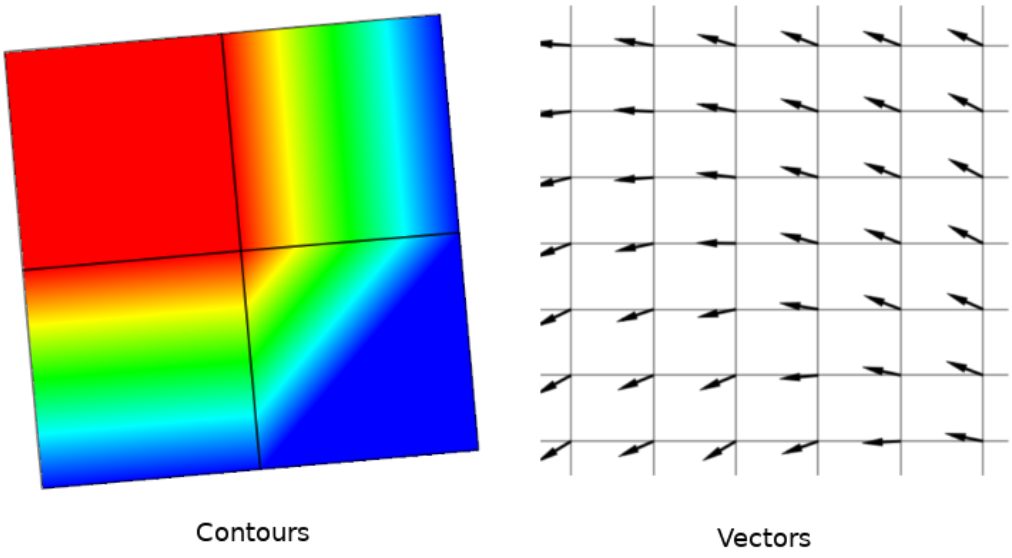


Figura14.3: Possível visualização de dados de malha

14.2 Formatos suportados

QGIS accesses mesh data using the MDAL drivers, and natively supports a variety of formats. Whether QGIS can edit a mesh layer depends on the format and the mesh structure type.

To load a mesh dataset into QGIS, use the *Mesh* tab in the *Data Source Manager* dialog. Read *Carregando uma camada de malha* for more details.

14.3 Mesh Dataset Properties

The *Layer Properties* dialog for a mesh layer provides general settings to manage dataset groups of the layer and their rendering (active dataset groups, symbology, 2D and 3D rendering). It also provides information about the layer.

Para acessar a caixa de diálogo:guilabel:*Propriedades da camada*:

- No painel:guilabel:*Camadas*, clique duas vezes na camada ou clique com o botão direito do mouse e selecione: guilabel:` Propriedades… `no menu pop-up;
- Vá para: seleção de menu: menu Camada -> Propriedades da camada … `quando a camada estiver selecionada.

The mesh *Layer Properties* dialog provides the following sections:

Tabela14.2: Tabs of the Mesh Layer Properties

Information	*Source*	*Symbology* ([1])
Labels ([1])	*3D View* ([1])	*Rendering*
Temporal	*Elevation*	*Metadata*

Nota

Most of the properties of a mesh layer can be saved to or loaded from a `.qml` using the *Style* menu at the bottom of the dialog. More details at *Gerenciar Estilos Personalizados*.

[1] Also available in the *Layer styling panel*

14.3.1 Propriedades da informação

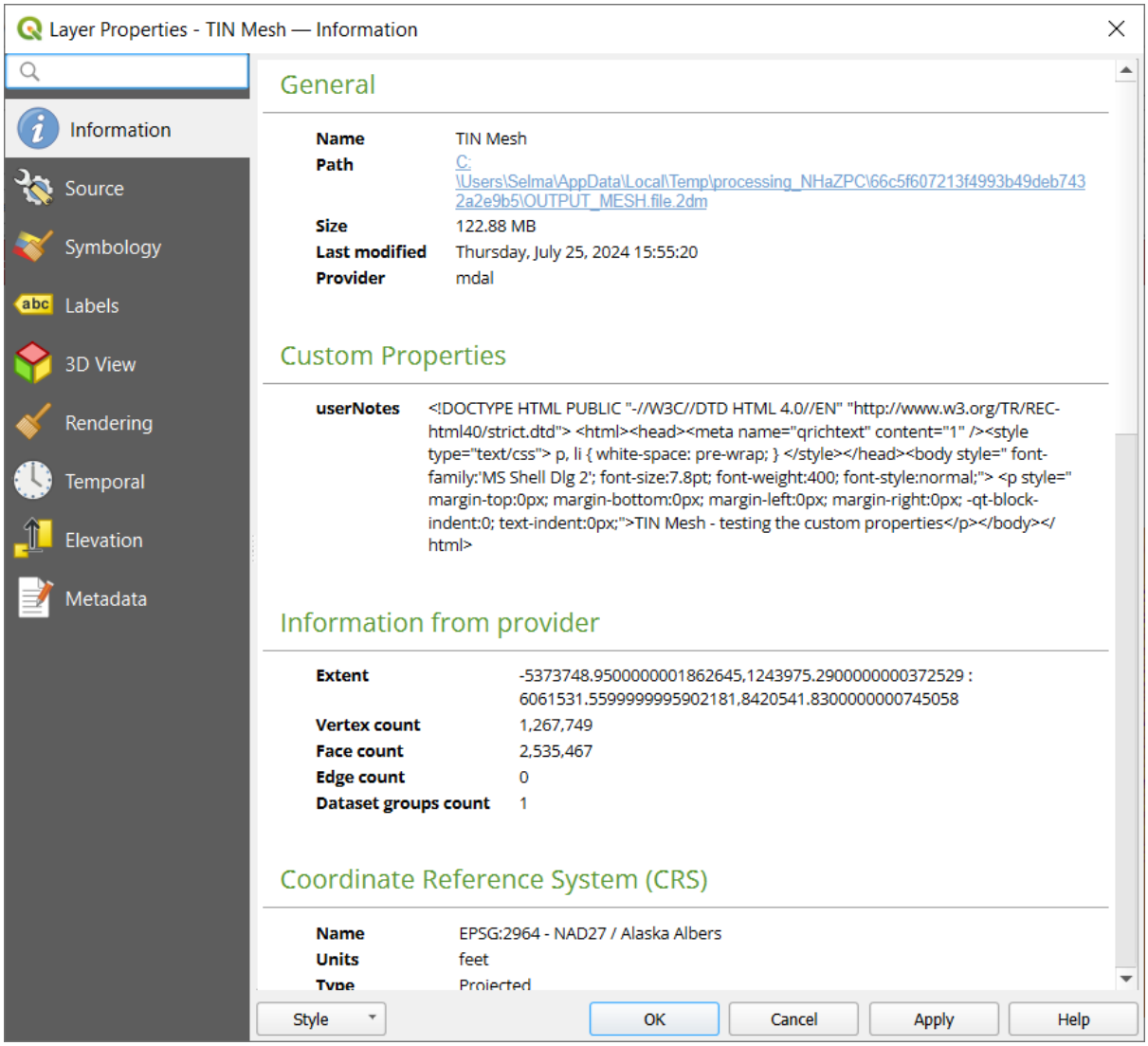


Figura14.4: Mesh Layer Information Properties

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- general such as name in the project, source path, list of auxiliary files, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- com base no provedor da camada: extensão, vértice, face, arestas e/ou contagem de grupos de conjuntos de dados
- o Sistema de Referência de Coordenadas: nome, unidades, método, precisão, referência (ou seja, se é estático ou dinâmico)
- extracted from filled *metadata*: access, extents, links, contacts, history…

14.3.2 Propriedades de origem

The *Source* tab displays basic information about the selected mesh, including:

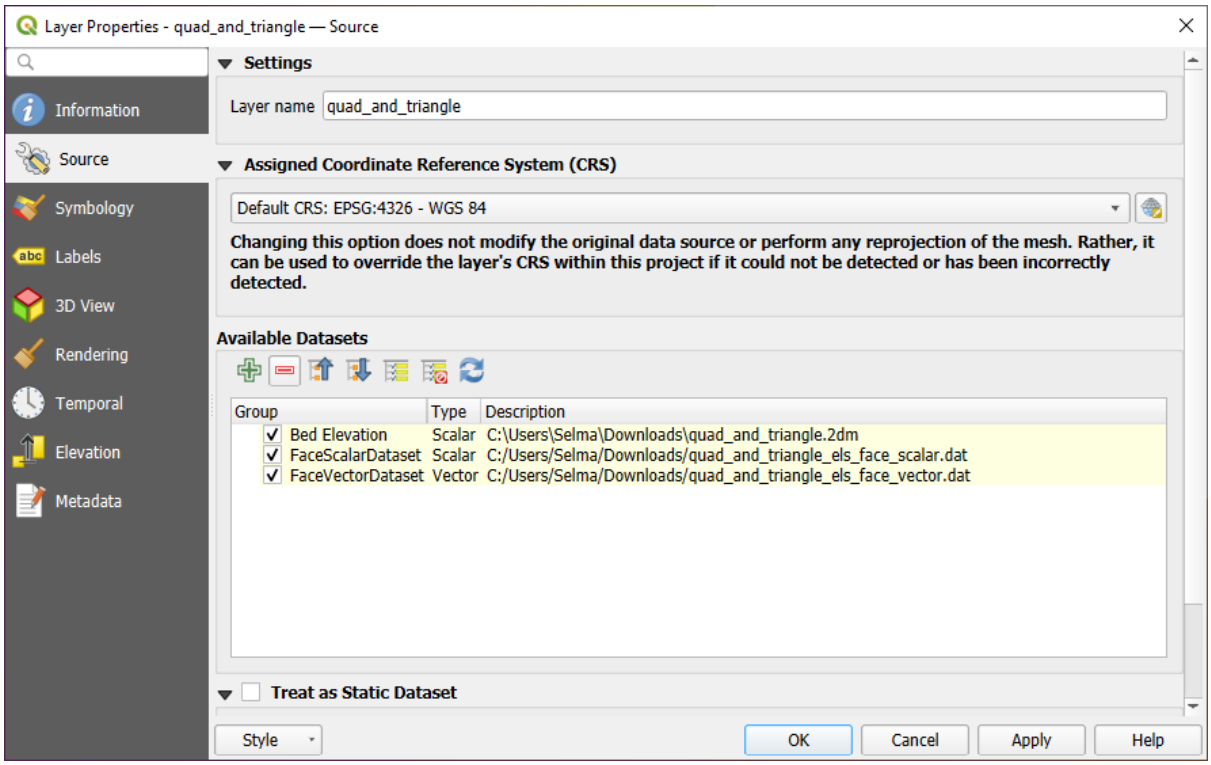


Figura14.5: Mesh Layer Source Properties

- the layer name to display in the *Layers* panel
- setting the Coordinate Reference System: Displays the layer's *Assigned Coordinate Reference System (CRS)*.

  You can change the layer's CRS by selecting a recently used one in the drop-down list or clicking on *Select CRS* button (see *Seletor do Sistemas de Referência de Coordenadas*). Use this process only if the CRS applied to the layer is wrong or if none was applied.
- The *Available datasets* frame lists all the dataset groups (and subgroups) in the mesh layer, with their type and description in a tree view. Both regular datasets (i.e. their data is stored in the file) and virtual datasets (which are *calculated on the fly*) are listed.
  - Use the Assign Extra Dataset to Mesh button to add more groups to the current mesh layer. You can add dataset group to a mesh layer with the same name, but not from the same URI. Dataset group names are automatically renamed to "Original Name_Number".
  - Use Remove Extra Dataset from Mesh to remove additional datasets groups from the mesh layer. Note that only dataset groups not associated with the mesh source file can be removed.
  - Collapse all and Expand all the dataset tree, in case of embedded groups
  - Se você estiver interessado em poucos conjuntos de dados, poderá desmarcar os outros e torná-los indisponíveis no projeto
  - Clique duas vezes sobre um nome e você poderá renomear o conjunto de dados.
  - Reset to defaults: checks all the groups and renames them back to their original name in the provider.
  - Clique com o botão direito do mouse em um grupo de conjuntos de dados virtuais e você poderá:

    * *Remove dataset group* from the project
    * *Save dataset group as…* a file on disk, to any supported format. The new file is kept assigned to the current mesh layer in the project.

- Checking the *Treat as static dataset* group allows to ignore the *map temporal navigation* properties while rendering the mesh layer. For each active dataset group (as selected in *Symbology* ► *Datasets* tab), you can:
  - set to *None*: the dataset group is not displayed at all
  - *Display dataset*: e.g., for the "bed elevation" dataset which is not time aware
  - extract a particular date time: the dataset matching the provided time is rendered and stays fixed during map navigation.

14.3.3 Propriedades da simbologia

Click the *Symbology* button to activate the dialog. Symbology properties are divided into several tabs:

- *Datasets*
- *Contornos*
- *Vetores*
- *Rendering*
- *Stacked mesh averaging method*

Conjunto de dados

The tab Datasets is the main place to control and set which datasets will be used for the layer. It presents the following items:

- *Groups* available in the mesh dataset, with whether they provide:
  - scalar dataset
  - or vector dataset: by default, each vector dataset has a scalar dataset representing its magnitude automatically generated.

  Click on the icon next to the dataset name to select the group and type of data to represent.
- *Selected dataset group(s) metadata*, with details on:
  - the mesh type: edges or faces
  - o tipo de dados: vértices, arestas, faces ou volume
  - se é do tipo vetorial ou não
  - o nome original na camada de malha
  - a unidade, se aplicável
- *blending mode* available for the selected datasets.

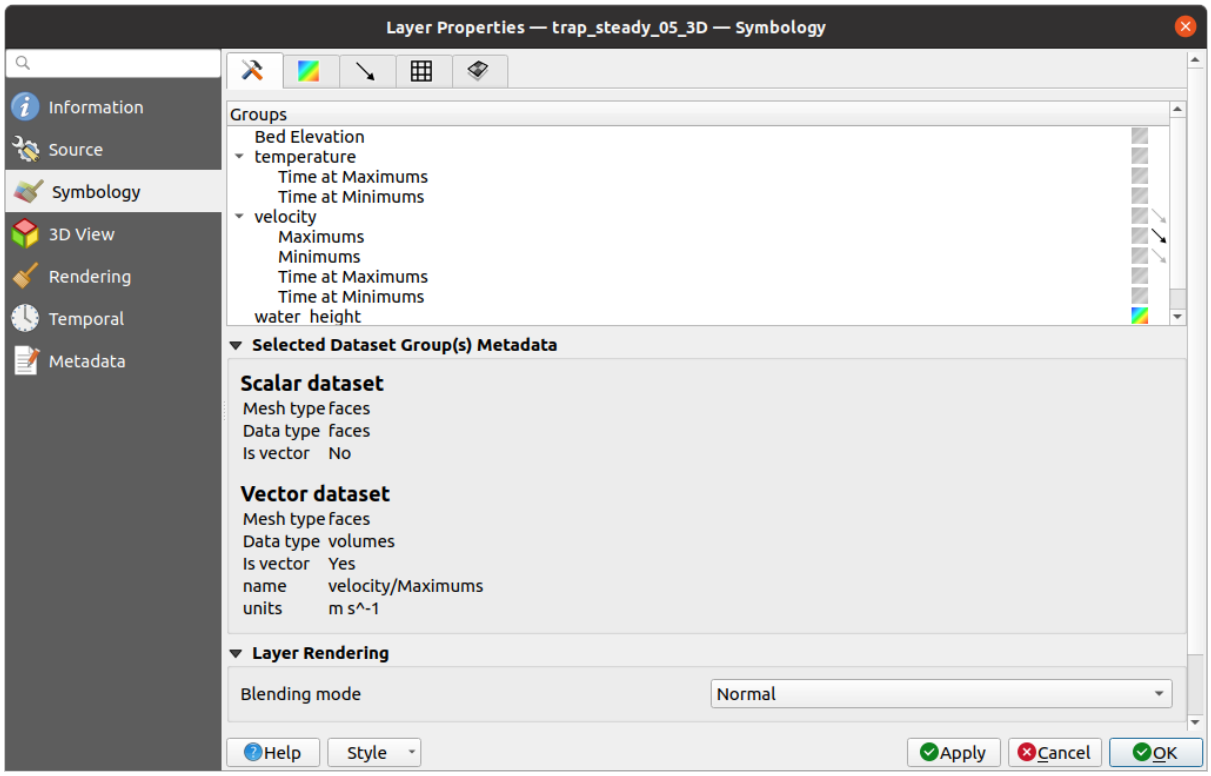


Figura14.6: Mesh Layer Datasets

Você pode aplicar simbologia ao vetor e/ou grupo escalar selecionado usando as próximas abas.

Simbologia de Contornos

Nota

The Contours tab can be activated only if a scalar dataset has been selected in the *Datasets* tab.

In the Contours tab you can see and change the current visualization options of contours for the selected group, as shown in Fig.14.7 below:

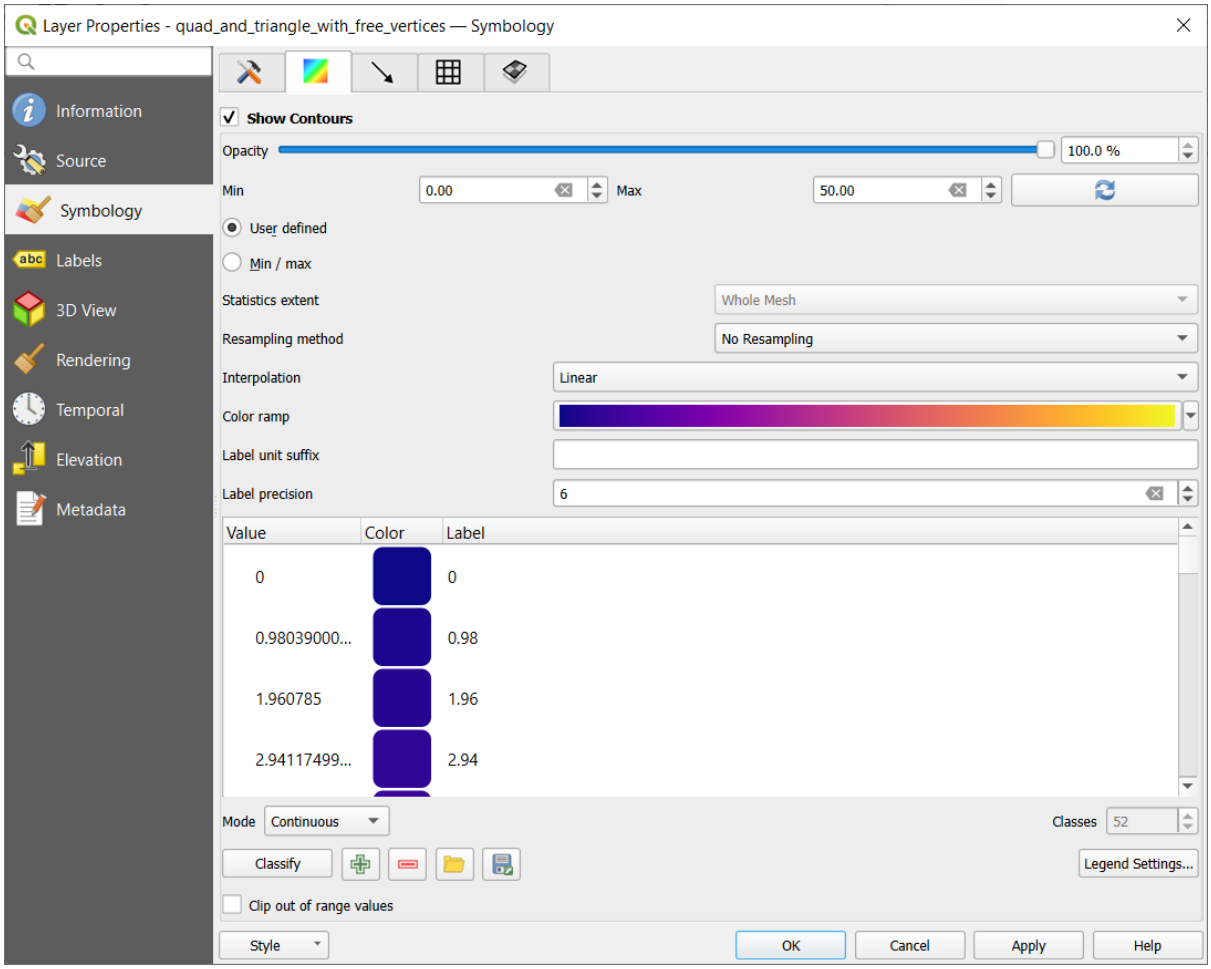


Figura14.7: Styling Contours in a Mesh Layer

- For 1D mesh, set the *Stroke width* of the edges. This can be a fixed size for the whole dataset, or vary along the geometry (more details with the *interpolated line renderer*)
- Use the slider or the spinbox to set the *Opacity* of the current group, if of a 2D mesh type.
- *User Defined* allows you to enter the range of values you want to represent on the current group: use Load to fetch the min and max values of the current group or enter custom values if you want to exclude some.
- Select *Min/Max* to set the renderer's minimum and maximum values based on the chosen extent.
- *Statictics extent* can be *Whole mesh*, *Current canvas* or *Updated canvas*. *Updated canvas* means that min/max values used for the rendering will change with the canvas extent (dynamic stretching).
- For 2D/3D meshes, select the *Resampling method* to interpolate the values on the surrounding vertices to the faces (or from the surrounding faces to the vertices) using the *Neighbour average* method. Depending on whether the dataset is defined on the vertices (respectively on the faces), QGIS defaults this setting to *None* (respectively *Neighbour average*) method in order to use values on vertices and keep the default rendering smooth.
- Classify the dataset using the *color ramp shader* classification.

Simbologia de Vetores

Nota

The Vectors tab can be activated only if a vector dataset has been selected in the *Datasets* tab. Click on the Vectors icon on the right side of the *Datasets* tab.

Na aba Vetores você pode ver e alterar as opções atuais de visualização de vetores para o grupo selecionado, como mostrado em Fig.14.8:

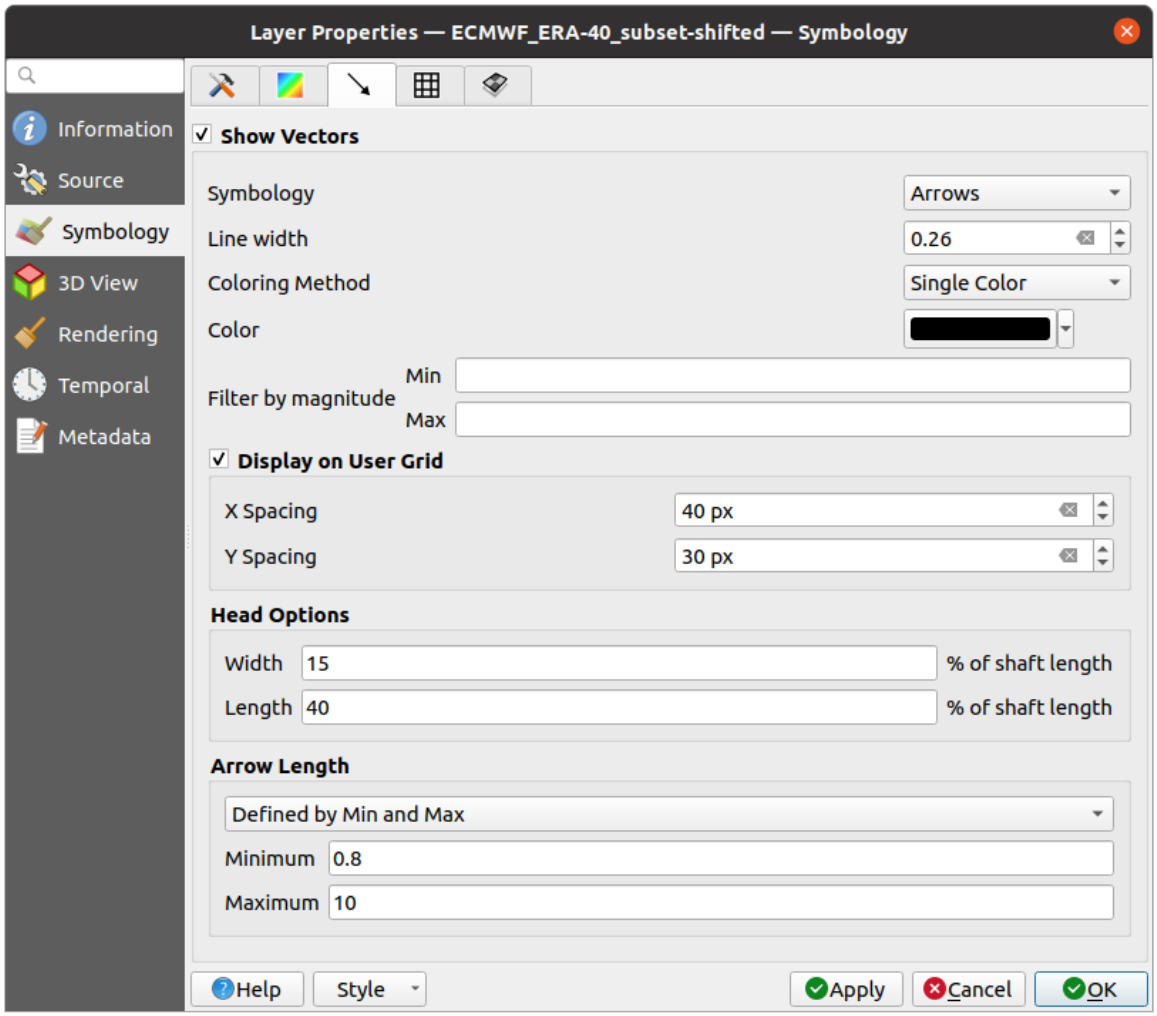


Figura14.8: Styling Vectors in a Mesh Layer with arrows

Mesh vector dataset can be styled using various types of *Symbology*:

- **Arrows**: vectors are represented with arrows at the same place as they are defined in the raw dataset (i.e. on the nodes or center of elements) or on a user-defined grid (hence, they are evenly distributed). The arrow length is proportional to the magnitude of the arrow as defined in the raw data but can be scaled by various methods.
- **Streamlines**: vectors are represented with streamlines seeded from start points. The seeding points can start from the vertices of the mesh, from a user grid or randomly.
- **Traces**: a nicer animation of the streamlines, the kind of effect you get when you randomly throws sand in the water and see where the sand items flows.

- **Wind barbs**: vectors are represented with wind barbs, a common way to represent wind speed and direction.

As propriedades disponíveis dependem da simbologia selecionada, conforme mostrado na tabela a seguir.

Tabela14.3: Availability and meaning of the vectors symbology properties

Label, Description and Properties	Seta	Streamlines	Traços	Wind Barbs
Line width: Width of the vector representation	☑	☑	☑	☑
Coloring method: • a *Single color* assigned to all vectors • or a variable color based on vectors magnitude, using a *Color ramp shader*	☑	☑	☑	☑
Filter by magnitude: Only vectors whose length for the selected dataset falls between a *Min* and *Max* range are displayed	☑	☑		☑
Display on user grid: Places the vector on a grid with custom *X spacing* and *Y spacing* and interpolates their length based on neighbours	☑	☑		☑
Head options: *Length* and *Width* of the arrow head, as a percentage of its shaft length	☑			
Arrow length: • **Definido por Min e Max**: Você especifica o comprimento mínimo e máximo para as setas, e o QGIS irá interpolar seu tamanho com base na magnitude do vetor subjacente • **Escala para magnitude**: o comprimento da seta é proporcional à magnitude do vetor • **Fixed**: all the vectors are shown with the same length	☑			
Streamlines seeding method: • **On mesh/grid**: relies on the user grid to display the vectors • **Aleatoriamente**: o posicionamento do vetor é feito aleatoriamente em relação a uma determinada densidade		☑		
Particles count: The amount of “sand” you want to throw into visualisation			☑	
Max tail length: The time until the particle fades out			☑	
Wind Barbs: • **Length**: Specifies the length of the wind barbs. Choose the units from a drop-down menu. • **Data Units**: Specifies the units of wind speed data. Wind barbs encode the wind in knots (nautical miles per hour). Therefore, you should select the units that your data are in, or use a custom *Multiplier* to convert your data units to knots. This ensures that the wind speed is accurately represented.				☑

Renderização

In the tab Rendering tab, QGIS offers possibilities to display and customize the mesh structure. *Line width* and *Line color* can be set to represent:

- as arestas para malhas 1D
- Para malhas 2D:
  - *Native mesh rendering*: shows original faces and edges from the layer
  - *Triangular mesh rendering*: adds more edges and displays the faces as triangles

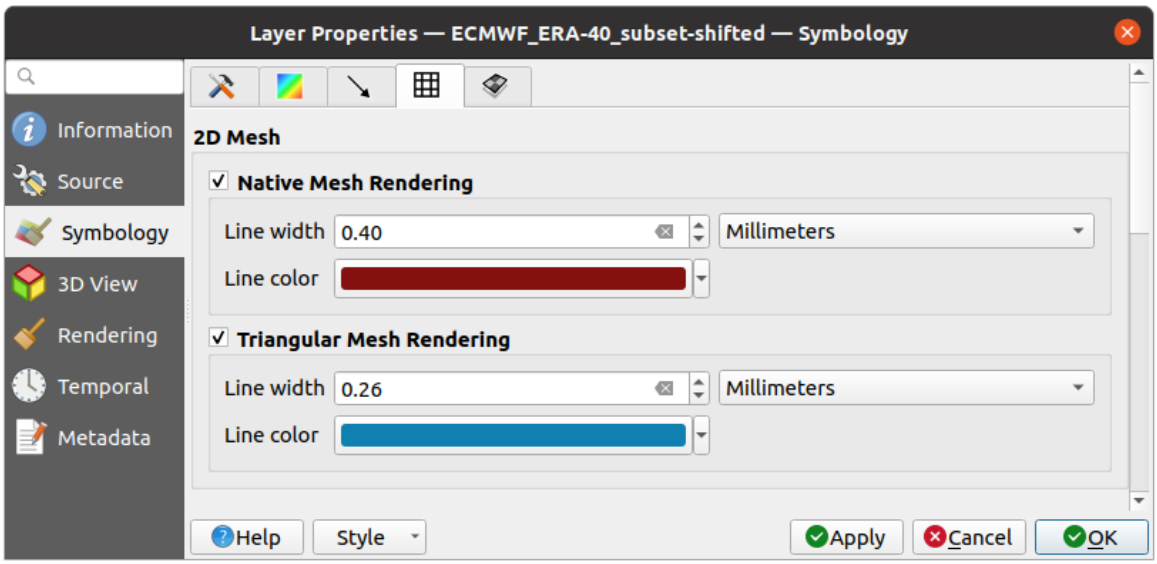


Figura14.9: 2D Mesh Rendering

Stacked mesh averaging method

3D layered meshes consist of multiple stacked 2D unstructured meshes each extruded in the vertical direction (`levels`) by means of a vertical coordinate. The vertices and faces have the same topology in each vertical level. Values are usually stored on the volumes that are regularly stacked over base 2d mesh. In order to visualise them on 2D canvas, you need to convert values on volumes (3d) to values on faces (2d) that can be shown in mesh layer. The Stacked mesh averaging method provides different averaging/interpolation methods to handle this.

You can select the method to derive the 2D datasets and corresponding parameters (level index, depth or height values). For each method, an example of application is shown in the dialog but you can read more on the methods at https://fvwiki.tuflow.com/Depth_Averaging_Results.

14.3.4 Propriedades das etiquetas

The *Labels* tab offers you dynamic labeling of mesh vertices and native mesh faces based on geometric properties and custom expressions. This dialog can also be accessed from the *Layer Styling* panel.

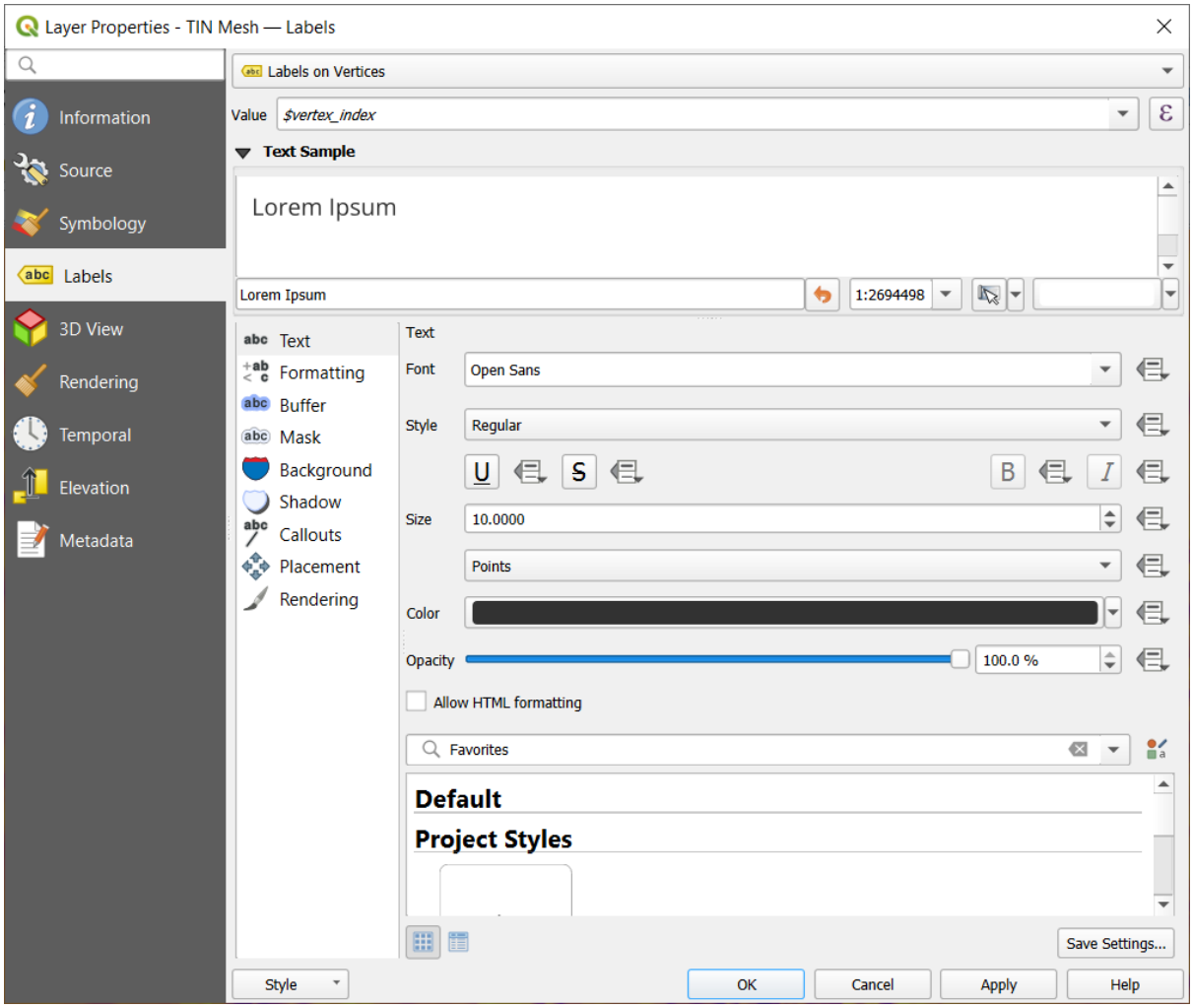


Figura14.10: Mesh Labels Properties

Choose the labeling method from the drop-down list. Available methods are:

- *No labels*: the default value, showing no labels from the layer.
- *Labels on vertices*: labels the mesh vertices as a point feature.
- *Labels on faces*: labels the mesh faces as a polygon feature.

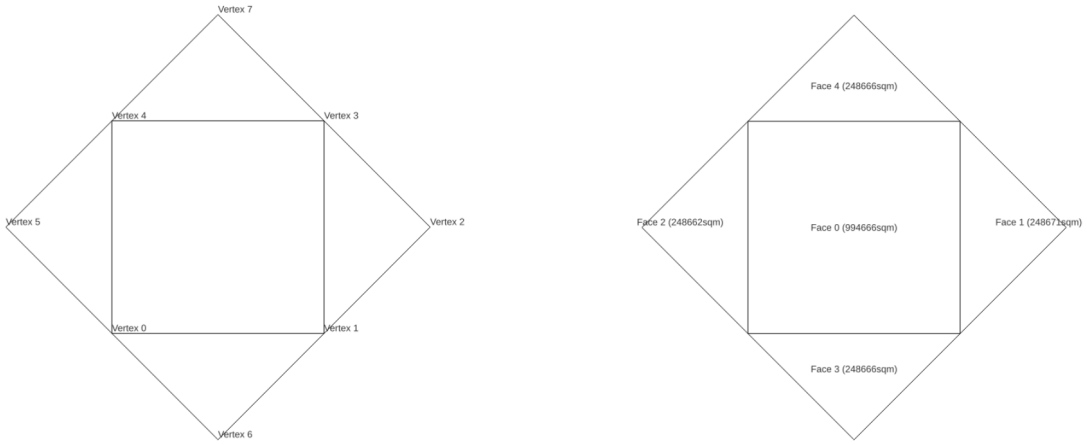


Figura14.11: Mesh Labeling methods

Once the labeling method has been selected, open Expression dialog and choose a *mesh function*. You can combine meshes functions with other functions from the *list* to modify the label.

Abaixo estão as opções exibidas para personalizar os rótulos, em várias guias:

- texto | *Texto*
- formatação de etiquetas | *Formatação*
- *Buffer*
- *Mask*
- *Background*
- *Shadow*
- *Callouts*
- *Placement*
- *Rendering*

A descrição de como definir cada propriedade é exposta em:ref:*showlabels*.

14.3.5 Propriedades da vista 3D

Mesh layers can be used as *terrain in a 3D map view* based on their vertices Z values. From the *3D View* properties tab, it's also possible to render the mesh layer's dataset in the same 3D view. Therefore, the vertical component of the vertices can be set equal to dataset values (for example, level of water surface) and the texture of the mesh can be set to render other dataset values with color shading (for example velocity).

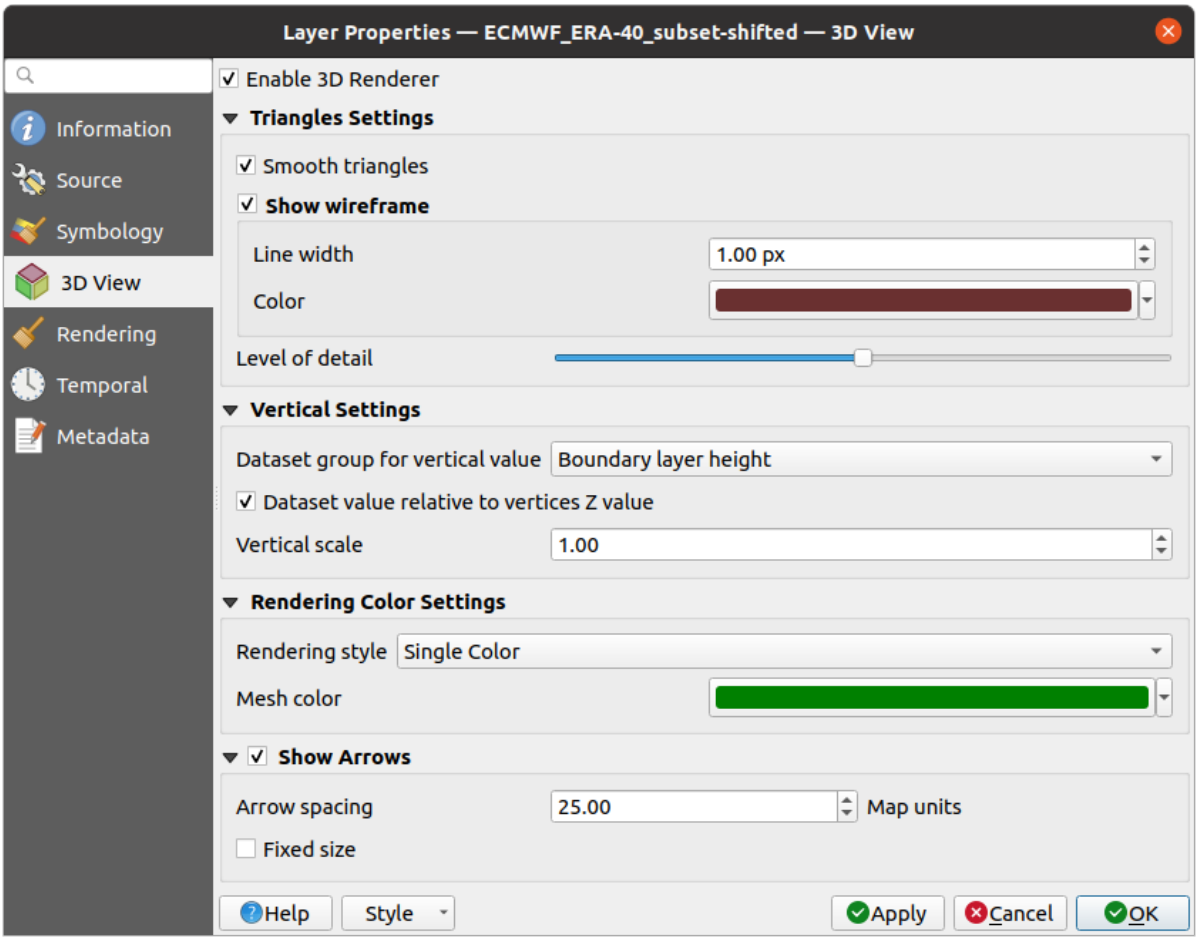


Figura14.12: Propriedades 3D do conjunto de dados de malha

Check *Enable 3D Renderer* and you can edit following properties:

- Under *Triangle settings*
  - *Smooth triangles*: Angles between consecutive triangles are smoothed for a better 3D rendering
  - *Show wireframe* whose you can set the *Line width* and *Color*
  - *Level of detail*: Controls how *simplified* the mesh layer to render should be. On the far right, it is the base mesh, and the more you go left, the more the layer is simplified and is rendered with less details. This option is only available if the *Simplify mesh* option under the *Rendering* tab is activated.
- *Vertical settings* to control behavior of the vertical component of vertices of rendered triangles:
  - *Dataset group for vertical value*: the dataset group that will be used for the vertical component of the mesh
  - *Dataset value relative to vertices Z value*: whether to consider the dataset values as absolute Z coordinate or relative to the vertices native Z value
  - *Vertical scale*: the scale factor to apply to the dataset Z values
- *Rendering color settings* with a *Rendering style* that can be based on the color ramp shader set in *Simbologia de Contornos* (*2D contour color ramp shader*) or as a *Single color* with an associated *Mesh color*
- *Show arrows*: displays arrows on mesh layer dataset 3D entity, based on the same vector dataset group used in the *vector 2D rendering*. They are displayed using the 2D color setting. It's also possible to define the *Arrow spacing* and, if it's of a *Fixed size* or scaled on magnitude. This spacing setting defines also the max size of arrows because arrows can't overlap.

14.3.6 Propriedades de Renderização

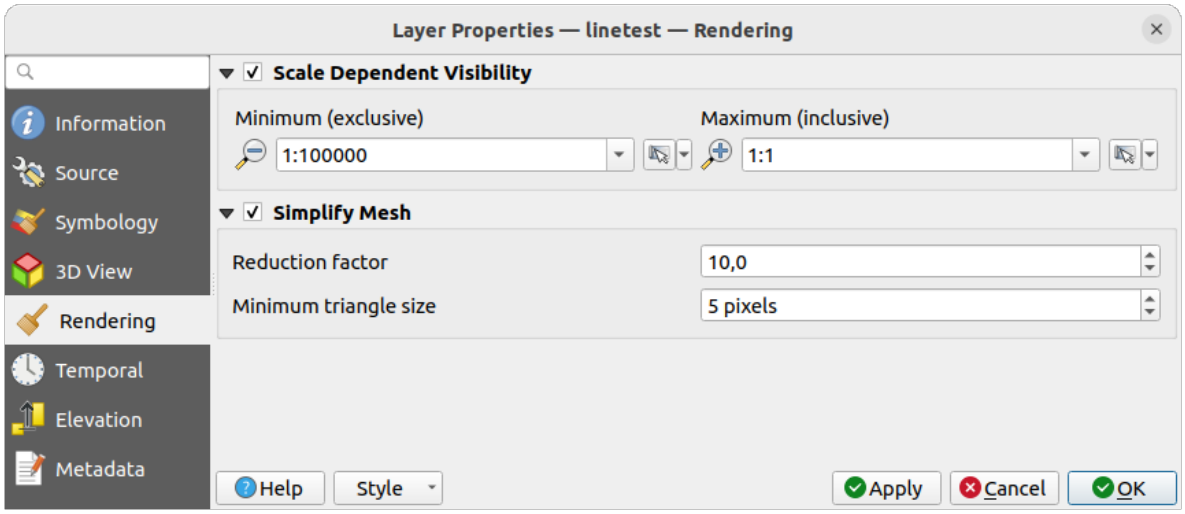


Figura14.13: Mesh rendering properties

Under the *Scale dependent visibility* group box, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale, defining a range of scale in which mesh elements will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Visibility Scale Selector* for more information.

Nota

You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

Como as camadas de malha podem ter milhões de faces, sua renderização às vezes pode ser muito lenta, especialmente quando todas as faces são exibidas na visualização, embora sejam muito pequenas para serem visualizadas. Para acelerar a renderização, você pode simplificar a camada de malha, resultando em uma ou mais malhas representando diferentes *níveis de detalhes* e selecionar em qual nível de detalhe você gostaria que o QGIS renderizasse a camada de malha. Observe que a malha simplificada contém apenas faces triangulares.

From the *Rendering* tab, check *Simplify mesh* and set:

- a *Reduction factor*: Controls generation of successive levels of simplified meshes. For example, if the base mesh has 5M faces, and the reduction factor is 10, the first simplified mesh will have approximately 500 000 faces, the second 50 000 faces, the third 5000,… If a higher reduction factor leads quickly to simpler meshes (i.e. with triangles of bigger size), it produces also fewer levels of detail.
- *Minimum triangle size*: the average size (in pixels) of the triangles that is permitted to display. If the average size of the mesh is lesser than this value, the rendering of a lower level of details mesh is triggered.

14.3.7 Propriedades Temporais

The *Temporal* tab provides options to control the rendering of the layer over time. It allows to dynamically display temporal values of the enabled dataset groups. Such a dynamic rendering requires the *temporal navigation* to be enabled over the map canvas.

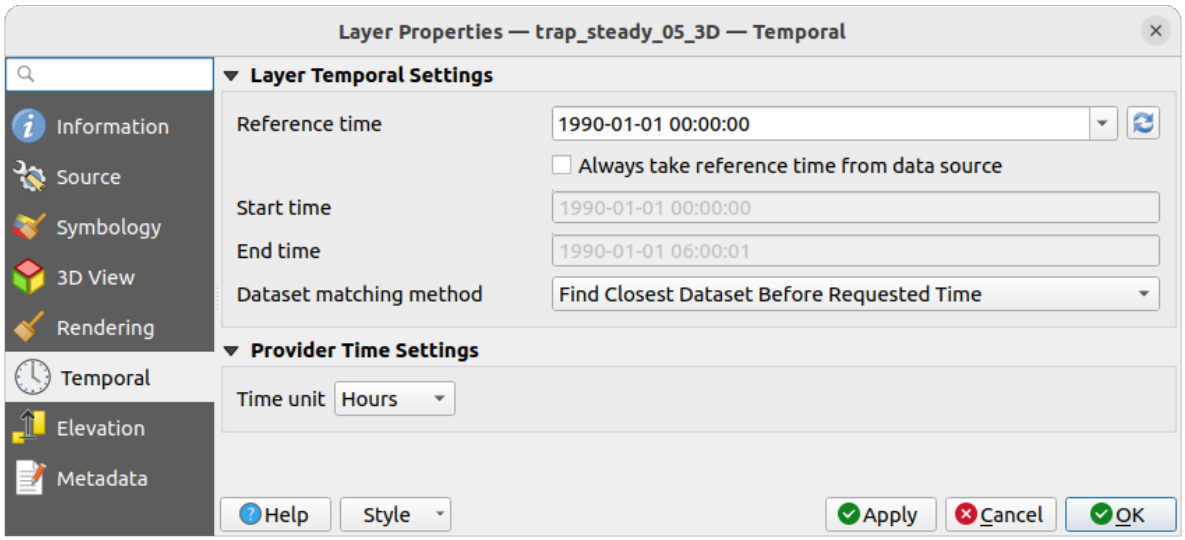


Figura14.14: Propriedades temporais da malha

Configurações da camada temporal

- *Reference time* of the dataset group, as an absolute date time. By default, QGIS parses the source layer and returns the first valid reference time in the layer's dataset group. If unavailable, the value will be set by the project time range or fall back to the current date. The *Start time* and *End time* to consider are then calculated based on the internal timestamp step of the dataset.

  It is possible to set a custom *Reference time* (and then the time range), and revert the changes using the Reload from provider button. With *Always take reference time from data source* checked, you ensure that the time properties are updated from the file each time the layer is reloaded or the project reopened.

- *Dataset matching method*: determines the dataset to display at the given time. Options are *Find closest dataset before requested time* or *Find closest dataset from requested time (after or before)*.

Provider time settings

- *Time unit* extracted from the raw data, or user defined. This can be used to align the speed of the mesh layer with other layers in the project during map time navigation. Supported units are *Seconds*, *Minutes*, *Hours* and *Days*.

14.3.8 Elevation Properties

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view* and *2D map view* and its appearance in the *profile tool charts*. Specifically, you can configure how heights from your dataset are interpreted:

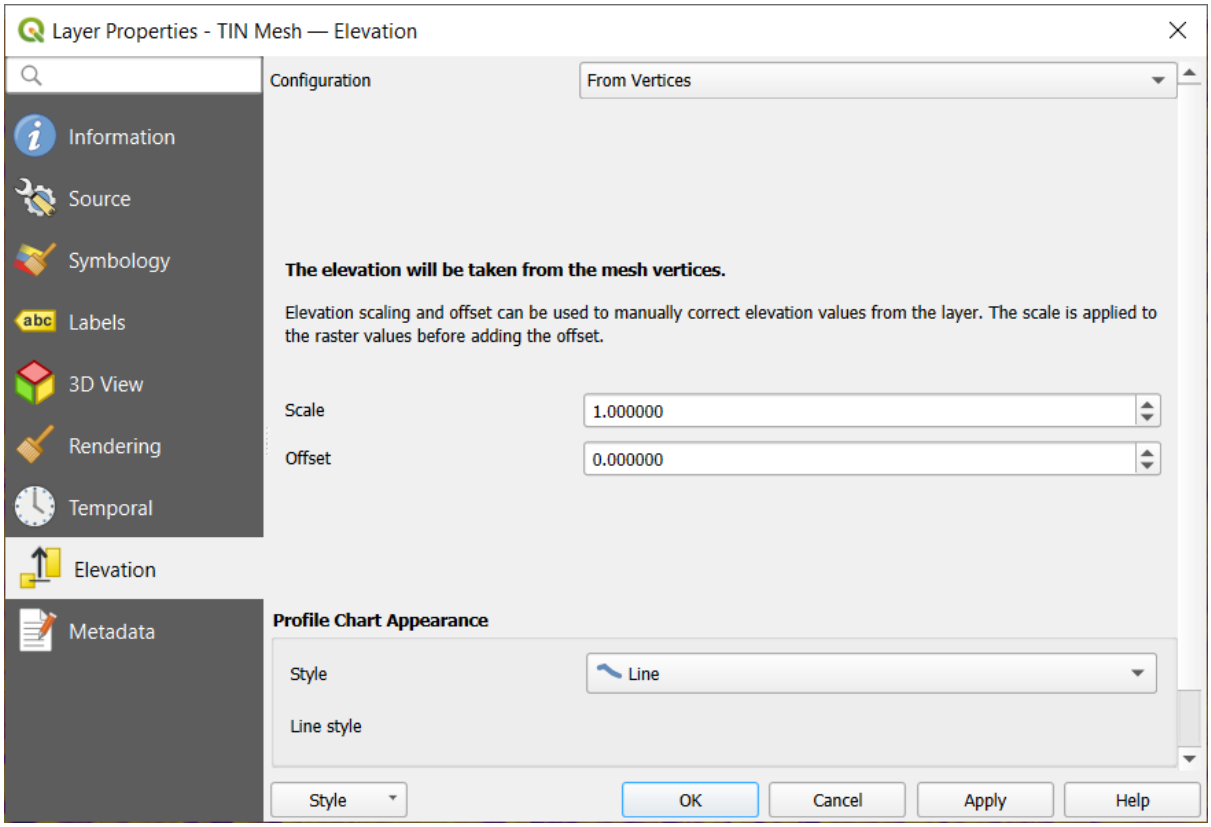


Figura14.15: Mesh Elevation properties

- *From vertices*: how the mesh layer vertices Z values should be interpreted as terrain elevation. You can apply a *Scale* factor and an *Offset*. This setting is available for *3D map view* and *profile tool charts*.
- *Fixed Elevation Range*: the mesh layer is linked to a fixed elevation range. This mode is applicable when a layer has either a single fixed elevation or a range (slice) of elevation values. If a range is specified, mesh values will be extruded over this range. You can set the *Lower* and *Upper* elevation range values for the layer, and specify whether the lower or upper *Limits* are inclusive or exclusive. When enabled, the layer will only be visible in *elevation filtered 2D maps* when the layer's range is included in the map's Z range.
- *Fixed Elevation Range Per Group*: each group in the mesh layer is associated with a fixed elevation range. This mode can be used when a layer has elevation data exposed through different dataset groups. This feature is exposed as a user-editable table for dataset groups with lower and upper values. You can either populate the lower and upper values manually or use an ε *Expression* to auto-fill all group values based on an expression. When enabled, the layer will be filtered in the *2D map view*, displaying only values within the map filter ranges. In *3D map view* or *profile tool charts*, the full extent of the layer will be displayed, ignoring the filtering.
- *Profile Chart Appearance*: controls the rendering of the mesh elements elevation in the profile chart. The profile *Style* can be set as:
  - a *Line* with a specific *Line style*
  - an elevation surface rendered using a fill symbol either above (*Fill above*) or below (*Fill below*) the elevation curve line. The surface symbology is represented using:
    - a *Fill style*
    - and a *Limit*: the maximum (respectively minimum) altitude determining how high the fill surface will be

14.3.9 Propriedades dos Metadados

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metadados* for more information.

14.4 Exploring mesh datasets

To identify mesh elements and their associated datasets, use the *Identify Features* tool from the *Attributes* toolbar. Click on a mesh element in the map canvas and the *Identify Results* panel opens, displaying information about the mesh layer and the identified element. The panel shows:

- Layer name
- Active dataset groups (scalar and/or vector) for the clicked element will be displayed first. On the other hand, inactive dataset groups are shown afterwards. For each group, the following derived information may also be shown, if applicable:
  - **Source** of the dataset (the file where it is stored).
  - **Time step** of the dataset value being displayed. For non-temporal dataset groups, this field remains empty.
  - **Vector x-component** and **Vector y-component** values for vector datasets.

Please note that when temporal navigation is disabled, the results only contain values from the *static dataset groups* defined.

- Geometry information shows the coordinates of the clicked point and may contain derived information in the following:
  - Centroid of the face.
  - Coordinates of a snapped vertex (if the click was snapped to a vertex).
  - Center of a snapped edge (if the click was snapped to an edge).

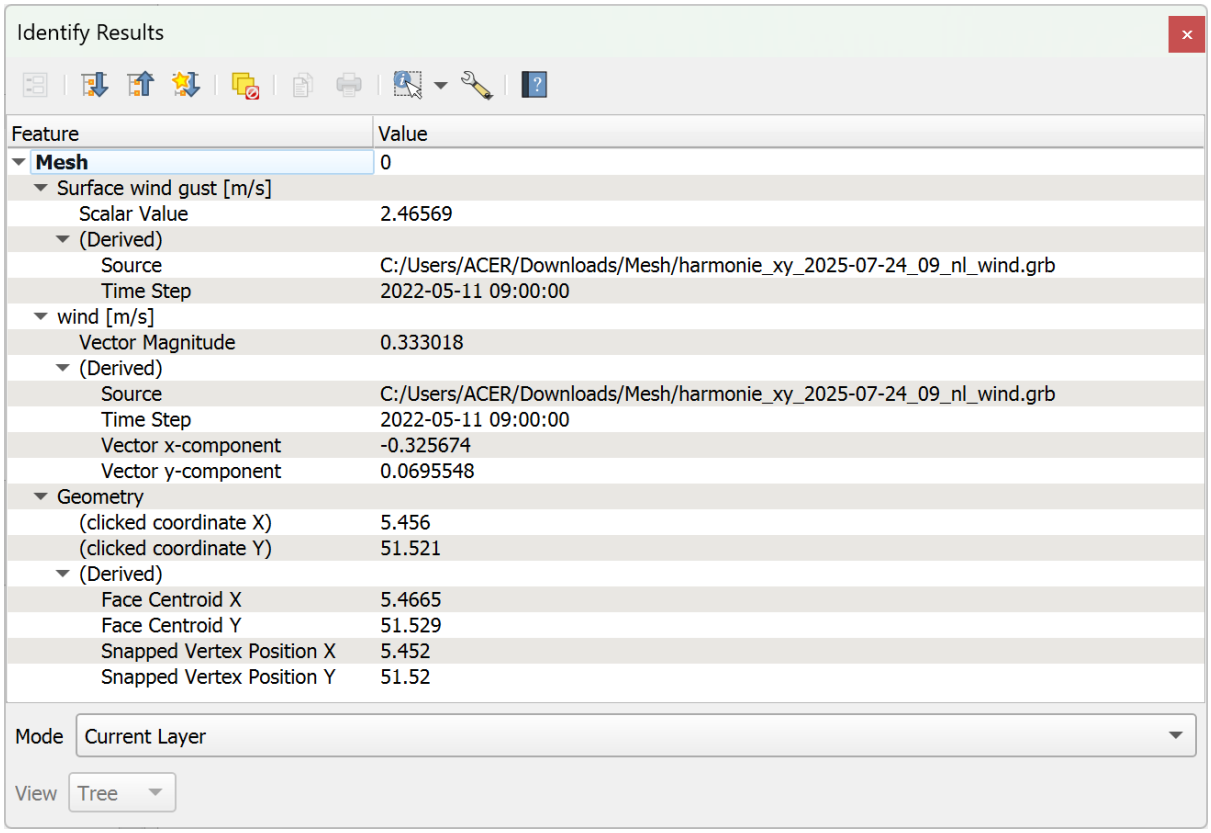


Figura14.16: Mesh Identify Results panel

14.5 Editando uma camada de malha

QGIS allows to *create a mesh layer* from scratch or based on an existing one. You can create/modify the geometries of the new layer whom you can assign datasets afterwards. It's also possible to edit an existing mesh layer. Because the editing operation requires a frames-only layer, you will be asked to either remove any associated datasets first (make sure you have them available if they still are necessary) or create a copy (only geometries) of the layer.

Nota

QGIS does not allow to digitize edges on mesh layers. Only vertices and faces are mesh elements that can be created. Also not all supported mesh formats can be edited in QGIS (see permissions).

14.5.1 Visão geral das ferramentas de digitalização de malhas

Para interagir com ou editar um elemento da camada base da malha, estão disponíveis as seguintes ferramentas.

Tabela14.4: Ferramentas para digitalização de malha

Label	Finalidade	Localização
Edições atuais	Acesso para salvar, reverter ou cancelar mudanças em todas as camadas ou em camadas selecionadas simultaneamente	*Digitizing* toolbar
Alternar para editar	Ativar/desativar a camada no modo de edição	*Digitizing* toolbar
Salvar Edições	Salvar as alterações feitas na camada	*Digitizing* toolbar
Desfazer	Desfazer a(s) última(s) mudança(s) - `Ctrl+Z`	*Digitizing* toolbar
Refazer	Refazer a(s) última(s) ação(ões) desfeita(s) - `Ctrl+Shift+Z`	*Digitizing* toolbar
Ativar ferramentas avançadas	Ligar/desligar o *Painel de Digitalização Avançada*	:guilabel:Ferramentas de *Digitalização avançada*
Reindex Faces and Vertices	Recrie o índice e renumere os elementos da malha para otimização	:guilabel:Menu *Malha*
Digitize Mesh Elements	Selecionar/criar vértices e faces	*Mesh Digitizing* toolbar
Select Mesh Elements by Pol	Selecionar vértices e faces sobrepostas por um polígono desenhado	*Mesh Digitizing* toolbar
Selecione Elementos de Malh	Selecionar vértices e faces usando uma expressão	*Mesh Digitizing* toolbar
Transformar Coordenadas do	Modificar coordenadas de uma seleção de vértices	*Mesh Digitizing* toolbar
Force by Selected Geometrie	Dividir faces e restringir o valor Z usando uma geometria linear	*Mesh Digitizing* toolbar

14.5.2 Explorando a lógica de atribuição de valor Z

When a mesh layer is turned into edit mode, a *Vertex Z value* widget opens at the top right of the map canvas. By default, its value corresponds to the *Default Z value* set in *Settings* ► *Options* ► *Digitizing* tab. When there are selected vertices, the widget displays the average Z value of the selected vertices.

During editing, the *Vertex Z value* is assigned to new vertices. It is also possible to set a custom value: edit the widget, press `Enter` and you will override the default value and make use of this new value in the digitizing process. Click the icon in the widget to reset its value to the Options default value.

Regras de atribuição

This is configured in the drop-down menu of the *Digitize mesh elements* tool as *New vertex Z value* option. Following methods are available:

Tabela14.5: Z value assignment methods

Method	Description
Project terrain	Z value is always taken or calculated from the *project reference terrain*. • If it is **Flat terrain**, the value corresponds to the *Terrain height* set in the project properties. • If it is a **Mesh** layer, the value is interpolated from the surrounding vertices of the mesh. • If it is a **DEM (raster layer)**, the value is assigned from the raster pixel value. • If there is no terrain under the new vertex, the value falls back to the *Default Z value* set in *digitizing options*.
Z widget	Z value is taken from the *Vertex Z value* widget or the *Advanced Digitizing Panel* *Z* widget (if it is in Locked state).
Prefer mesh, then Z widget	Interpolates the Z value from the active mesh layer if the vertex is on an edge or face, otherwise falls back to the *Vertex Z value* or *Advanced Digitizing Panel* *Z* widget (if it is in Locked state).
Prefer mesh, then terrain	Interpolates the Z value from the active mesh layer if the vertex is on the edge or face, otherwise uses the project reference terrain value.

The following detailed logic describes the behavior of the **Prefer mesh, then Z widget** strategy.

Tabela14.6: Matriz de atribuição de valor Z ao novo vértice

Criação de vértice	Vertices selected?	Fonte do valor atribuído	Valor Z atribuído
"Free" vertex, not connected to any face or edge of a face	Não	*Valor Z do vértice*	Padrão ou definido pelo usuário
		Advanced Digitizing Panel (if *z* widget is in Locked state)	*z* widget if in Locked state
	Sim	*Valor Z do vértice*	Average of the selected vertices or user-defined
Vértice em uma borda	—	Camada de malha	Interpolado a partir dos vértices da aresta
Vértice em uma face	—	Camada de malha	Interpolado a partir dos vértices da face
Vértice encaixado em uma feição vetorial 2D	—	*Valor Z do vértice*	Padrão ou definido pelo usuário
Vértice encaixado em um vértice vetorial 3D	—	Camada vetorial	Vértice
Vértice encaixado em um segmento vetorial 3D	—	Camada vetorial	Interpolado ao longo do segmento vetorial

Nota

The *Vertex Z value* widget is deactivated if the *Advanced Digitizing Panel* is enabled and no mesh element is selected. The latter's *z* widget then rules the Z value assignment.

Modificando o valor Z de vértices existentes

Para modificar o valor Z dos vértices, a maneira mais direta é:

1. Selecione um ou vários vértices. O elemento *Vértice valor Z* exibirá a altura média da seleção.
2. Alterar o valor do elemento.
3. Pressione `Enter`. O valor inserido é atribuído aos vértices e se torna o valor padrão dos próximos vértices.

Outra maneira de alterar o valor Z de um vértice é movê-lo e encaixá-lo em uma característica de camada vetorial com a capacidade do valor Z. Se mais de um vértice for selecionado, o valor Z não pode ser alterado desta forma.

The *Transform mesh vertices* dialog also provides means to modify the Z value of a selection of vertices (along with their X or Y coordinates).

14.5.3 Selecionando elementos de malha

To select mesh elements, you can use the following tools:

- Digitize Mesh Elements to select different mesh elements, see more at *Usando Digitalização de elementos de malha*.
- Select Mesh Elements by Polygon to select different mesh elements by polygon, see more at *Usando Selecionar elementos de malha por polígono*.
- Select Mesh Elements by Expression to select different mesh elements by expression, you can choose to *Select by vertices* or *Select by faces*, see more at *Usando Selecione Elementos de Malha por Expressão*.
- Select Isolated Vertices to select all vertices that are not part of any mesh face.
- Select All Vertices to select all vertices of the mesh layer.

Usando *Digitalização de elementos de malha*

Activate the Digitize Mesh Elements tool. Hover over an element and it gets highlighted, allowing you to select it.

- Clique em um vértice e ele será selecionado.
- Clique no pequeno quadrado no centro de uma face ou borda, e ele é selecionado. Os vértices conectados também são selecionados. Por outro lado, selecionando todos os vértices de uma borda ou de uma face também seleciona esse elemento.
- Arraste um retângulo para selecionar elementos sobrepostos (uma face selecionada vem com todos os seus vértices). Pressione `Alt` se você quiser selecionar apenas elementos completamente contidos.
- Para adicionar elementos a uma seleção, pressione `Shift` enquanto os seleciona.
- Para remover um elemento da seleção, pressione `Ctrl` e selecione-o novamente. Uma face desmarcada também desmarcará todos os seus vértices.

Usando *Selecionar elementos de malha por polígono*

Ativar a ferramenta Selecione Elementos de Malha por Polígono e:

- Desenhe um polígono (clique esquerdo para adicionar vértice, `Backspace` para desfazer o último vértice, `Esc` para abortar o polígono e clique direito para validá-lo) sobre as geometrias da malha. Quaisquer vértices e faces parcialmente sobrepostos serão selecionados. Pressione `Alt` enquanto desenha, se você quiser selecionar apenas elementos completamente contidos.
- Clique com o botão direito do mouse sobre a geometria de uma camada vetorial, selecione-a na lista que aparece e quaisquer vértices e faces parcialmente sobrepostas da camada de malha serão selecionados. Utilize `Alt` enquanto desenha para selecionar apenas os elementos completamente contidos.
- Para adicionar elementos a uma seleção, pressione `Shift` enquanto os seleciona.
- Para remover um elemento da seleção, pressione `Ctrl` enquanto desenha sobre o polígono de seleção.

Usando *Selecione Elementos de Malha por Expressão*

Another tool for mesh elements selection is Select Mesh Elements by Expression. When pressed, the tool opens the mesh *expression selector dialog* from which you can:

1. Selecione o método de seleção:
   - *Selecionar por vértices*: aplica a expressão entrada aos vértices, e retorna os correspondentes e suas eventuais bordas/face associadas
   - guilabel:*Selecionar por faces*: aplica a expressão entrada às faces, e retorna as que correspondem e suas bordas/vértices associados
2. Write the expression of selection. Depending on the selected method, available functions in the *Meshes group* will be filtered accordingly.
3. Execute a consulta definindo como a seleção deve se comportar e pressione:
   - *Selecionar*: substitui qualquer seleção existente na camada
   - *Adicionar à seleção atual*
   - *Remover da seleção atual*

14.5.4 Modificando elementos de malha

Adicionando vértices

Para adicionar vértices a uma camada de malha:

1. Pressione o :sup:Botão *Digitalizar elementos da malha*
2. A *Vertex Z value* widget is always available on the top right corner of the map canvas during digitizing. It allows setting or overriding the Z coordinate assigned to new vertices, depending on the selected *Z value assignment method*.
3. You can choose other method for assigning Z values to new vertices from the drop-down menu in the Digitize mesh elements tool.
4. Em seguida, clique duas vezes:
   - fora de uma face: adiciona um “vértice livre”, que é um vértice não vinculado a nenhuma face. Este vértice é representado por um ponto vermelho quando a camada está em modo de edição.

- on the edge of existing face(s): adds a vertex on the edge, splits the touching face(s) into triangles connected to the new vertex.
- dentro de uma face: divide a face em triângulos cujas arestas conectam os vértices circundantes ao novo vértice.

  With the *Refine neighboring faces when adding vertices* option enabled in the Digitize mesh elements button drop-down menu, a check is applied to triangular faces that share at least one vertex with the face the new vertex is added to. If their edges do not satisfy Delaunay triangulation rules, they are flipped accordingly.

Adicionando faces

Para adicionar faces a uma camada de malha:

1. Pressione o :sup:Botão *Digitalizar elementos da malha*
2. A *Vertex Z value* widget appears on the top right corner of the map canvas. Set this value to the Z coordinate you would like to assign to the subsequent vertices.
3. Hover over a vertex and click the small triangle that appears next to it.
4. Move the cursor to the next vertex position; you can snap to existing vertex or left-click to add a new one.
5. Proceed as above to add as many vertices as you wish for the face. Press `Backspace` button to undo the last vertex.
6. While moving the mouse, a rubberband showing the shape of the face is displayed. If it is shown in green, then the expected face is valid and you can right-click to add it to the mesh. If it is red, the face is not valid (e.g. because it self-intersects, overlaps an existing face or vertex, creates a hole, …) and can't be added. You'd need to undo some vertices and fix the geometry.
7. Pressione `Esc` para abortar a digitalização da face.
8. Clique com o botão direito do mouse para validar a face.

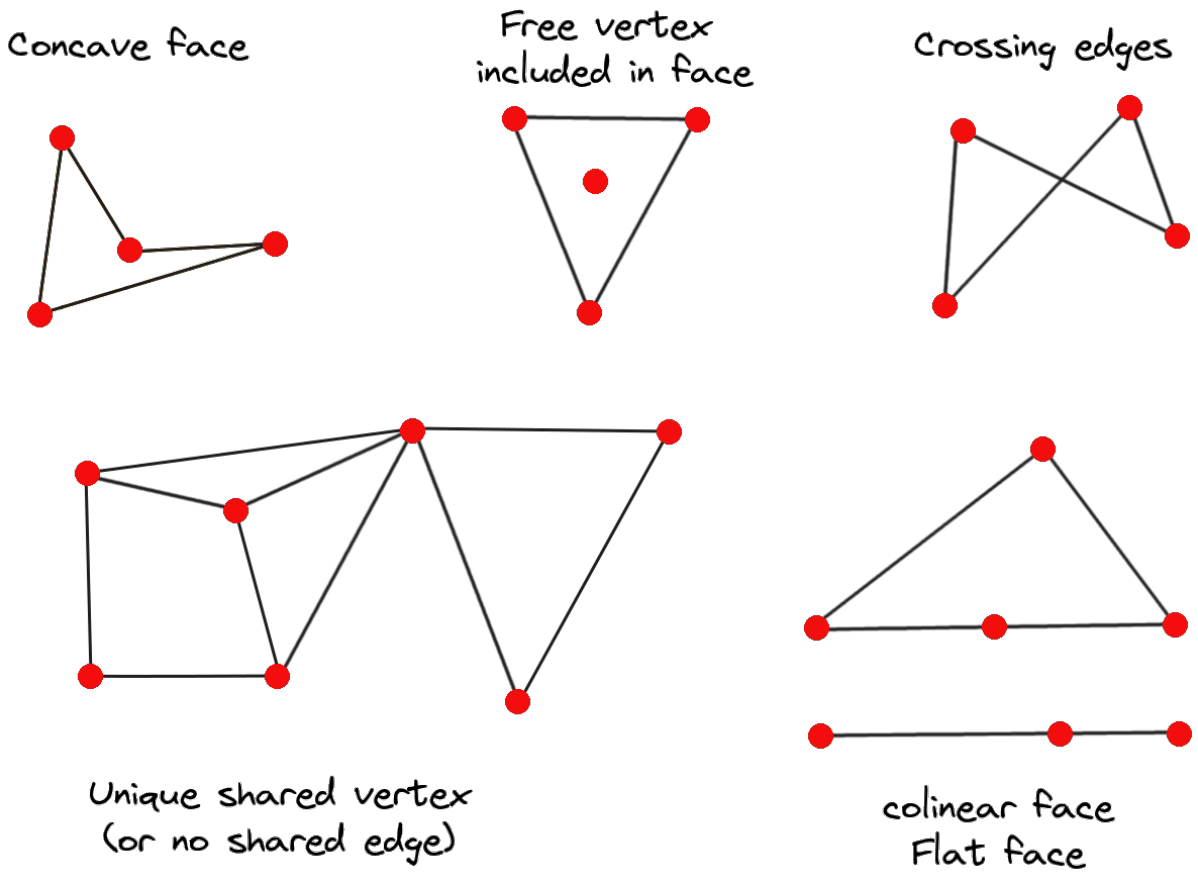


Figura14.17: Examples of invalid mesh

Removendo elementos de malha

1. ref:*Selecionar os Elementos Alvo <select_mesh_elements>*
2. Habilitar a :sup:Ferramenta *Digitalizar elementos da malha*
3. Clique com o botão direito do mouse e selecione:
   - *Remover os vértices selecionados e preencher o(s) buraco(s) ou pressione :kbd:`Ctrl+Del`*: remove os vértices e as faces ligadas e preenche o(s) buraco(s) triangular(es) a partir dos vértices vizinhos
   - guilabel:*Remover Vértices Selecionados Sem Furo(s) de Preenchimento(s) ou pressione :kbd:`Ctrl+Shift+Del`*: remove os vértices e as faces ligadas e não preenche o(s) furo(s)
   - guilabel:*Remover a(s) face(s)selecionada(s)* ou pressione `Shift+Del`: remove as faces mas mantém os vértices

Essas opções também estão acessíveis no menu contextual ao passar o mouse sobre um único item sem selecionar.

Movendo elementos de malha

Para mover vértices e faces de uma camada de malha:

1. ref:*Selecionar os Elementos Alvo <select_mesh_elements>*
2. Habilitar a :sup:Ferramenta *Digitalizar elementos da malha*
3. Para começar a mover o elemento, clique em um vértice ou no centroide de uma face/aresta
4. Mova o cursor para o local de destino (é suportado a adesão para características vetoriais).
5. If the new location does not generate an *invalid mesh*, the moved elements appear in green. Click again to release them at this location. Faces whose vertices are all selected are translated, their neighbors are reshaped accordingly.

Transformando vértices de malha

A ferramenta Transformar Coordenadas de Vértices oferece uma maneira mais avançada de mover vértices, editando suas coordenadas X, Y e/ou Z graças a expressões.

1. Selecione os vértices que deseja editar as coordenadas
2. Press Transform Vertices Coordinates. A dialog opens with a mention of the number of selected vertices. You can still add or remove vertices from the selection.
3. Dependendo das propriedades que você deseja modificar, você precisa verificar a *coordenada X*, :guilabel:` coordenada Y` e/ou *valor Z*.
4. Em seguida, digite a posição alvo na caixa, seja como um valor numérico ou uma expressão (utilizando a **|expressão|** diálogo de expressão)
5. With the Import Coordinates of the Selected Vertex pressed, the X, Y and Z boxes are automatically filled with its coordinates whenever a single vertex is selected. You can adjust Z value by using *Get the Z value from the project terrain* button and in that case the Z box is deactivated. A convenient and quick way to adjust vertices individually.
6. Pressione *Pré-visualização da transformação* para simular a nova localização dos vértices e pré-visualizar a malha com transformação.
   - Se a pré-visualização estiver verde, a malha transformada é válida e você pode aplicar a transformação.
   - Se a pré-visualização estiver vermelha, a malha transformada é inválida e você não poderá aplicar a transformação até que ela seja corrigida.
7. Pressione *Aplicar Transformação* para modificar as coordenadas selecionadas para o conjunto de vértices.

Remodelando a geometria da malha

O menu contextual

1. Habilitar o Digitalizar elementos da malha
2. Selecione o(s) item(ns) de malha, ou não
3. Passe o mouse sobre um elemento de malha, ele fica destacado.
4. Clique com o botão direito do mouse e você pode:
   - ref:*remove o(s) item(ns) <remove_mesh_items>*
   - *Dividir face(s) selecionada(s)* (:guilabel:`Dividir Face Atual'): divide a face que você está pairando sobre ou cada face de malha quadrangular selecionada em dois triângulos

- :guilabel:`Triangulação Delaunay com vértices selecionados': constrói faces triangulares utilizando vértices livres selecionados.
- guilabel:*Refinar a(s) face(s) selecionada(s)* (:guilabel:`Refinar a face atual'): divide a face em quatro faces, com base nos vértices adicionados no meio de cada borda (um triângulo resulta em triângulos, um quad em quads). Também triangula as faces adjacentes ligadas aos novos vértices.

Os marcadores de borda

Quando o Digitar elementos de malha está ativo e você paira sobre uma borda, a borda é destacada e é possível interagir com ela. Dependendo do contexto, os seguintes marcadores podem estar disponíveis:

- um **quadrado**, no centro da aresta: clique nele para selecionar os vértices das extremidades.
- uma **cruz** se as duas faces de cada lado puderem ser fundidas: clique sobre ela para apagar a borda e fundir as faces.
- um **círculo** se a borda estiver entre dois triângulos: Clique sobre ele para inverter a borda, ou seja, conecte-o aos dois outros vértices "livres" das faces

A ferramenta :guilabel:`Força por Geometrias Selecionadas'

A ferramenta Forcar por geometrias selecionadas fornece formas avançadas de aplicar linhas de quebra utilizando a geometria de linhas. Uma linha de quebra forçará a malha a ter bordas ao longo da linha. Note que a linha de quebra não será considerada persistente uma vez terminada a operação; as bordas resultantes não atuarão mais como restrições e podem ser modificadas como qualquer outra borda. Isto pode ser usado, por exemplo, para modificar localmente uma camada de malha com linhas precisas, como margens de rios ou margens de aterros de estradas.

1. Habilitar a ferramenta :sup:`Força por Geometrias Selecionadas'
2. Indicar a geometria a ser usada como "linha de força"; ela pode ser:
   - escolhida a partir de uma linha ou polígono na tela do mapa: clique com o botão direito sobre o vetor e selecione-o na lista do menu contextual.
   - a virtual line drawn over the mesh frame: left-click to add vertices, right-click for validation. Vertices Z value is set through the *Vertex Z value* widget or the *z* widget if the *Advanced Digitizing Panel* is on. If the line is snapped to a mesh vertex or a 3D vector feature's vertex or segment, the new vertex takes the snapped element Z value.

Mesh faces that overlap the line geometry or the polygon's boundary will be affected in a way that depends on options you can set from the Force by Selected Geometries tool drop-down menu:

- :guilabel:`Adicionar um novo vértice nas bordas de interseção': com esta opção, um novo vértice é adicionado cada vez que a linha forçadora intercepta uma borda. Esta opção leva à divisão ao longo da linha de cada face encontrada.

  Sem esta opção, as faces encontradas são removidas e substituídas por faces provenientes de uma triangulação com apenas os vértices existentes mais os vértices das linhas de forçagem (novos vértices também são adicionados na borda da fronteira que intercepta as linhas de forçagem).

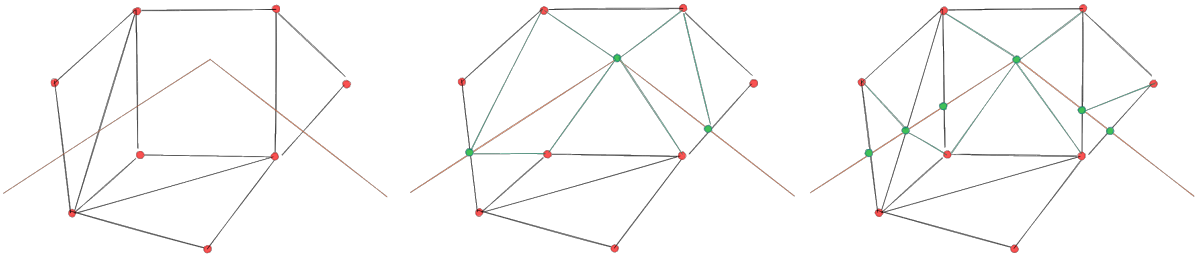


Figura14.18: Forçar a malha usando uma geometria de linha - Resultados sem (meio) e com (direita) novo vértice na interseção das bordas

- guilabel:`Interpolar o valor Z de': definir como é calculado o novo valor Z dos vértices. Pode ser de:
  - the *Mesh* itself: the new vertices Z value is interpolated from vertices of the face they fall within
  - ou a *Linha forçada*: se a linha é definida por uma característica vetorial 3D ou uma linha desenhada, então o novo valor dos vértices Z é derivado de sua geometria. No caso da característica de linha 2D, o novo valor Z dos vértices é o valor *Valor do vértice Z*.
- *Tolerância*: quando um vértice de malha existente estiver mais próximo da linha do que o valor de tolerância, não crie um novo vértice na linha, mas utilize o vértice existente em seu lugar. O valor pode ser definido em *Medidores em Escala* ou em *Unidades do mapa* (mais detalhes em *Seletor de Unidade*).

14.5.5 Reindexação de malhas

During edit, and in order to allow quick undo/redo operations, QGIS keeps empty places for deleted elements, which may lead to growing memory use and inefficient mesh structuring. The *Mesh* ► *Reindex Faces and Vertices* tool is designed to remove these holes and renumber the indices of faces and vertices so that they are continuous and somewhat reasonably ordered. This optimizes relation between faces and vertices and increases the efficiency of calculation.

Nota

A ferramenta :guilabel:`Reindexar Faces e Vértices' salva a camada e limpa as pilhas de desfazer/redo, desabilitando qualquer rollback.

14.6 Calculadora de Malha

The *Mesh Calculator* tool from the top *Mesh* menu allows you to perform arithmetic and logical calculations on existing dataset groups to generate a new dataset group (see Fig.14.19).

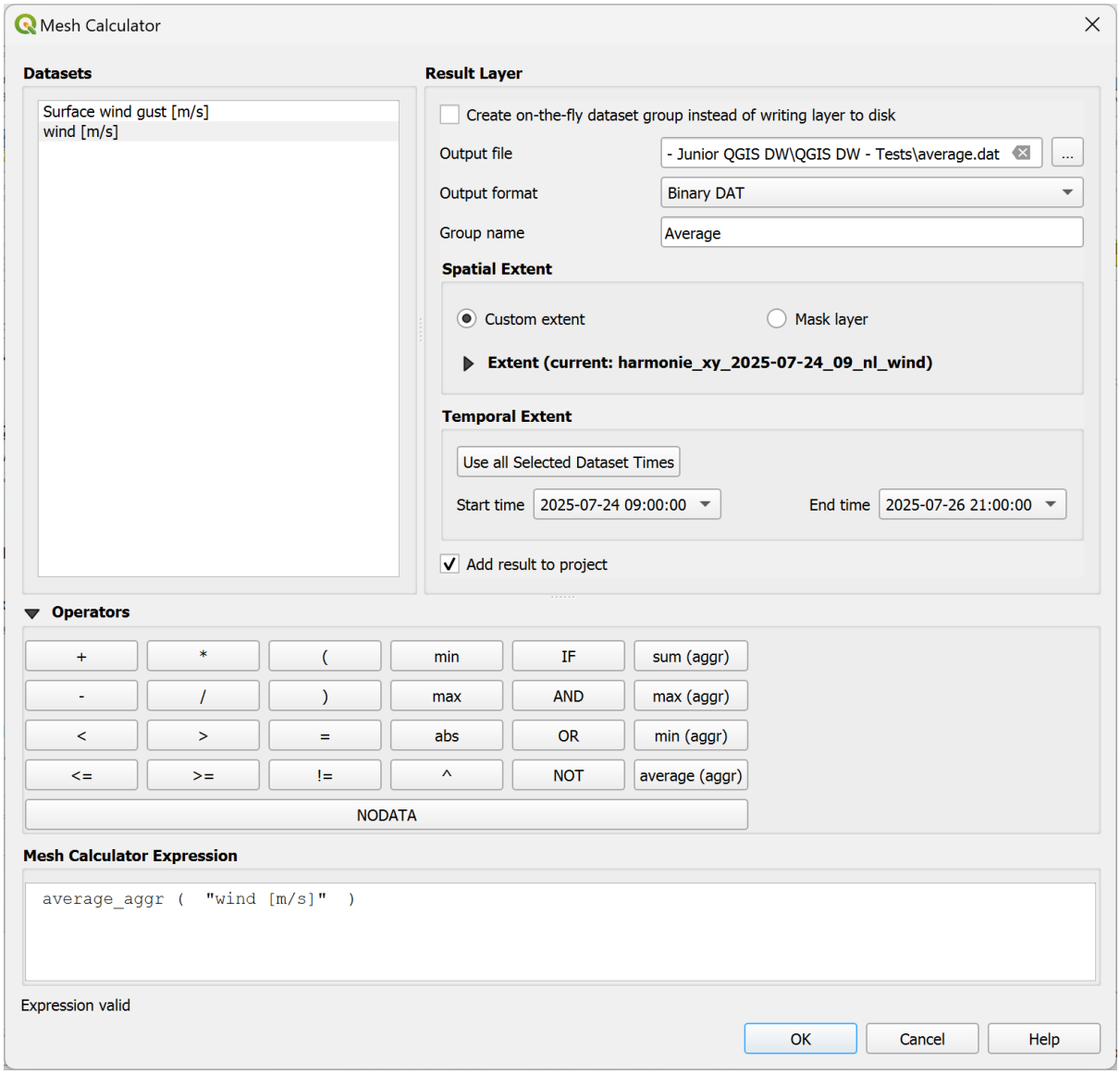


Figura14.19: Calculadora de Malha

The *Datasets* list contains all dataset groups in the active mesh layer. To use a dataset group in an expression, double click its name in the list and it will be added to the *Mesh calculator expression* field. You can then use the operators to construct calculation expressions, or you can just type them into the box.

The *Result Layer* helps you configure properties of the output layer:

- *Create on-the-fly dataset group instead of writing layer to disk*:
  - If unchecked, the output is stored on disk as a new plain file. An *Output File* path and an *Output Format* are required.
  - If checked, a new dataset group will be added to the mesh layer. Values of the dataset group are not stored in memory but each dataset is calculated when needed with the formula entered in the mesh calculator.

That virtual dataset group is saved with the project, and if needed, it can be removed or made persistent in file from the layer *Source* properties tab.

In either case, you should provide a *Group Name* for the output dataset group.

- The *Spatial extent* to consider for calculation can be:
  - defined as a *Custom extent* using the *extent selector* widget.
  - defined by a polygon layer (*Mask layer*) of the project: the polygon features geometry are used to clip the mesh layer datasets
- The *Temporal extent* to take into account for datasets can be set with the *Start time* and *End time* options, selected from the existing dataset groups timesteps. They can also be filled using the *Use all selected dataset times* button to take the whole range.
- With the *Add result to project* checkbox, the result layer will automatically be added to the legend area and can be visualized. Checked by default.

The *Operators* section contains all available operators. To add an operator to the mesh calculator expression box, click the appropriate button. Mathematical calculations (`+`, `-`, `*`, ...) and statistical functions (`min`, `max`, `sum (aggr)`, `average (aggr)`, ...) are available. Conditional expressions (`=`, `!=`, `<`, `>=`, `IF`, `AND`, `NOT`, ...) return either 0 for false and 1 for true, and therefore can be used with other operators and functions. The `NODATA` value can also be used in the expressions.

The *Mesh Calculator Expression* widget shows and lets you edit the expression to execute.

CAPÍTULO 15

Trabalhando com Blocos Vetoriais

15.1 O que são Blocos Vetoriais?

Vector tiles are packets of geographic data, packaged into pre-defined roughly-square shaped “tiles” for transfer over the web. They combine pre-rendered raster map tiles and vector map tiles. The vector tile server returns vector map data, which has been clipped to the boundaries of each tile, instead of a pre-rendered map image. The clipped tiles represent the zoom-levels of the vector tile service, derived from a pyramid approach. Using this structure, the data-transfer is reduced in comparison to un-tiled vector maps. Only data within the current map view, and at the current zoom level need to be transferred. Also, compared to a tiled raster map, data transfer is also greatly reduced, as vector data is typically much smaller than a rendered bitmap. Vector tiles do not have any styling information assigned so QGIS needs to apply a cartographic style in order to display the data.

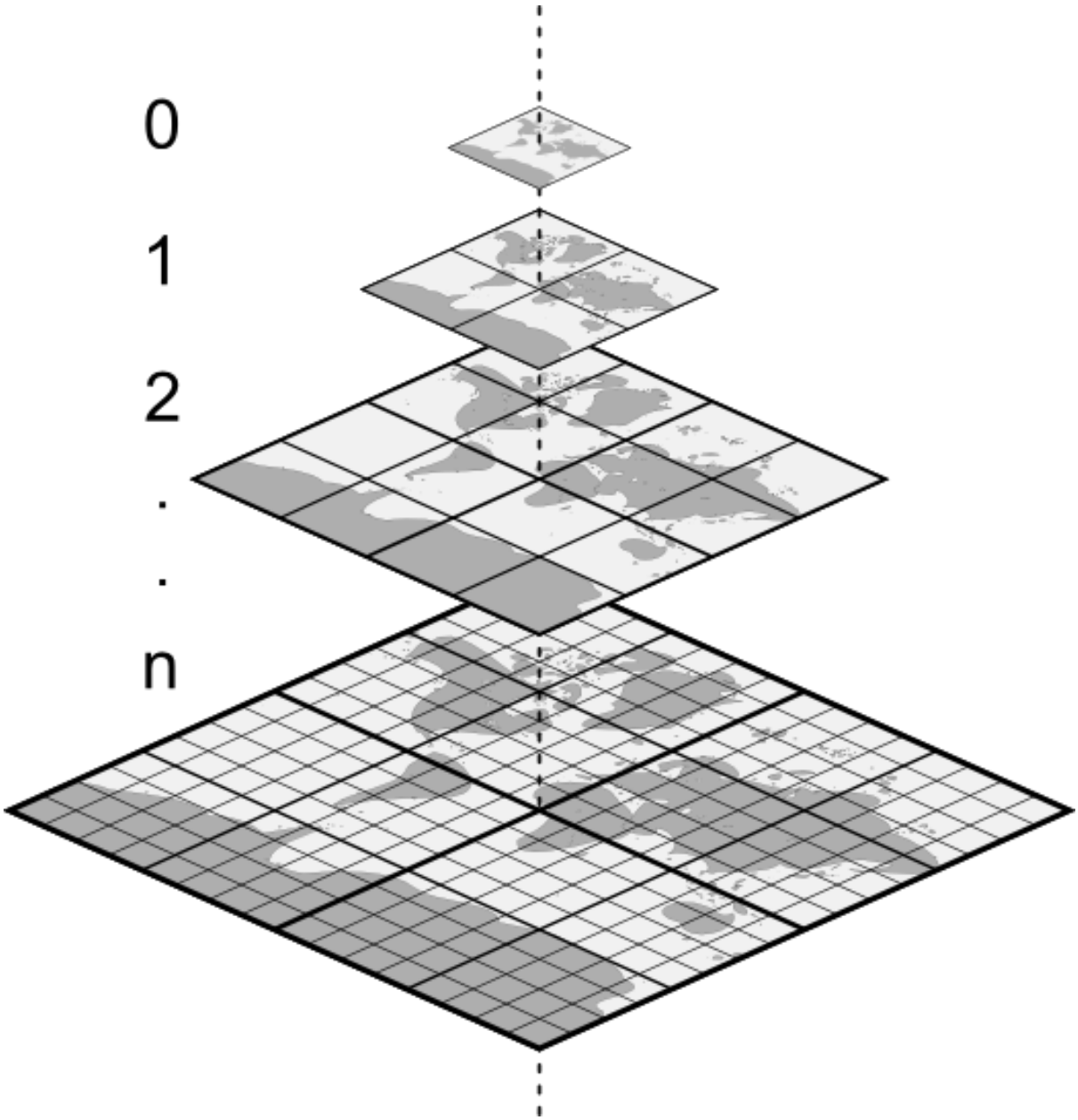


Figura15.1: Pyramid structure of vector tiles with zoom-levels

15.2 Formatos Suportados

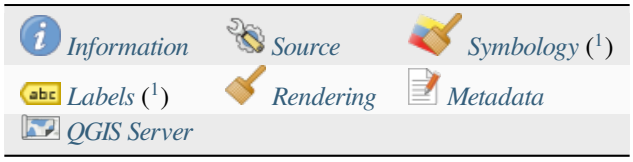
There is support for vector tiles through:

- remote sources (HTTP/S) - with XYZ template - `type=xyz&url=http://example.com/{z}/{x}/{y}.pbf`
- local files - with XYZ template - e.g. `type=xyz&url=file:///path/to/tiles/{z}/{x}/{y}.pbf`
- local MBTiles database - e.g. `type=mbtiles&url=file:///path/to/file.mbtiles`

To load a vector tiles dataset into QGIS, use the *Vector Tile* tab in the *Data Source Manager* dialog. Read *Using Vector Tiles services* for more details.

15.3 Vector Tiles Dataset Properties

The vector tiles *Layer Properties* dialog provides the following sections:



15.3.1 Propriedades da informação

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- based on the provider of the layer: name, URI, source type and path, number of zoom levels
- custom properties, used to store in the active project additional information about the layer. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- the Coordinate Reference System: name, units, method, accuracy, reference (i.e., whether it is static or dynamic)
- picked from the *filled metadata*: access, extents, links, contacts, history…

15.3.2 Propriedades de origem

The *Source* tab displays basic information about the selected vector tile, including:

- o *Nome da Camada* para ser mostrado no *Painel de Camadas*;
- the *Coordinate Reference System*: Displays the layer's *Coordinate Reference System (CRS)*. You can change the layer's CRS, by selecting a recently used one in the drop-down list or clicking on the Select CRS button (see *Seletor do Sistemas de Referência de Coordenadas*). Use this process only if the layer CRS is wrong or not specified.
- For file based vector tiles (`.vtpk` or `.mbtiles`), a *Layer source* group indicates the path to the source of the dataset and allows for replacing the loaded layer. Edit the path shown in the text box or press … Browse to select another file on the disk.

[1] Also available in the *Layer styling panel*

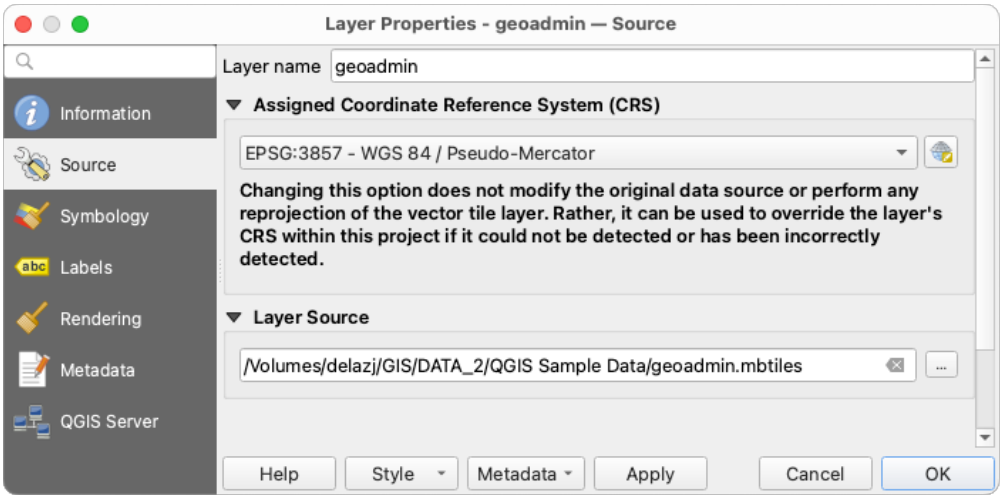


Figura15.2: Vector Tiles Properties - Source Dialog

15.3.3 Symbology and Labels Properties

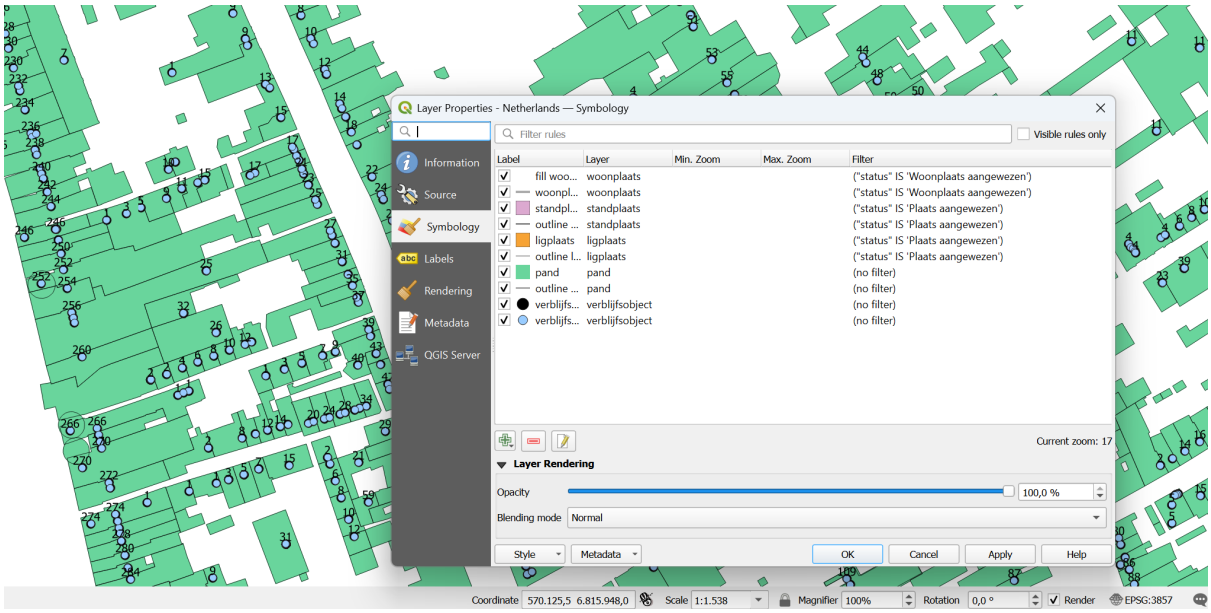


Figura15.3: Vector Tile Layer Symbology

Setting rules

As vector tiles consist of point, line and polygon geometries, the respective symbols are available. To apply a cartographic style (with symbology and/or labels), you can either:

- Use a *Style URL* when creating the *Vector Tiles Connection*. The symbology will be shown immediately in the *Symbology* tab after the layer is loaded in QGIS.
- Or build your own symbology and labeling in the corresponding tabs of the layer properties. By default, QGIS assigns an identical symbol to the features based on their geometry type.

In both cases, setting a style for a vector tile relies on a set of *rules* applied to the features, indicating:

- a vector *symbol* or *label*

- a *Label*, a title for comprehensive identification of the rule
- the name of a particular *Layer* the rule should apply to, if not applied to `(all layers)`
- a *Min. Zoom* and a *Max. Zoom*, for the range of display. Symbology and labeling can be dependent on the zoom level.
- a *Filter*, a QGIS expression to identify the features to apply the style to

Each rule is added pressing the Add rule button and selecting the type of symbols (*Marker*, *Line*, *Fill*) corresponding to the features geometry type. You can as well Remove selected rules or Edit current rule.

At the bottom the *Current Zoom* is shown. Check the *Visible rules only* option at the top of the dialog to filter the list of rules to only those that are visible at the current zoom level. This makes it easier to work with complex vector styling and to locate troublesome rules. The *Filter rules* text box also helps you easily find a rule, by searching the *Label*, *Layer* and *Filter* fields.

In Fig.15.3 we set up style for the OpenStreetMap `landuse` layer. For better visibility most of the rules are deselected.

Renderização da camada

From the *Symbology* tab, you can also set some options that invariably act on all features of the layer:

- *Opacity*: You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your vector layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Blending mode*: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlaying layers are mixed through the settings described in *Modos de Mistura*.

Estilos

Available at the bottom of most of the tabs, the *Styles* ► menu provides shortcuts to save, load, create, switch styles to apply to the vector tiles. Vector tiles can have their style saved from QGIS as *QML* files and they can be imported as:

- *QML* arquivos (*QML - O Formato de Arquivo de Estilo QGIS*)
- *MapBox GL Json* style configuration files

More details at *Salvar e Compartilhar Propriedades da Camada*.

15.3.4 Propriedades de Renderização

Under *Scale dependent visibility*, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Visibility Scale Selector* for more information.

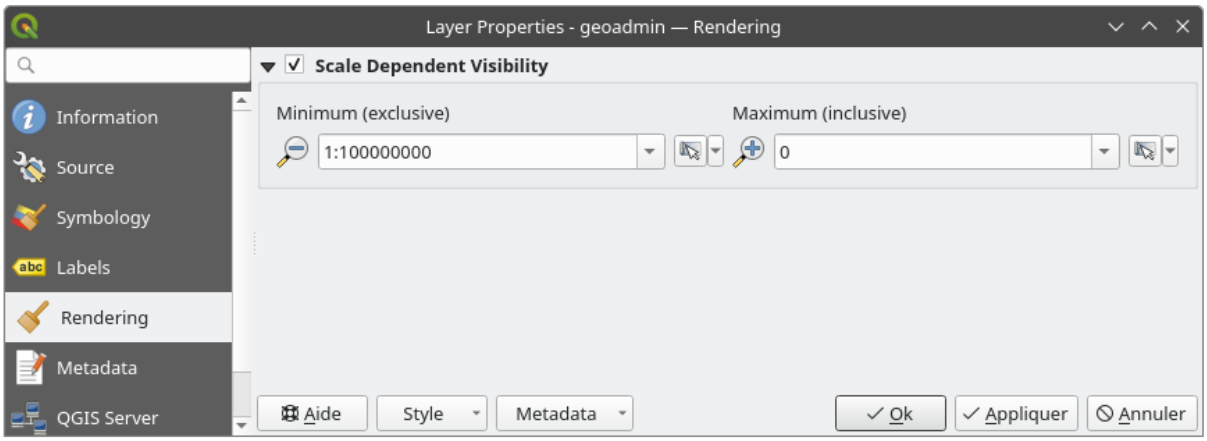


Figura15.4: Vector Tiles Properties - Rendering Dialog

15.3.5 Propriedades dos Metadados

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metadados* for more information.

15.4 QGIS Server Properties

The *QGIS Server* tab helps you configure settings of the data when published by QGIS Server. The configuration concerns:

- *Description*: provides information to describe the data, such as *Short name*, *Title*, *Abstract*, a *Keyword list*, and a *Data URL* whose *Format* can be in `text/html`, `text/plain` or `application/pdf`.
- *Attribution*: a *Title* and *URL* to identify who provides the data
- *Metadata URL*: a list of *URL* for the metadata that can be of `FGDC` or `TC211` *Type*, and in `text/plain` or `text/xml` *Format*
- *Legend URL*: a *URL* for the legend, in either `image/png` or `image/jpeg` *Format*

CAPÍTULO 16

Working with Point Clouds

16.1 Introduction to Point Clouds

What is A Point Cloud?

A point cloud is a three-dimensional image of a space made up of many individual of data points (up to billions, even trillions). Each of the points has an x, y and z coordinate. Depending on the capture method, point clouds usually also have additional attributes that came from the capture, such as color values or intensity. These attributes can be used, for instance, to display point clouds in different colors. In QGIS, a point cloud can be used to generate a three-dimensional image of the landscape (or of another space).

Supported Formats

QGIS supports the data formats Entwine Point Tile (EPT) and LAS/LAZ. To work with point clouds, QGIS always saves the data in EPT. EPT is a storage format that consists of several files stored in a common folder. To allow quick access to the data, EPT uses indexing. For more information on the EPT format, see entwine homepage

If the data is in LAS or LAZ format, QGIS will convert it to EPT when it is loaded for the first time. Depending on the size of the file, this may take some time. In this process, a subfolder is created in the folder in which the LAS/LAZ file is located according to the scheme `ept_` + `name_LAS/LAZ_file`. If such a subfolder already exists, QGIS loads the EPT immediately (which leads to a reduced loading time).

Worth Knowing

In QGIS it is not (yet) possible to edit point clouds. If you want to manipulate your point cloud, you can use CloudCompare, an open source point cloud processing tool. Also the Point Data Abstraction Library (PDAL - similar to GDAL) offers you options to edit point clouds (PDAL is command line only).

Due to the large number of data points, it is not possible to display an attribute table of point clouds in QGIS. However, the *Identify tool* supports point clouds, so you can display all attributes, even of a single data point.

If you want to create a new layer, in the same or another supported format, from an existing point cloud layer, see *Creating new layers from an existing layer*.

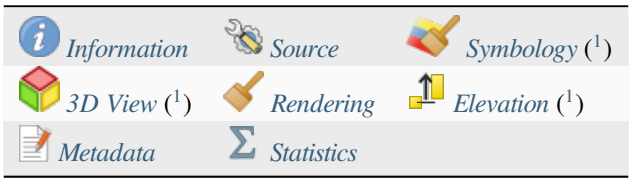
16.2 Point Clouds Properties

The *Layer Properties* dialog for a point cloud layer offers general settings for the layer and its rendering. It also provides information about the layer.

Para acessar a caixa de diálogo:guilabel:*Propriedades da camada*:

- In the *Layers* panel, double-click the layer or right-click and select *Properties…* from the context menu;
- Vá para: seleção de menu: menu Camada -> Propriedades da camada … `quando a camada estiver selecionada.

The point cloud *Layer Properties* dialog provides the following sections:



Nota

Most of the properties of a point cloud layer can be saved to or loaded from a `.qml` file using the *Style* menu at the bottom of the properties dialog. More details at *Salvar e Compartilhar Propriedades da Camada*

16.2.1 Propriedades da informação

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- General such as name in the project, source path, last save time and size, the used provider
- custom properties, used to store in the active project additional information about the layer. Default custom properties may include *layer notes*. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- Based on the provider of the layer: extent and number of points
- The Coordinate Reference System: name, units, method, accuracy, reference (i.e. whether it's static or dynamic)
- Metadata delivered by the provider: creation date, version, data format, scale X/Y/Z, …
- Picked from the *Metadata* tab (where they can be edited): access, extents, links, contacts, history…

[1] Also available in the *Layer styling panel*

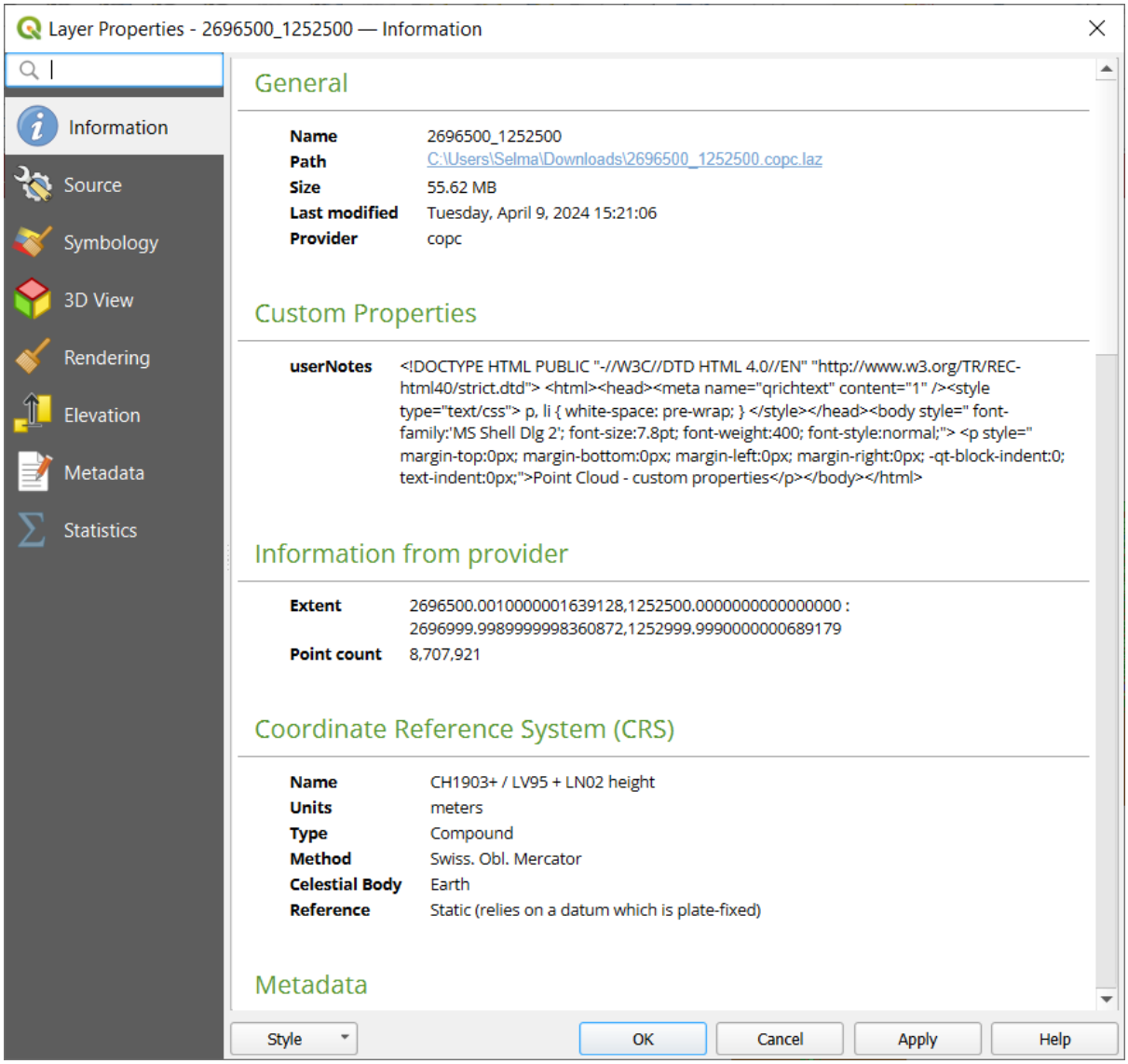


Figura16.1: Point cloud information tab

16.2.2 Propriedades de origem

In the *Source* tab you can see and edit basic information about the point cloud layer:

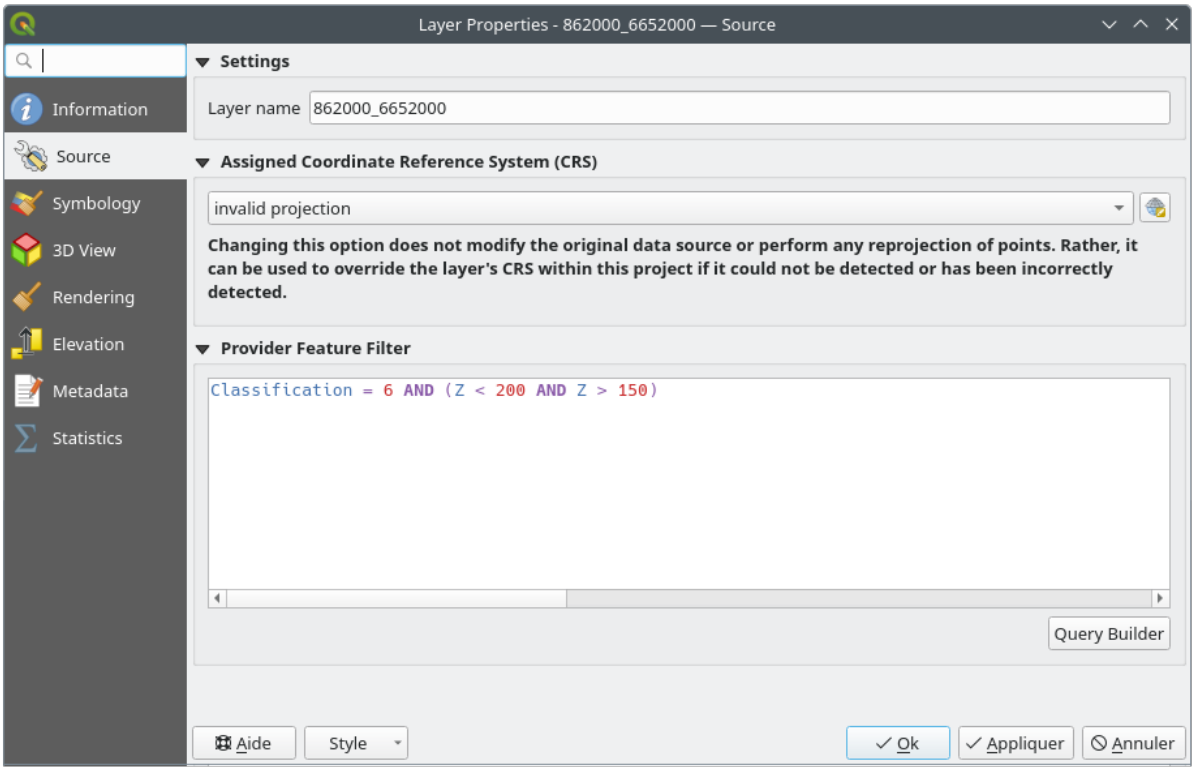


Figura16.2: Point cloud source tab

- *Settings*: Set a Layer name different from the layer filename that will be used to identify the layer in the project (in the Layers Panel, with expressions, in print layout legend,…)
- *Assigned Coordinate Reference System (CRS)*: Here you can change the layer's assigned *Coordinate Reference System*, selecting a recently used one in the drop-down list or clicking on set Projection Select CRS button (see *Seletor do Sistemas de Referência de Coordenadas*). Use this process only if the CRS applied to the layer is a wrong one or if none was applied.
- *Provider Feature Filter*: allows to restrict data to use in the current project to a subset, using functions and expression at the PDAL data provider level. Press the *Query Builder* button at the bottom to start setting the filter.

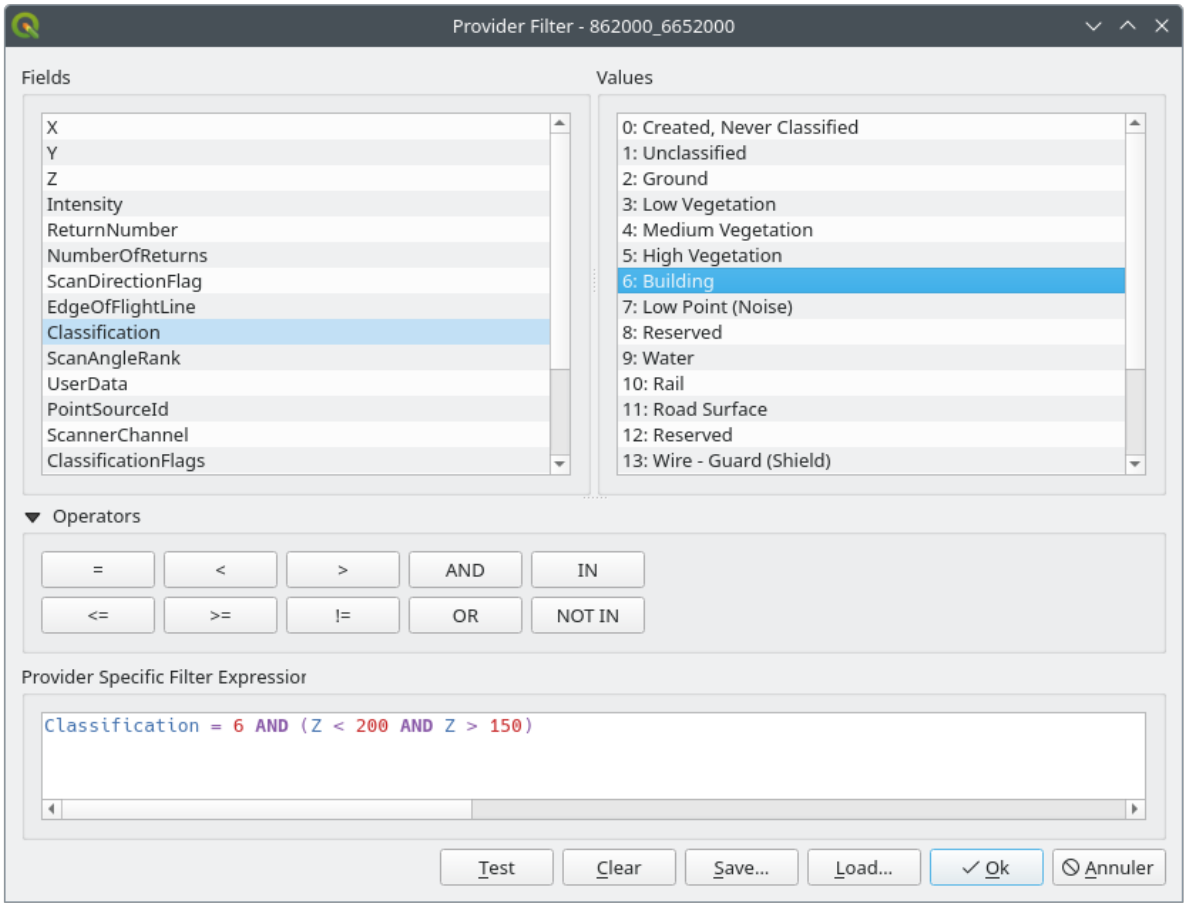


Figura16.3: Filtering point cloud layer to buildings at specific altitude

In the bottom part of the dialog, you can type a *Provider specific filter expression*. Such an expression can be built thanks to:

- *Fields*: the list contains all the attributes of the point cloud layer. To add an attribute to the expression, double-click its name or just type it into the text box.
- *Values*: frame lists values or statistics of the currently selected attribute, as a `key:value` pair. To add a value to the expression field, double click its name in the list: depending on the type of attribute, either the key or the value will be added to the expression. Or type the value in the expression text box.
- *Operators*: this toolbar contains all usable operators. To add an operator to the expression field, click the appropriate button. Relational operators (= , > , ...) and logical operators (`AND`, `OR`, ...) are available.

The *Test* button helps you check the syntax of your query. Use the *Clear* button to wipe the query and revert the layer to its original state (i.e., fully load all the points in the layer). It is possible to *Save...* the query as a `.QQF` file, or *Load...* the query from a file into the dialog.

When a filter is applied, QGIS treats the resulting subset as if it were the entire layer. For example if you applied the *filter above* for filtering buildings, you can not e.g. display, query, save or edit points that are of vegetation classification because they are not part of the subset.

Dica

** As camadas filtradas são indicadas no painel Camadas **

In the *Layers* panel, a filtered layer is listed with a Filter icon next to it indicating the query used when the mouse hovers over the icon. Double-click the icon to open the *Query Builder* dialog for edit. This can

also be achieved through the *Layer* ► *Filter…* menu.

16.2.3 Propriedades da simbologia

In the *Symbology* tab the settings for the rendering of the point cloud are made. In the upper part, the settings of the different feature renderers can be found. In the lower part, there are sections with which general settings for the entire layer can be made and which apply over feature renderers.

Feature Rendering types

There are different options for rendering point clouds that can be selected using the drop-down menu at the top of the *Symbology* tab (see Fig.16.4):

- *Extent Only*: Only a bounding box of the extent of the data is displayed; convenient for overviewing the data extent. As usual, the *Symbol widget* helps you configure any properties (color, stroke, opacity, sub-layers, …) you'd like for the box.
- *Attribute by Ramp*: The data is drawn over a color gradient. See *Attribute by Ramp Renderer*
- *RGB*: Draw the data using red, green and blue color values. See *RGB Renderer*
- *Classification*: The data is drawn using different colors for different classes. See *Classification Renderer*

When a point cloud is loaded, QGIS follows a logic to select the best renderer:

- if the dataset contains color information (red, green, blue attributes), the RGB renderer will be used
- else if the dataset contains a `Classification` attribute, the classified renderer will be used
- else it will fall back to rendering based on Z attribute

If you do not know the attributes of the point cloud, the *Statistics tab* provides a good overview of which attributes are contained in the point cloud and in which ranges the values are located.

For each renderer, you can improve the data display adjusting the *point symbol size* or enabling *surface triangulation*.

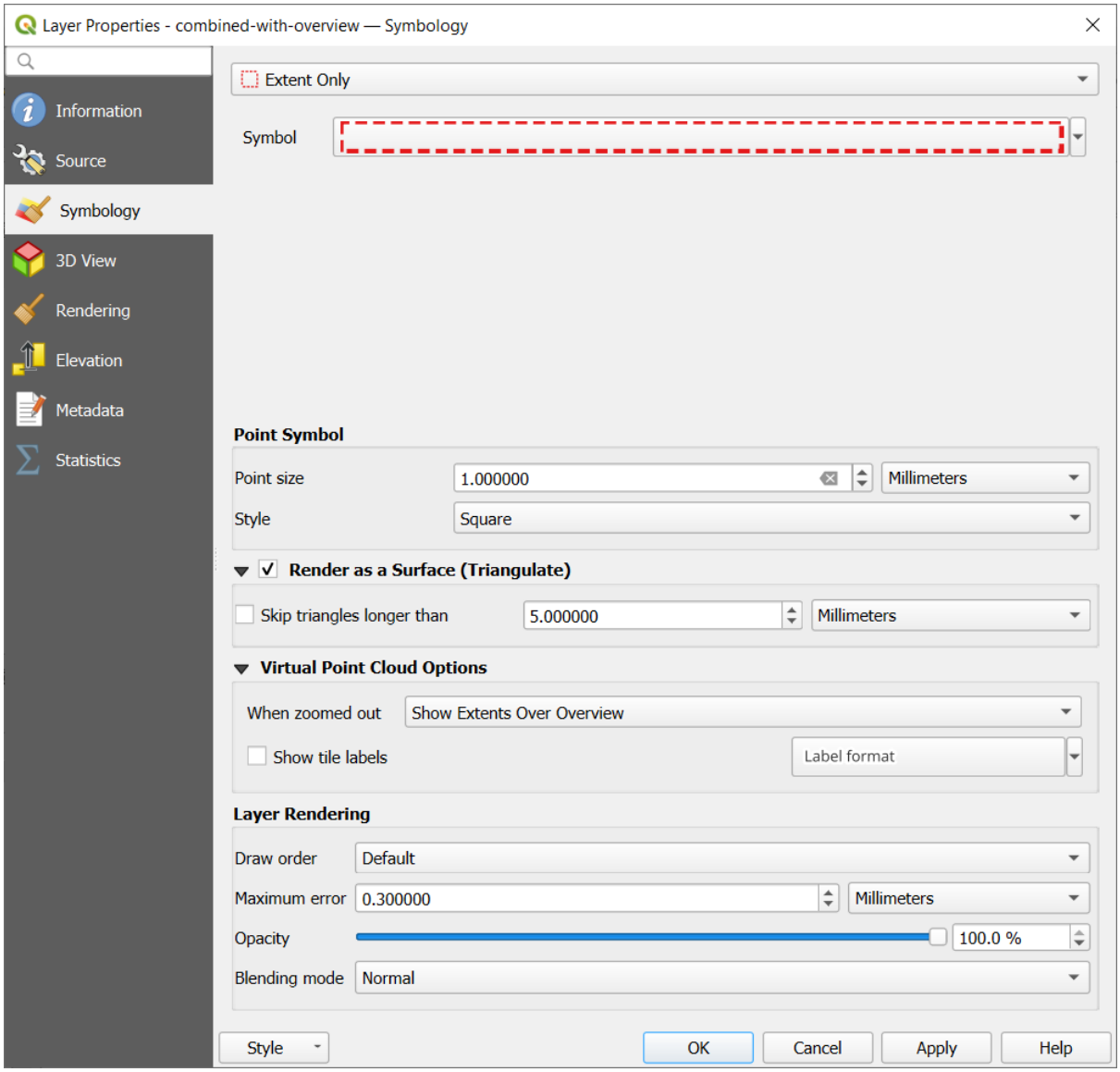


Figura16.4: Point cloud symbology tab

Attribute by Ramp Renderer

With *Attribute by Ramp*, the data can be displayed by numerical values over a color gradient. Such numerical values can be, for example, an existing intensity attribute or the Z-value. Depending on a minimum and a maximum value, the other values are spread to the color gradient via interpolation. The distinct values and their assignment to a certain color are called “color map” and are shown in the table. There are various setting options, which are described below the figure.

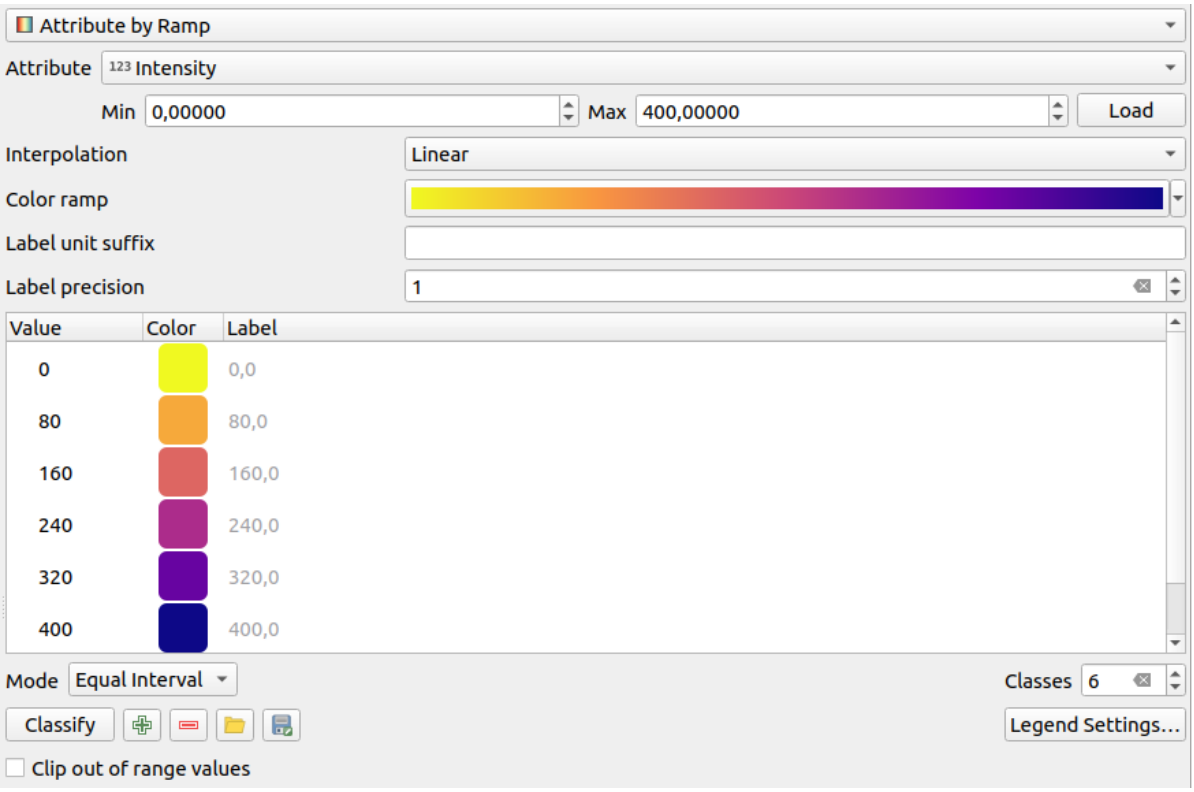


Figura16.5: Point cloud symbology tab: Attribute by Ramp

- *Min* and *Max* define the range that is applied to the color ramp: the *Min* value represents the left, the *Max* value the right end of the color ramp, the values in between are interpolated. By default QGIS detects the minimum and the maximum from the selected attribute but they can be modified. Once you have changed the values, you can restore the defaults by clicking on the *Load* button.
- The *Interpolation* entry defines how values are assigned their color:
  - *Discrete* (a <= symbol appears in the header of the *Value* column): The color is taken from the closest color map entry with equal or higher value
  - *Linear* The color is linearly interpolated from the color map entries above and below the pixel value, meaning that to each dataset value corresponds a unique color
  - *Exact* (a = symbol appears in the header of the *Value* column): Only pixels with value equal to a color map entry are applied a color; others are not rendered.
- The *Color ramp* widget helps you select the color ramp to assign to the dataset. As usual with *this widget*, you can create a new one and edit or save the currently selected one.
- The *Label unit suffix* adds a label after the value in the legend, and the *Label precision* controls the number of decimals to display.

The classification *Mode* helps you define how values are distributed across the classes:

- *Continuous*: Classes number and color are fetched from the color ramp stops; limits values are set following stops distribution in the color ramp (you can find more information on stops in *Setting a Color Ramp*).
- *Equal interval*: The number of classes is set by the *Classes* field at the end of the line; limits values are defined so that the classes all have the same magnitude.

The classes are determined automatically and shown in the color map table. But you can also edit these classes manually:

- Double clicking in a *Value* in the table lets you modify the class value

- Double clicking in the *Color* column opens the *Seletor de Cor* widget, where you can select a color to apply for that value
- Double clicking in the *Label* column to modify the label of the class
- Right-clicking over selected rows in the color table shows a contextual menu to *Change Color…* and *Change Opacity…* for the selection

Below the table there are the options to restore the default classes with *Classify* or to manually Add values or Delete selected values from the table.

Since a customized color map can be very complex, there is also the option to Load an existing color map or to Save it for use in other layers (as a `txt` file).

If you have selected *Linear* for *Interpolation*, you can also configure:

- *Clip out of range values* By default, the linear method assigns the first class (respectively the last class) color to values in the dataset that are lower than the set *Min* (respectively greater than the set *Max*) value. Check this setting if you do not want to render those values.
- *Legend settings*, for display in the *Layers* panel and in the *layout legend*. Customization works the same way as with a raster layer (find more details at *Customize raster legend*).

RGB Renderer

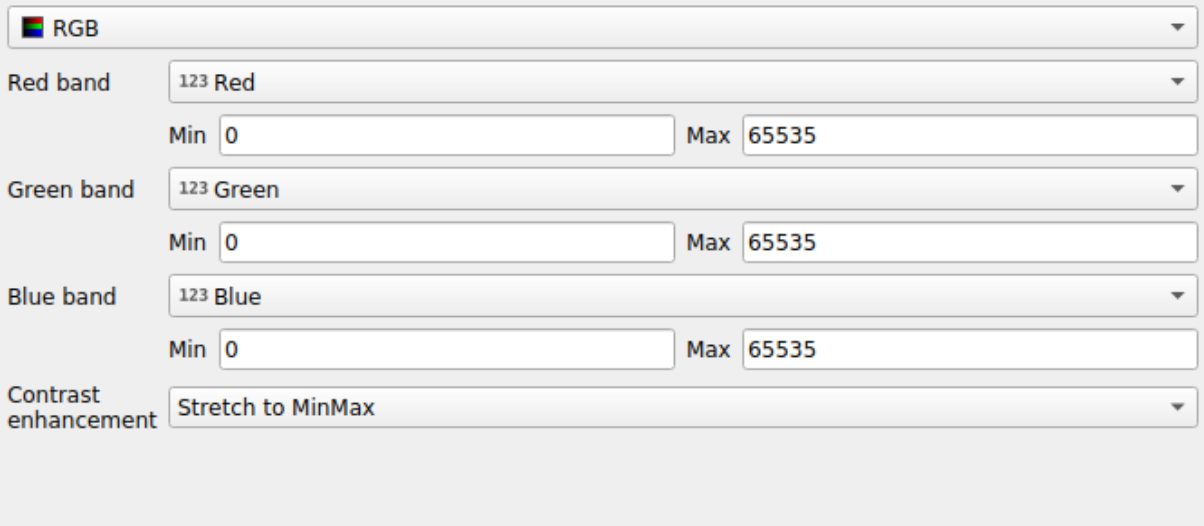


Figura16.6: The point cloud RGB renderer

With the *RGB* renderer, three selected attributes from the point cloud will be used as the red, green and blue component. If the attributes are named accordingly, QGIS selects them automatically and fetches *Min* and *Max* values for each band and scales the coloring accordingly. However, it is also possible to modify the values manually.

A *Contrast enhancement* method can be applied to the values: *No Enhancement*, *Stretch to MinMax*, *Stretch and Clip to MinMax* and *Clip to MinMax*

Nota

The *Contrast enhancement* tool is still under development. If you have problems with it, you should use the default setting *Stretch to MinMax*.

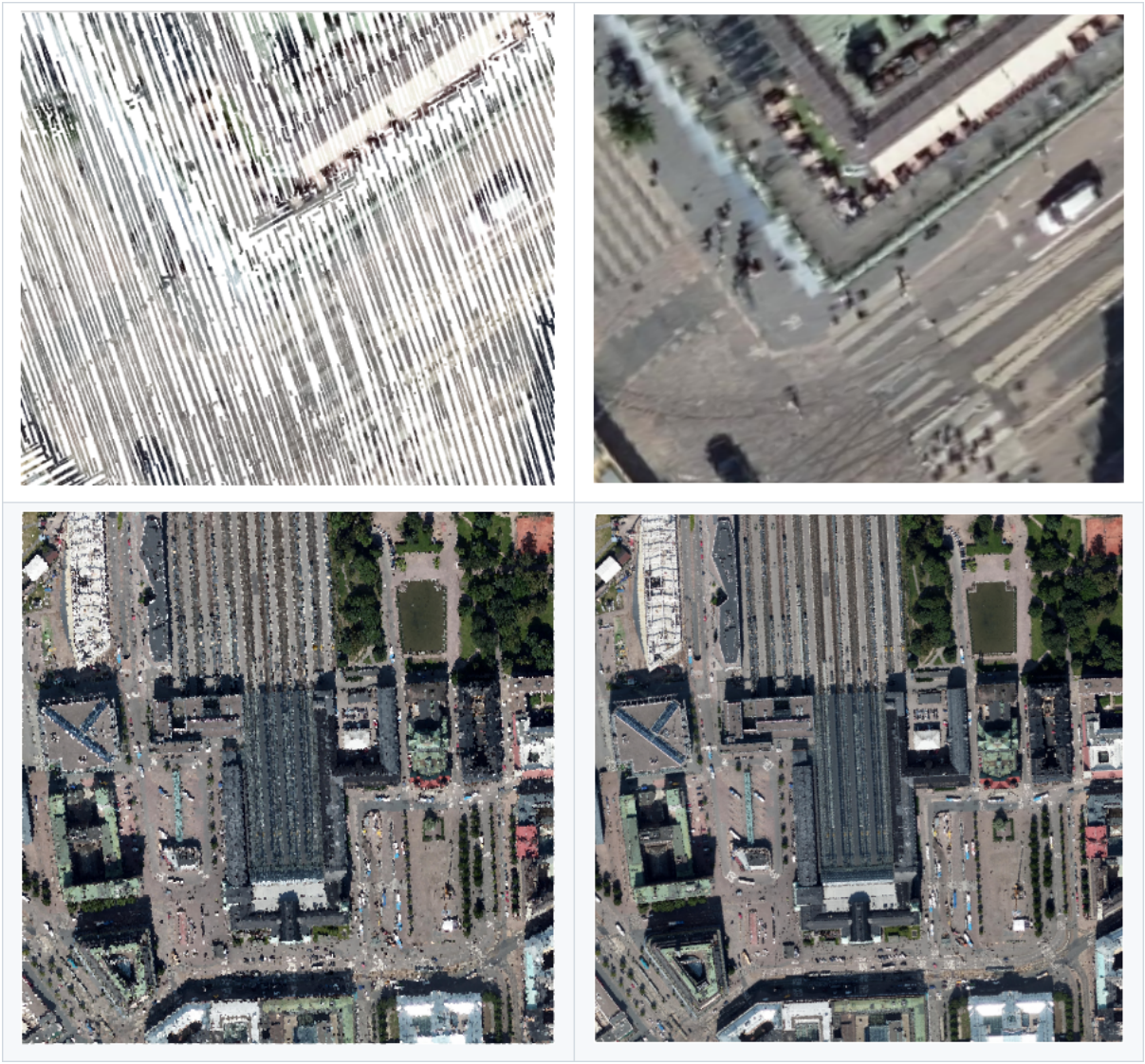


Figura16.7: Example of RGB renderer (left) combined with surface triangulation option (right)

Classification Renderer

In the *Classification* rendering, the point cloud is shown differentiated by color on the basis of an attribute. Any type of attribute can be used (numeric, string, …). Point cloud data often includes a field called `Classification`. This usually contains data determined automatically by post-processing, e.g. about vegetation. With *Attribute* you can select the field from the attribute table that will be used for the classification. By default, QGIS uses the definitions of the LAS specification (see table 'ASPRS Standard Point Classes' in the PDF on ASPRS home page). However, the data may deviate from this schema; in case of doubt, you have to ask the person or institution from which you received the data for the definitions.

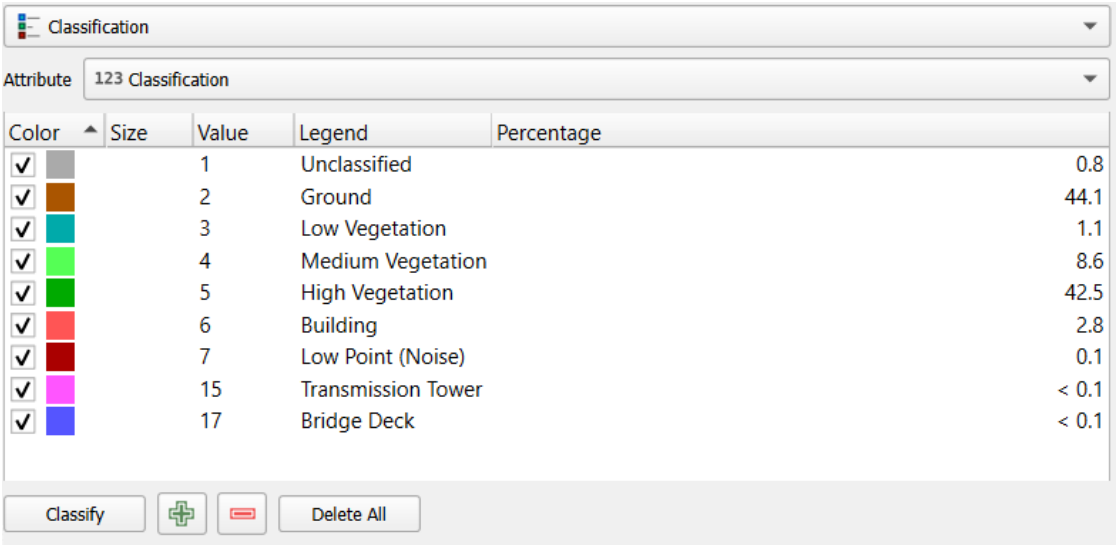


Figura16.8: The point cloud classification renderer

In the table all used values are displayed with the corresponding color and legend. At the beginning of each row there is a check box; if it is unchecked, this value is no longer shown on the map. With double click in the table, you can modify:

- the *Color*: the *Seletor de Cor* widget opens
- the *Size*: assigning a size of `0` to a category will revert it to use the default *point size* set for the layer
- the *Value*
- the *Legend*
- the *Percentage* column will show you the representation of the category within the layer.

You can also right-click one or more rows to open a context menu with the options to change the color, opacity or point size.

Below the table there are buttons with which you can change the default classes generated by QGIS:

- With the *Classify* button the data can be classified automatically: all values that occur in the attributes and are not yet present in the table are added
- With Add and Delete, values can be added or removed manually
- *Delete All* removes all values from the table

Dica

In the *Layers* panel, you can right-click over a class leaf entry of a layer to quickly configure visibility of the corresponding features.

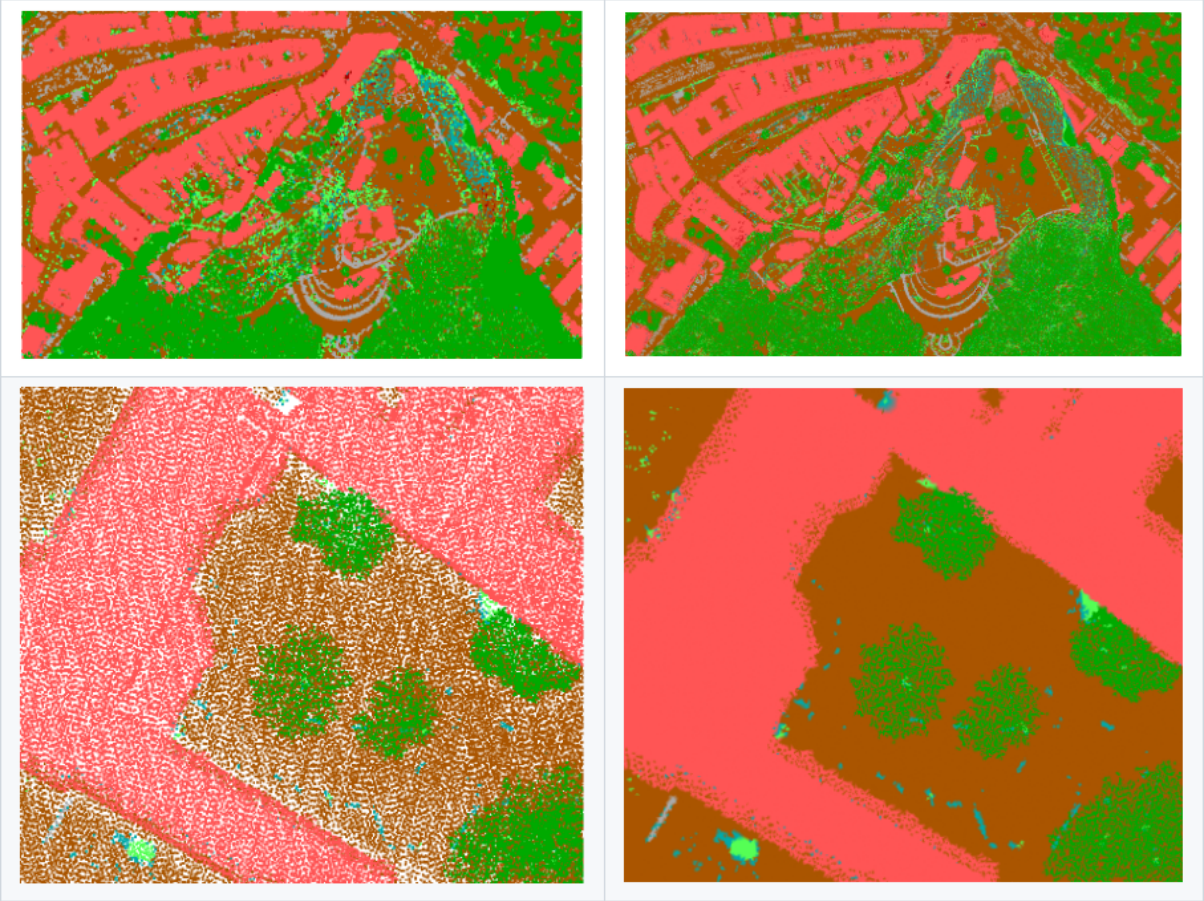


Figura16.9: Example of classification renderer (left) combined with surface triangulation option (right)

Point Symbol

Under *Point Symbol*, the size and the unit (e.g. millimeters, pixels, inches) with which each data point is displayed can be set. Either *Circle* or *Square* can be selected as the style for the points.

Render as a surface (Triangulate)

Check *Render as surface (Triangulate)* to enable the triangulation of the point cloud layer in the 2D view. This option allows rendering triangles instead of points. Each point keeps its color for interpolation in the triangle. You can control the horizontal length of computed triangles:

By checking the *Skip triangles longer than* option and setting up the threshold value, you can control the maximum lenght of a side of the triangles to consider in the horizontal plan. This can be particularly useful if you want to identify actual holes in the data.

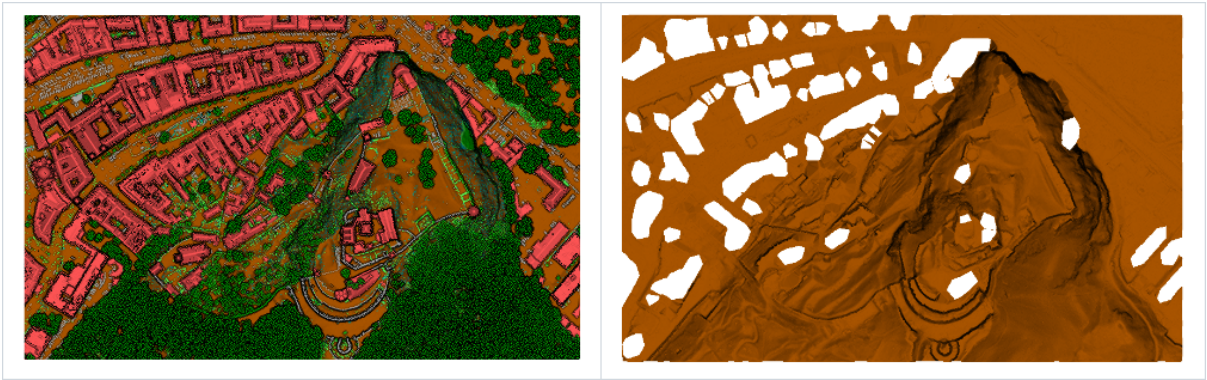


Figura16.10: Rendering data as a surface with map shading (left) and with map shading, filtering large triangles (right)

Virtual Point Cloud Options

The *Virtual Point Cloud Options* are available only when the layer is a *virtual point cloud (VPC)*. QGIS renders the overview of the VPC when zoomed out, if an overview is present. This provides a seamless transition from the overview to the detailed display of individual point clouds as you zoom in.

You can control how the VPC is displayed when zoomed out using the available options:

- *Show Extents Only*: Only the extents of the underlying point clouds are displayed.
- *Show Overview Only*: Only the overview is displayed (if available).
- *Show Extents Over Overview*: The extents are displayed on top of the overview.

You can also choose to *Show tile labels* to display the tile names and set the label format.

Layer Rendering

In the *Layer Rendering* section you have the following options to modify the rendering of the layer:

- *Draw order*: allows to control whether point clouds rendering order on 2d map canvas should rely on their Z value. It is possible to render :
  - with the *Default* order in which the points are stored in the layer,
  - from *Bottom to top* (points with larger Z values cover lower points giving the looks of a true ortho photo),
  - or from *Top to bottom* where the scene appears as viewed from below.
- *Maximum error*: Point clouds usually contains more points than are needed for the display. By this option you set how dense or sparse the display of the point cloud will be (this can also be understood as 'maximum allowed gap between points'). If you set a large number (e.g. 5 mm), there will be visible gaps between points. Low value (e.g. 0.1 mm) could force rendering of unnecessary amount of points, making rendering slower (different units can be selected).
- *Opacity*: You can make the underlying layer in the map canvas visible with this tool. Use the slider to adapt the visibility of your layer to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Blending mode*: You can achieve special rendering effects with this tool. The pixels of your overlaying and underlying layers are mixed through the settings described in *Modos de Mistura*.
- *Eye dome lighting*: this applies shading effects to the map canvas for a better depth rendering. Rendering quality depends on the *draw order* property; the *Default* draw order may give sub-optimal results. Following parameters can be controlled:
  - *Strength*: increases the contrast, allowing for better depth perception

  - *Distance*: represents the distance of the used pixels off the center pixel and has the effect of making edges thicker.

16.2.4 Propriedades da vista 3D

In the *3D View* tab you can make the settings for the rendering of the point cloud in 3D maps.

3D Rendering modes

Following options can be selected from the drop down menu at the top of the tab:

- *No Rendering*: Data are not displayed
- *Follow 2D Symbology*: Syncs features rendering in 3D with *symbology assigned in 2D*
- *Single Color*: All points are displayed in the same *color* regardless of attributes
- *Attribute by Ramp*: Interpolates a given attribute over a color ramp and assigns to features their matching color. See *Attribute by Ramp Renderer*.
- *RGB*: Use different attributes of the features to set the Red, Green and Blue color components to assign to them. See *RGB Renderer*.
- *Classification*: differentiates points by color on the basis of an attribute. See *Classification Renderer*.

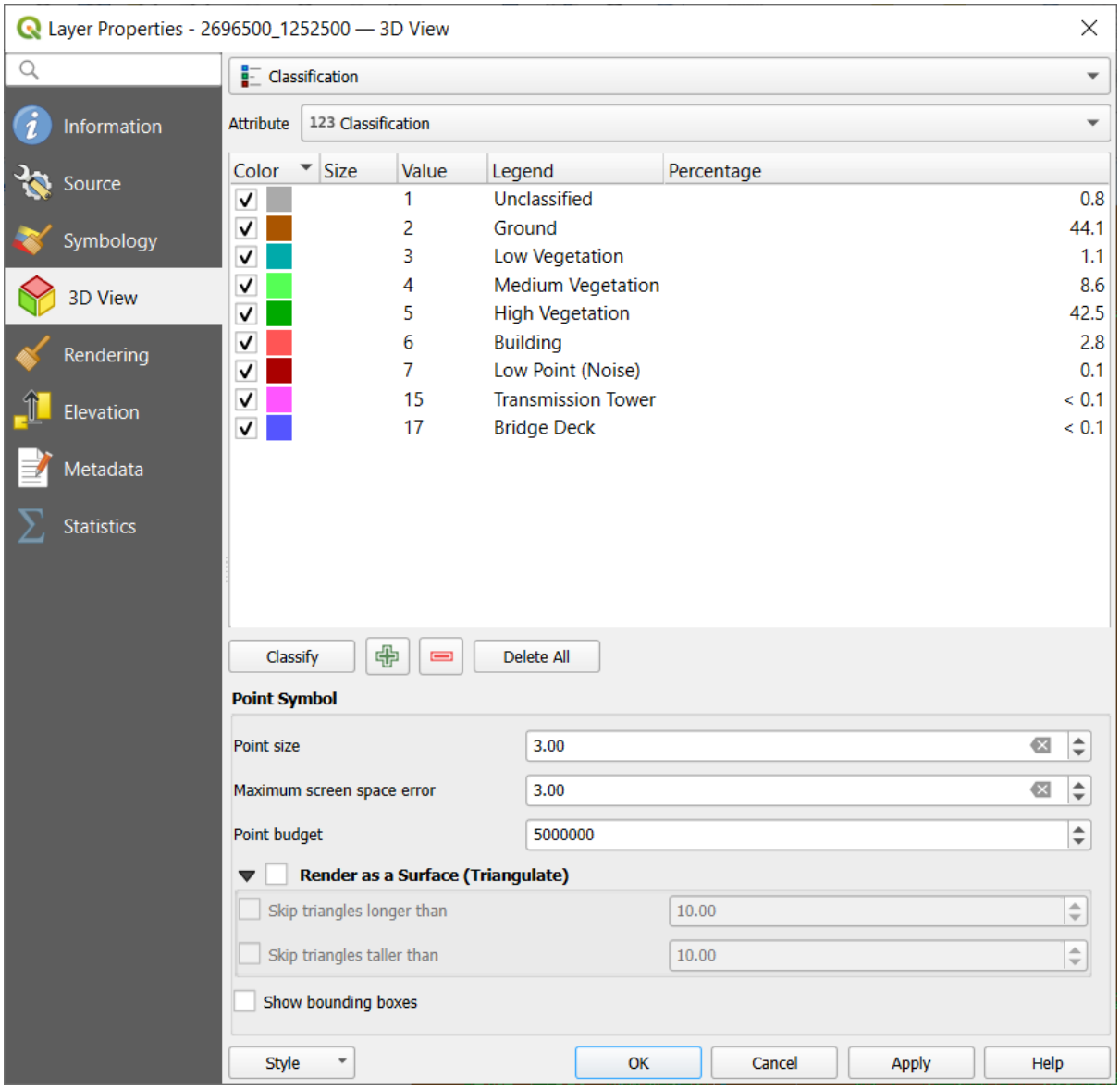


Figura16.11: The point cloud 3D view tab with the classification renderer

3D Point Symbol

In the lower part of the *3D View* tab you can find the *Point Symbol* section. Here you can make general settings for the entire layer which are the same for all renderers. There are the following options:

- *Point size*: The size (in pixels) with which each data point is displayed can be set
- *Maximum screen space error*: By this option you set how dense or sparse the display of the point cloud will be (in pixels). If you set a large number (e.g. 10), there will be visible gaps between points; low value (e.g. 0) could force rendering of unnecessary amount of points, making rendering slower (you can find more details at *Symbology Maximum error*).
- *Point budget*: To avoid long rendering, you can set the maximum number of points that will be rendered
- Check *Render as surface (Triangulate)* to render the point cloud layer in the 3D view with a solid surface obtained by triangulation. You can control dimensions of the computed triangles:
  - *Skip triangles longer than* a threshold value: sets in the horizontal plan, the maximum length of a side

of the triangles to consider

  - *Skip triangles taller than* a threshold value: sets in the vertical plan, the maximum height of a side of the triangles to consider

- *Show bounding boxes*: Especially useful for debugging, shows bounding boxes of nodes in hierarchy

16.2.5 Propriedades de Renderização

Under the *Scale dependent visibility* group box, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scale, defining a range of scale in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Visibility Scale Selector* for more information.

Nota

You can also activate scale dependent visibility on a layer from within the *Layers* panel: right-click on the layer and in the contextual menu, select *Set Layer Scale Visibility*.

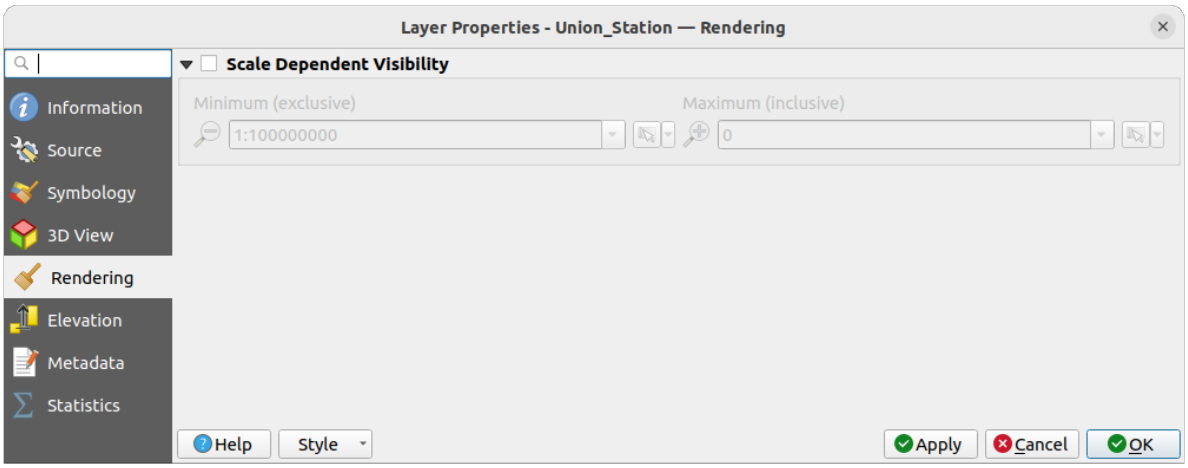


Figura16.12: The point cloud rendering tab

16.2.6 Elevation Properties

In the *Elevation* tab, you can set corrections for the Z-values of the data. This may be necessary to adjust the elevation of the data in 3D maps and its appearance in the *profile tool charts*. There are following options:

- *Vertical Reference System*: If the CRS of your point cloud layer is a compound one (including a Z dimension), then the vertical CRS used for the layer will be automatically derived from the vertical component of the layer's CRS. In this case, you cannot manually set a different vertical CRS, and the option to change it will be disabled. If your point cloud layer uses a horizontal (2D) CRS (though uncommon), you can manually select a specific vertical CRS by clicking on the Select CRS. Vertical reference systems are supported for point cloud layers in:
  - *Elevation profiles*
  - *Identify Tool Results*
  - *3D map views*
- Under *Elevation* group:

  - You can set a *Scale*: If `10` is entered here, a point that has a value Z = `5` is displayed at a height of `50`.
  - An *offset* to the z-level can be entered. This is useful to match different data sources in its height to each other. By default, the lowest z-value contained in the data is used as this value. This value can also be restored with the Refresh button at the end of the line.

- Under *Profile Chart Accuracy*, the *Maximum error* helps you control how dense or sparse the points will be rendered in the elevation profile. Larger values result in a faster generation with less points included.
- Under *Profile Chart Appearance*, you can control the point display:
  - *Point size*: the size to render the points with, in supported units (millimeters, map units, pixels, …)
  - *Style*: whether to render the points as *Circle* or *Square*
  - Apply a single *Color* to all the points visible in the profile view
  - Check *Respect layer's coloring* to instead show the points with the color assigned via their *2D symbology*
  - *Apply opacity by distance from curve effect*, reducing the opacity of points which are further from the profile curve

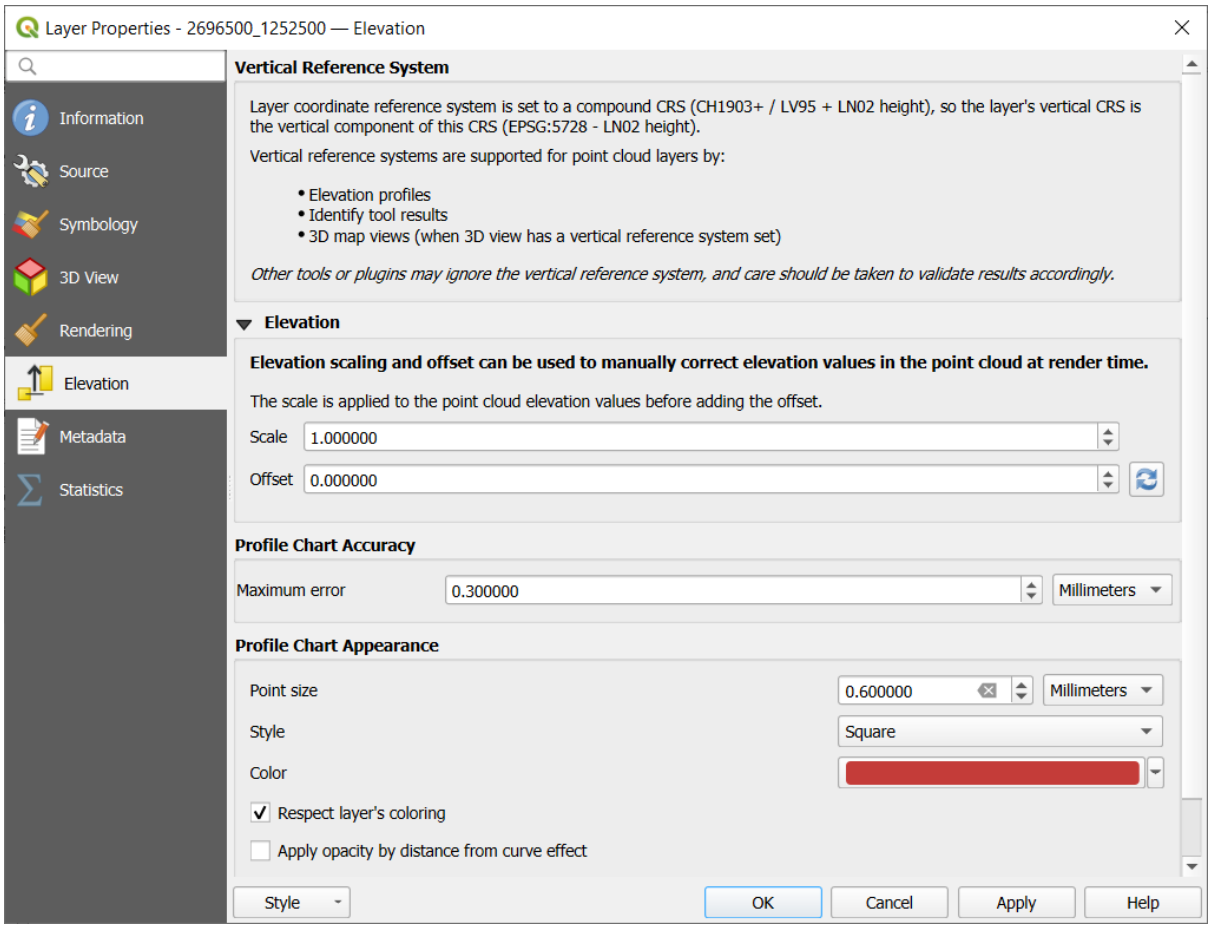


Figura16.13: The point cloud elevation tab

16.2.7 Propriedades dos Metadados

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metadados* for more information.

16.2.8 Statistics Properties

In the *Statistics* tab you can get an overview of the attributes of your point cloud and their distribution.

At the top you will find the section *Attribute Statistics*. Here all attributes contained in the point cloud are listed, as well as some of their statistical values: *Minimum*, *Maximum*, *Mean*, *Standard Deviation*

If there is an attribute *Classification*, then there is another table in the lower section. Here all values contained in the attribute are listed, as well as their absolute *Count* and relative *%* abundance.

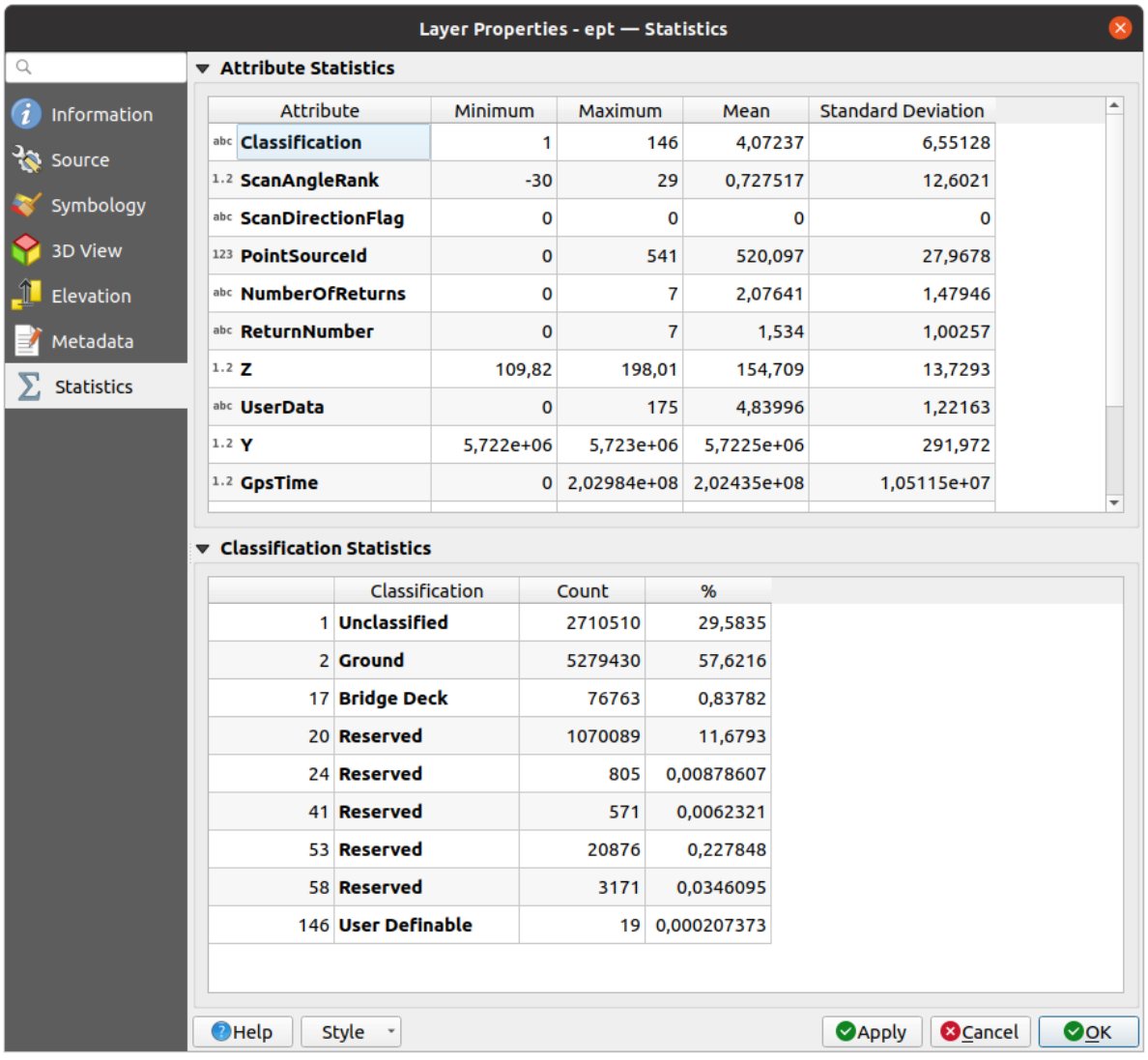


Figura16.14: The point cloud statistics tab

16.3 Virtual point cloud

Lidar surveys of larger areas are often multi-terabyte datasets with many billions of points. Having such large datasets represented as a single point cloud file is not practical due to the difficulties of storage, transfer, display and analysis. Point cloud data are therefore typically stored and distributed split into square tiles (e.g. `1km x 1km`), each tile having a more manageable file size (e.g. ~200 MB when compressed).

Tiling of data solves the problems with size of data, but it introduces issues when processing or viewing an area of interest that does not fit entirely into a single tile. Users need to develop workflows that take into account multiple tiles and special care needs to be taken to deal with data near edges of tiles to avoid unwanted artefacts in outputs. Similarly, when viewing point cloud data, it becomes cumbersome to load many individual files and apply the same symbology.

Here is an example of several point cloud tiles loaded in QGIS. Each tile is styled based on min/max Z values of the tile, creating visible artefacts on tile edges. The styling has to be adjusted for each layer separately:

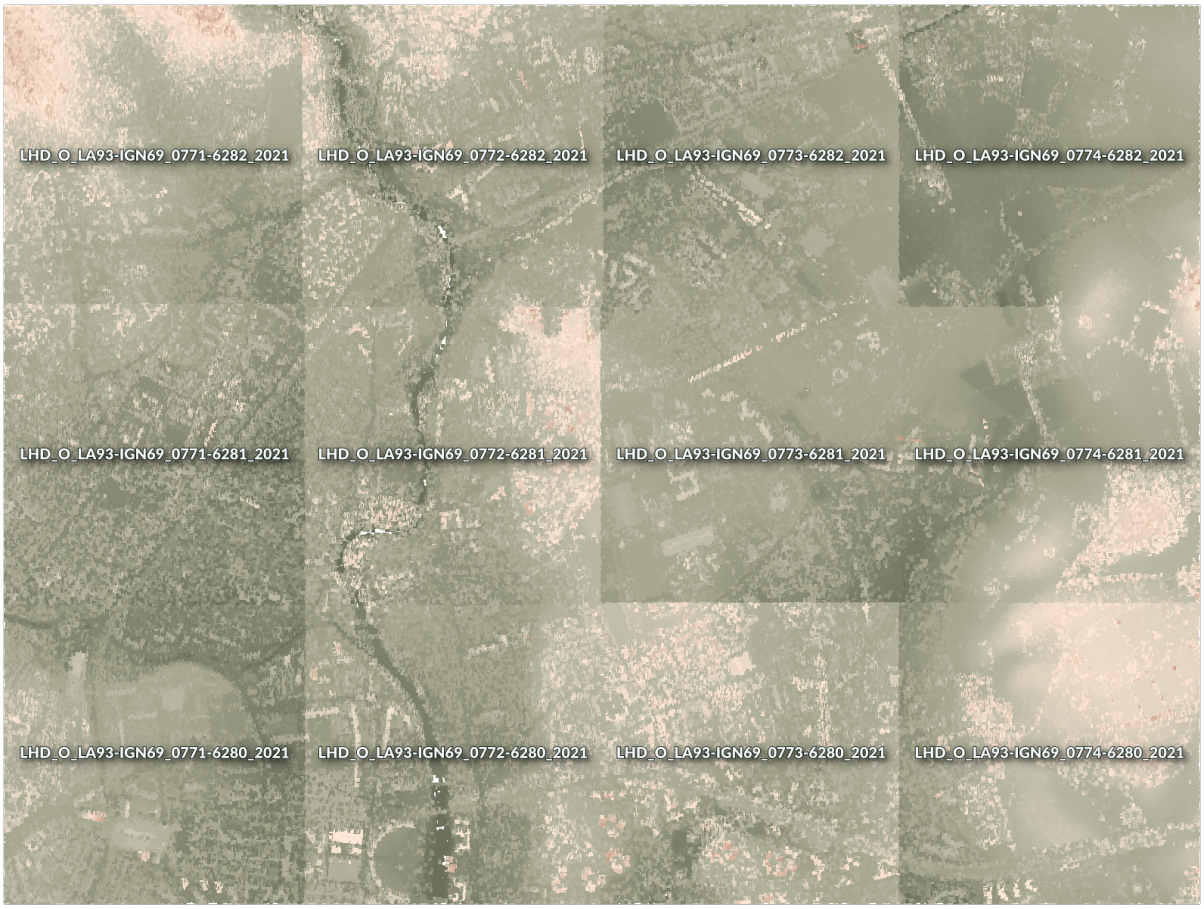


Figura16.15: Individual point cloud tiles loaded, with artefacts on edges

In the GIS world, many users are familiar with the concept of virtual rasters. A virtual raster is a file that simply references other raster files with actual data. In this way, GIS software then treats the whole dataset comprising many files as a single raster layer, making the display and analysis of all the rasters listed in the virtual file much easier.

Borrowing the concept of virtual rasters from GDAL, **virtual point cloud (VPC)** is a file format that references other point cloud files. Software supporting virtual point clouds handles the whole tiled dataset as a single data source.

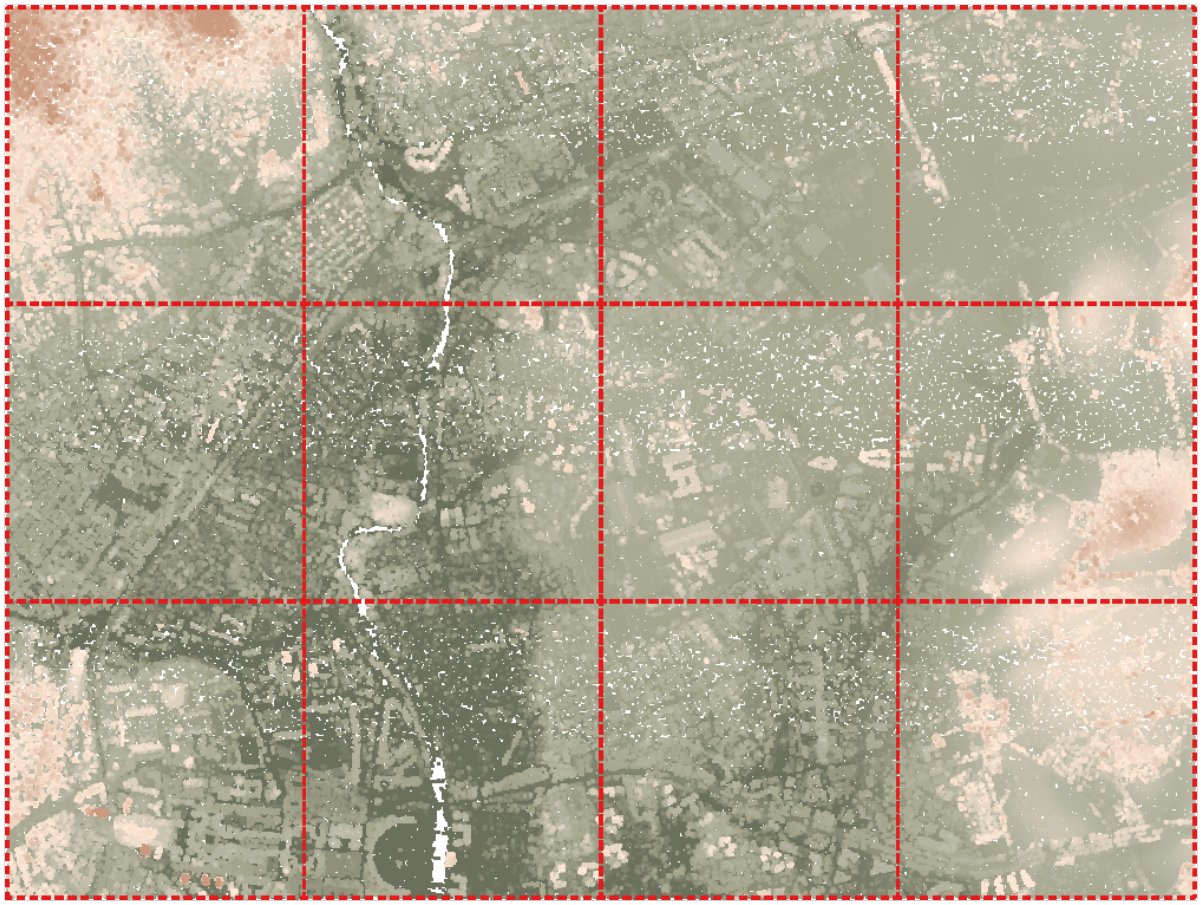


Figura16.16: The virtual point cloud

Displaying and manipulating virtual point cloud is much more fluent and easy.

At the core, a virtual point cloud file is a simple JSON file with `.vpc` extension, containing references to actual data files (e.g. `.LAS`, `.LAZ` or `.COPC` files) and additional metadata extracted from the files. Even though it is possible to write VPC files by hand, it is strongly recommended to create them using an automated tool, such as:

- The Processing *Build virtual point cloud (VPC)* algorithm
- The `build_vpc` command of PDAL wrench tool

For more details, please refer to the VPC specification that also contains best practices and optional extensions (such as overviews).

CAPÍTULO 17

Working with 3D Tiles

17.1 What are 3D Tiles?

3D tiles are specification for streaming and rendering large-scale 3D geospatial datasets. They use a hierarchical structure to efficiently manage and display 3D content, optimising performance by dynamically loading appropriate levels of detail. This technology is widely used in urban planning, architecture, simulation, gaming, and virtual reality, providing a standardised and interoperable solution for visualising complex geographical data.

Figura17.1: Example of 3D tiles

See *Using 3D tiled scene services* for instructions on how to add these data sources to QGIS.

17.1.1 State of 3D Tiles Support

Currently, QGIS supports two formats of 3D tiles:

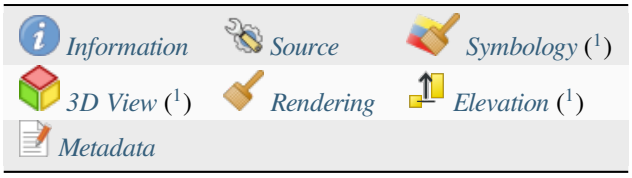
- "Cesium" 3D Tiles are used primarily for complex 3D models of buildings or whole cities. Such datasets can be provided by cloud-based platforms such as Cesium Ion or Google (*Photorealistic 3D Tiles*).
- Quantized Mesh tiles are used for terrain elevation data.

Supported *3D Tiles* features:

- QGIS currently has partial support for 3D Tiles 1.0 and partial support for 3D Tiles 1.1.
- For 3D Tiles 1.0, the tile format Batched 3D Model (`b3dm`) is supported. The tile formats Instanced 3D Model (`i3dm`), Point Cloud (`pnts`) and Composite (`cmpt`) are not supported yet.
- For 3D Tiles 1.1, the `3DTILES_content_gltf` extension is supported, so tile contents may also be encoded as glTF 2.0 (text or binary). The `EXT_mesh_gpu_instancing` extension is not supported yet.
- Currently only explicit tiling is supported. Implicit tiling is not supported yet
- Currently only tilesets using EPSG:4978 are supported.

17.2 3D Tiles Properties

The 3D tiles *Layer Properties* dialog provides the following sections:



17.2.1 Propriedades da informação

The *Information* tab is read-only and represents an interesting place to quickly grab summarized information and metadata on the current layer. Provided information are:

- based on the provider of the layer: name, URL, source type and path, number of zoom levels
- custom properties, used to store in the active project additional information about the layer. More properties can be created and managed using PyQGIS, specifically through the setCustomProperty() method.
- o Sistema de Referência de Coordenadas: nome, unidades, método, precisão, referência (ou seja, se é estático ou dinâmico)
- picked from the *filled metadata*: access, extents, links, contacts, history…

17.2.2 Propriedades de origem

The *Source* tab displays basic information about the selected 3D tile, including:

- o *Nome da Camada* para ser mostrado no *Painel de Camadas*;
- the *Coordinate Reference System*: Displays the layer's *Coordinate Reference System (CRS)*. You can change the layer's CRS, by selecting a recently used one in the drop-down list or clicking on the Select CRS button (see *Seletor do Sistemas de Referência de Coordenadas*). Use this process only if the layer CRS is wrong or not specified.

[1] Also available in the *Layer styling panel*

17.2.3 Propriedades da simbologia

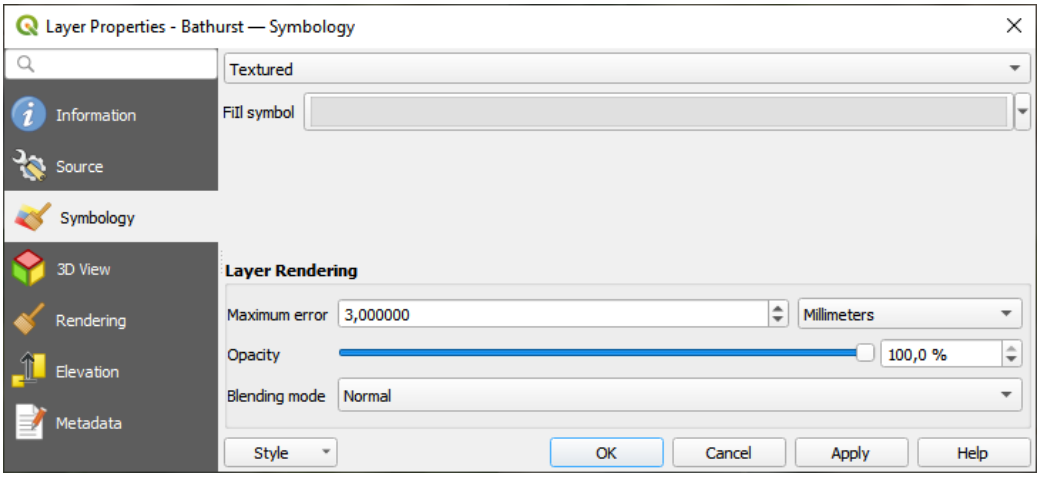


Figura17.2: 3D Tile Layer Symbology

By default, the layer is styled using texture, but you can change it to see the wireframe mesh behind the scene by choosing *Wireframe* in the drop-down menu. You can also, change the mesh fill and line symbols similar to the vector polygons. Checking *Use texture colors* will render each mesh element with the average value of the full texture. This is a good option to try when dealing with a large dataset and want to get a quick overview of the data.

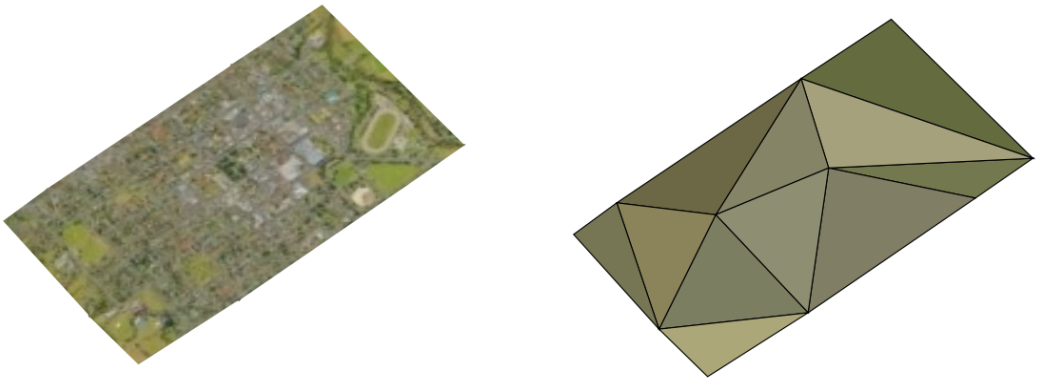


Figura17.3: 3D Tiles - textured and wireframe

From the *Symbology* tab, you can also set some options that invariably act on all features of the layer:

- *Maximum error*: This parameter determines the level of detail displayed in the 3D model. Similar to point clouds, 3D tiles often contain more information than necessary for visual representation. By adjusting this setting, you control the balance between display density and rendering speed. A larger value (e.g., 5 mm) may introduce noticeable gaps between elements, while a smaller value (e.g., 0.1 mm) could lead to the rendering of an excessive number of details, potentially slowing down the rendering process. Different units can be selected to tailor the setting to your specific needs.
- *Opacity*: Adjusts the visibility of the underlying layer on the map canvas using this tool. Use slider to tailor the visibility of your scene layer according to your preferences. Alternatively, specify the exact percentage of

visibility through the menu next to the slider.

- *Blending mode*: You can achieve special rendering effects with these tools that you may previously only know from graphics programs. The pixels of your overlaying and underlaying layers are mixed through the settings described in *Modos de Mistura*.

17.2.4 Propriedades da vista 3D

- *Maximum screen space error*: Determines the threshold for swapping terrain tiles with more detailed ones (and vice versa) - i.e. how soon the 3D view will use higher quality tiles. Lower numbers mean more details in the scene at the expenses of increased rendering complexity.
- *Show bounding boxes*: Shows 3D bounding boxes of the terrain tiles (useful for troubleshooting terrain issues).

To view the data you can open *New 3D map view*.

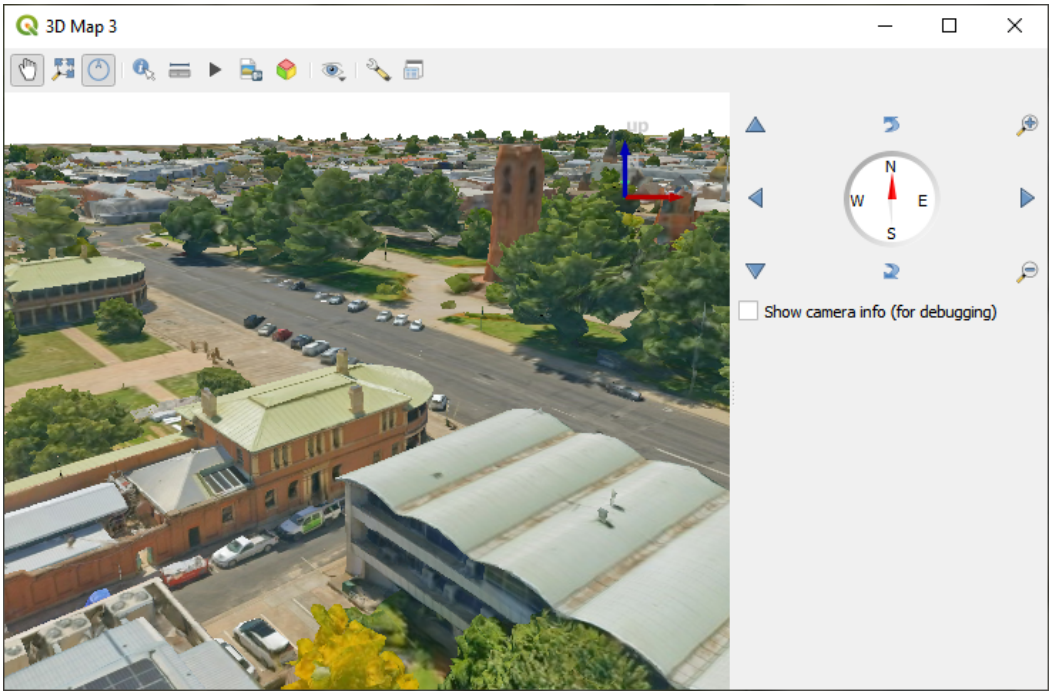


Figura17.4: Visualização de Mapa 3D

17.2.5 Propriedades de Renderização

Under *Scale dependent visibility*, you can set the *Maximum (inclusive)* and *Minimum (exclusive)* scales, defining a range of scales in which features will be visible. Out of this range, they are hidden. The Set to current canvas scale button helps you use the current map canvas scale as boundary of the range visibility. See *Visibility Scale Selector* for more information.

17.2.6 Elevation Properties

The *Elevation* tab provides options to control the layer elevation properties within a *3D map view*. Specifically, you can set:

- *Elevation Surface*: how the 3D layer vertices Z values should be interpreted as terrain elevation. You can apply a *Scale* factor and an *Offset*.

17.2.7 Propriedades dos Metadados

The *Metadata* tab provides you with options to create and edit a metadata report on your layer. See *Metadados* for more information.

CAPÍTULO 18

Laying out the maps

With Print Layouts and Reports you can create maps and atlases, and print them or save them as image, PDF or SVG files.

18.1 Visão Geral do Layout de Impressão

The print layout provides growing layout and printing capabilities. It allows you to add elements such as the QGIS 2D or 3D map canvas, text labels, images, legends, scale bars, basic shapes, arrows, attribute and simple tables, elevation profiles and HTML frames… You can size, group, align, position and rotate each element and adjust their properties to create your layout.

The layout can be printed or exported to image formats, PostScript, PDF or to SVG. You can save the layout as a template and load it again in another session. Finally, generating several maps based on a template can be done through the atlas generator.

18.1.1 Sessão de Amostra para iniciantes

Before you start to work with the print layout, you need to load some raster or vector layers in the QGIS map canvas and adapt their properties to suit your own convenience. After everything is rendered and symbolized to your liking, click the New Print Layout icon in the *Project* toolbar or choose *Project* ► *New Print Layout*. You will be prompted to choose a title for the new layout.

Para demonstrar como criar um mapa, siga as próximas instruções.

1. On the left side, select the Add map toolbar button and draw a rectangle on the canvas holding down the left mouse button. Inside the drawn rectangle the QGIS map view to the canvas.
2. Select the Add scalebar toolbar button and click with the left mouse button on the print layout canvas. A scalebar will be added to the canvas.
3. Select the Add legend toolbar button and draw a rectangle on the canvas holding down the left mouse button. Inside the drawn rectangle the legend will be drawn.
4. Select the Select/Move item icon to select the map on the canvas and move it a bit.

5. Enquanto o item mapa ainda está selecionado, você também pode alterar o tamanho do item mapa. Clique enquanto mantém pressionado o botão esquerdo do mouse em um pequeno retângulo branco em um dos cantos do item mapa e arraste-o para um novo local para alterar seu tamanho.
6. Click the *Item Properties* panel on the left down side and find the setting for the orientation. Change the value of the setting *Map orientation* to '15.00°'. You should see the orientation of the map item change.
7. Now, you can print or export your print layout to image formats, PDF or to SVG with the export tools in *Layout* menu.
8. Finally, you can save your print layout within the project file with the Save Project button.

You can add multiple elements to the print layout. It is also possible to have more than one map view or legend or scale bar in the print layout canvas, on one or several pages. Each element has its own properties and, in the case of the map, its own extent. If you want to remove any elements from the layout canvas, you can do that with the `Delete` or the `Backspace` key.

18.1.2 The Layout Manager

The *Layout Manager* is the main window to manage print layouts in the project. It gives you an overview of existing print layouts and reports in the project and offers tools to:

- procurar por um layout;
- add new print layout or new report from scratch, template or duplicating an existing one;
- renomear ou excluir qualquer um deles;
- abri-los no projeto.

To open the layout manager dialog:

- from the main QGIS dialog, select *Project* ► *Layout Manager…* menu or click on the Layout Manager button in the *Project Toolbar*;
- from a print layout or report dialog, select *Layout* ► *Layout Manager…* menu or click on the Layout Manager button in the *Layout Toolbar*.

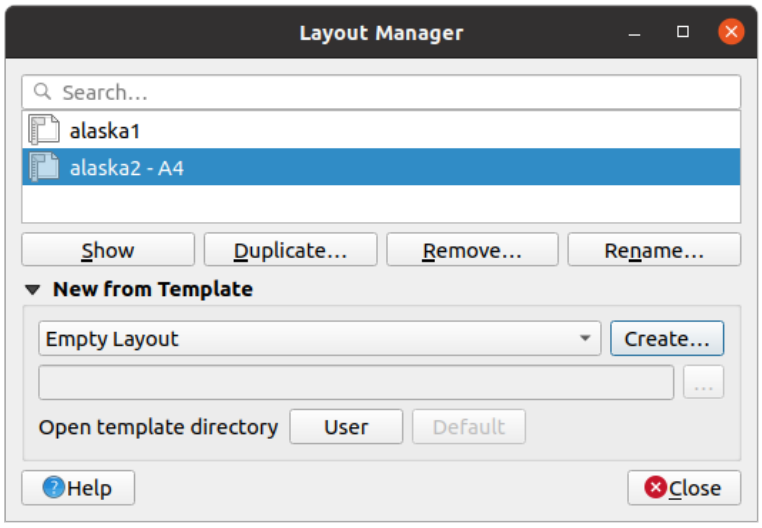


Figura18.1: The Print Layout Manager

The layout manager lists in its upper part all the available print layouts or reports in the project with tools to:

- show the selection: you can select multiple reports and/or print layout(s) and open them in one-click. Double-click a name also opens it;

- duplicate the selected print layout or report (available only if one item is selected): it creates a new dialog using the selected one as template. You'll be prompted to choose a new title for the new layout;
- rename the report or layout (available only if one item is selected): you'll be prompted to choose a new title for the layout;
- remove the layout: the selected print layout(s) will be deleted from the project.

In the lower part, it's possible to create new print layouts or reports from scratch or a template. By default, QGIS will look for templates in the *user profile* folder and the application template directories (accessible with the two buttons at the bottom of the frame) but also in any folder declared as *Path(s) to search for extra print templates* in *Settings* ► *Options* ► *Layouts*. Found templates are listed in the combobox. Select an item and press the *Create* button to generate a new report or print layout.

You can also use layout templates from a custom folder; in that case, select *specific* in the templates drop-down list, browse to the template and press *Create*.

> **Dica**
>
> **Creating template-based print layouts from Browser panel**
>
> Drag-and-drop a print layout template `.qpt` file from any file browser onto the map canvas or double-click it in the *Browser panel* generates a new print layout from the template.

18.1.3 Menus, tools and panels of the print layout

Opening the print layout provides you with a blank canvas that represents the paper surface when using the print option. Initially you find buttons on the left beside the canvas to add print layout items: 2D or 3D map canvases, text labels, images, legends, scale bars, basic shapes, arrows, attribute or simple tables, HTML frames, elevation profiles,… In this toolbar you also find buttons to navigate, zoom in on an area and pan the view on the layout as well as buttons to select any layout item and to move the extents of the map items.

Fig.18.2 shows the initial view of the print layout before any elements are added.

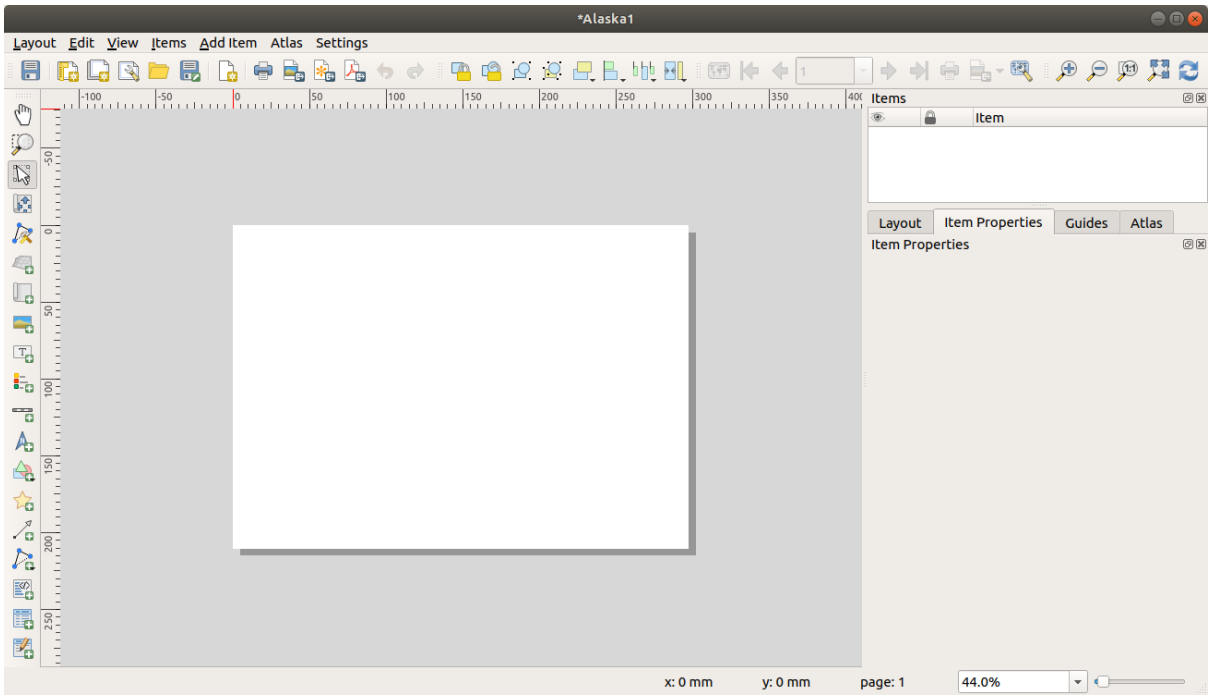


Figura18.2: Layout de impressão

On the right beside the canvas you find two set of panels. The upper one holds the panels *Items* and *Undo History* and the lower holds the panels *Layout*, *Item properties* and *Atlas generation*.

- The *Items* panel provides a list of all the print layout items added to the canvas and ways to globally interact with them (see *The Items Panel* for more information).
- The *Undo History* panel displays the history of all changes applied to the layout. With a mouse click, it is possible to undo and redo layout steps back and forth to a certain status.
- The *Layout* panel allows you to set general parameters to apply to the layout when exporting or working within (see *O Painel de Layout* for more details);
- The *Item Properties* panel displays the properties for the selected item. Click the Select/Move item icon to select an item (e.g., legend, scale bar or label) on the canvas. Then click the *Item Properties* panel and customize the settings for the selected item (see *Itens de layout* for detailed information on each item settings).
- The *Atlas* panel allows you to enable the generation of an atlas for the current layout and gives access to its parameters (see *Gerar um Atlas* for detailed information on atlas generation usage).

In the bottom part of the print layout window, you can find a status bar with mouse position, current page number, a combo box to set the zoom level, the number of selected items if applicable and, in the case of atlas generation, the number of features.

In the upper part of the print layout window, you can find menus and other toolbars. All print layout tools are available in menus and as icons in a toolbar.

The toolbars and the panels can be switched off and on using the right mouse button over any toolbar, or through *View* ► *Toolbars* ► or *View* ► *Panels* ►.

Menus e Ferramentas

Layout menu

The *Layout* provides action to manage the layout:

- Save the project file directly from the print layout window.
- Create a new and blank print layout with *New Layout…*.
- *Duplicate Layout…* : Create a new print layout by duplicating the current one.
- Remove the current layout with *Delete Layout…*.
- Open the *Layout Manager…*.
- *Layouts* ► : Open an existing print layout.

Once the layout is designed, with *Save as Template* and *Add Items from Template* icons, you can save the current state of a print layout session as a `.qpt` template file and load its items again in another session/print layout.

In the *Layout* menu, there are also powerful ways to share geographical information produced with QGIS that can be included in reports or published. These tools are *Export as Image…*, *Export as PDF…*, *Export as SVG…* and *Print…*.

Below is a list of all the available tools in this menu with some convenient information.

Tabela18.1: “Layout” Menu Tools

Ferramenta	Atalho	Barra de Ferramentas	Referencia
Save Project	`Ctrl+S`	*Layout*	*Introduzindo projetos do QGIS*
New Layout	`Ctrl+N`	*Layout*	*The Layout Manager*
Duplicate Layout		*Layout*	*The Layout Manager*
Delete Layout			
Gerenciador de Layout…		*Layout*	*The Layout Manager*
Layouts ►			
Layout Properties…			*O Painel de Layout*
Rename Layout…			
Page Properties…			*Trabalhando com as propriedades da página*
Add Pages…		*Layout*	*Trabalhando com as propriedades da página*
Add Items from Template		*Layout*	*Criando um item esquema*
Save as Template…		*Layout*	*The Layout Manager*
Export as Image…		*Layout*	*Exportar como Imagem*
Export as SVG…		*Layout*	*Exportar como SVG*
Export as PDF…		*Layout*	*Exportar como PDF*
Printer Page Setup…	`Ctrl+Shift+P`		
Print…	`Ctrl+P`	*Layout*	*Criando um arquivo de Saída*
Fechar	`Ctrl+Q`		

Editar menu

The *Edit* menu offers tools to manipulate print layout items. It includes common actions like selection tools, Copy/Cut/Paste and undo/redo (see *The Undo History Panel: Revert and Restore actions*) functionality for the items in the layout.

When using the Paste action, the elements will be pasted according to the current mouse position. Using the *Edit* ► *Paste in Place* action or pressing `Ctrl+Shift+V` will paste the items into the current page, at the same position they were in their initial page. It ensures to copy/paste items at the same place, from page to page.

Below is a list of all the available tools in this menu with some convenient information.

Tabela18.2: “Edit” Menu Tools

Ferramenta	Atalho	Barra de Ferramentas	Referencia
Undo (last change)	`Ctrl+Z`	*Layout*	*The Undo History Panel: Revert and Restore actions*
Redo (last reverted change)	`Ctrl+Y`	*Layout*	*The Undo History Panel: Revert and Restore actions*
Delete	`Del`		
Cortar	`Ctrl+X`		
Copiar	`Ctrl+C`		

continua na próxima página

Tabela 18.2 – continuação da página anterior

Ferramenta	Atalho	Barra de Ferramentas	Referencia
Colar	`Ctrl+V`		
Paste in place	`Ctrl+Shift+V`		
Selecionar tudo	`Ctrl+A`		
Deselect all	`Ctrl+Shift+A`		
Invert Selection			
Select Next Item Below	`Ctrl+Alt+[`		
Select Next Item above	`Ctrl+Alt+]`		
Pan Layout	`P`	*Toolbox*	
Zoom	`Z`	*Toolbox*	
Select/Move Item	`V`	*Toolbox*	*Interagindo com itens de layout*
Move Content	`C`	*Toolbox*	*Ítem do Mapa*
Edit Nodes Item		*Toolbox*	*Os itens de forma baseados em nós*

ver menu

The *View* menu gives access to navigation tools and helps to configure general behavior of the print layout. Beside the common zoom tools, you have means to:

- Refresh view (if you find the view in an inconsistent state);
- enable a *grid* you could snap items to when moving or creating them. Grids setting is done in *Settings* ► *Layout Options…* or in the *Layout Panel*;
- enable *guides* you could snap items to when moving or creating them. Guides are red lines that you can create by clicking in the ruler (above or at the left side of the layout) and drag and drop to the desired location;
- *Smart Guides*: uses other layout items as guides to dynamically snap to as you move or reshape an item;
- *Clear Guides* to remove all current guides;
- *Show Bounding box* around the items to better identify your selection;
- *Show Rules* around the layout;
- *Show Pages* or set up pages to transparent. Often layout is used to create non-print layouts, e.g. for inclusion in presentations or other documents, and it's desirable to export the composition using a totally transparent background. It's sometimes referred to as "infinite canvas" in other editing packages.

In the print layout, you can change the zoom level using the mouse wheel or the slider and combo box in the status bar. If you need to switch to pan mode while working in the layout area, you can hold the `Spacebar` or the mouse wheel. With `Ctrl+Spacebar`, you can temporarily switch to Zoom In mode, and with `Ctrl+Alt+Spacebar`, to Zoom Out mode.

Panels and toolbars can be enabled from the *View* ► menu. To maximise the space available to interact with a composition you can check the *View* ► *Toggle Panel Visibility* option or press `Ctrl+Tab`; all panels are hidden and only previously visible panels are restored when unchecked.

It's also possible to switch to a full screen mode to have more space to interact with by pressing `F11` or using *View* ► *Toggle Full Screen*.

Tabela18.3: “View” Menu Tools

Ferramenta	Atalho	Barra de Ferramentas	Referencia
Atualizar	F5	*Navegação*	
Preview ►			
Zoom In	Ctrl++	*Navegação*	
Reduzir Zoom	Ctrl+-	*Navegação*	
Zoom to 100%	Ctrl+1	*Navegação*	
Zoom Full	Ctrl+0	*Navegação*	
Zoom to Width			
Show Grid	Ctrl+'		*Guides and Grid*
Snap to Grid	Ctrl+Shift+'		*Guides and Grid*
Show Guides	Ctrl+;		*Guides and Grid*
Snap to Guides	Ctrl+Shift+;		*Guides and Grid*
Smart Guides	Ctrl+Alt+;		
guilabel:*Gerenciamento de guias…*			*O Painel de Guias*
Limpar guias			*O Painel de Guias*
Show Rulers	Ctrl+R		
Show Bounding Boxes	Ctrl+Shift+B		
Show Pages			
Barra de Ferramentas ►			*Painéis e Barras de Ferramentas*
Painéis ►			*Painéis e Barras de Ferramentas*
Toggle Full Screen	F11		*Ver*
Toggle Panel Visibility	Ctrl+Tab		*Ver*

Items menu

The *Items* helps you configure items' position in the layout and the relations between them (see *Interagindo com itens de layout*).

Tabela18.4: “Items” Menu Tools

Ferramenta	Atalho	Barra de Ferramentas	Referencia
Group	Ctrl+G	*Actions*	*Agrupando itens*
Ungroup	Ctrl+Shift+G	*Actions*	*Agrupando itens*
Raise	Ctrl+]	*Actions*	*Alinhamento e distribuição*
Lower	Ctrl+[	*Actions*	*Alinhamento e distribuição*
Bring to Front	Ctrl+Shift+]	*Actions*	*Alinhamento e distribuição*
Send to Back	Ctrl+Shift+[	*Actions*	*Alinhamento e distribuição*
Lock Selected Items	Ctrl+L	*Actions*	*Bloqueando itens*
Unlock All	Ctrl+Shift+L	*Actions*	*Bloqueando itens*
Alinhar Itens ►		*Actions*	*Alinhamento e distribuição*
Distribuir itens ►		*Actions*	*Movendo e redimensionando itens*
Redimensionar ►		*Actions*	*Movendo e redimensionando itens*

Add Item menu

These are tools to create layout items. Each of them is deeply described in *Itens de layout* chapter.

Tabela18.5: “Add Item” Menu Tools

Ferramenta	Barra de Ferramentas	Referencia
Add Map	*Toolbox*	*Ítem do Mapa*
Add 3D Map	*Toolbox*	*O item de mapa 3D*
Adicionar Imagem	*Toolbox*	*O Item de Imagem*
Adicionar rótulo	*Toolbox*	*A rotulagem de item*
Adicionar Texto Dinâmico ►		*The Label Item*
Add Legend	*Toolbox*	*O Item Legenda*
Add Scale Bar	*Toolbox*	*Item da Barra de Escala*
Add North Arrow	*Toolbox*	*O Item de Seta Norte*
Add Shape ►	*Toolbox*	*Os Itens regulares do Shape*
► *Add Rectangle*	*Toolbox*	*Os Itens regulares do Shape*
► *Add Ellipse*	*Toolbox*	*Os Itens regulares do Shape*
► *Adicionar Triângulo*	*Toolbox*	*Os Itens regulares do Shape*
Adicionar Marcador	*Toolbox*	*O item marcador*
Adicionar Seta	*Toolbox*	*Item da Seta*
Adicionar Nó ►	*Toolbox*	*Os itens de forma baseados em nós*
► *Adicionar Polígono*	*Toolbox*	*Os itens de forma baseados em nós*
► *Add Polyline*	*Toolbox*	*Os itens de forma baseados em nós*
Adicionar HTML	*Toolbox*	*O quadro com item em HTML*
Add Attribute Table	*Toolbox*	*Item da tabela de atributos*
Add Fixed Table	*Toolbox*	*O item de tabela fixa*
Add Elevation Profile	*Toolbox*	*The Elevation Profile Item*

Menu do Atlas

Tabela18.6: “Atlas” Menu Tools

Ferramenta	Atalho	Barra de Ferramentas	Referencia
Pré-visualização do Atlas	`Ctrl+ALt+/`	*Atlas*	*Pré-visualizar e gerar um atlas*
|atlasPrimeira| *Primeira feição*	`Ctrl+<`	*Atlas*	*Pré-visualizar e gerar um atlas*
|atlasAnt| *Feição anterior*	`Ctrl+,`	*Atlas*	*Pré-visualizar e gerar um atlas*
|atlasProxima| *Próxima feição*	`Ctrl+.`	*Atlas*	*Pré-visualizar e gerar um atlas*
|atlasUltima| *Última feição*	`Ctrl+>`	*Atlas*	*Pré-visualizar e gerar um atlas*
|imprimirArquivo| *Imprimir Atlas…*		*Atlas*	*Pré-visualizar e gerar um atlas*
Exportar Atlas como Imagens…		*Atlas*	*Pré-visualizar e gerar um atlas*
|salvarComoSVG| *Exportar Atlas como SVG…*		*Atlas*	*Pré-visualizar e gerar um atlas*

continua na próxima página

Tabela 18.6 – continuação da página anterior

Ferramenta	Atalho	Barra de Ferramentas	Referencia
|salvarComoPDF| *Exportar Atlas como PDF…*		*Atlas*	*Pré-visualizar e gerar um atlas*
Configurações do Atlas		*Atlas*	*Gerar um Atlas*

Menu de configurações

The *Settings* ► *Layout Options…* menu is a shortcut to *Settings* ► *Options* ► *Layouts* menu of QGIS main canvas. Here, you can set some options that will be used as default on any new print layout:

- guilabel:`Padrões de layout' permite especificar a fonte padrão a ser utilizada;
- With *Grid appearance*, you can set the grid style and its color. There are three types of grid: **Dots**, **Solid** lines and **Crosses**;
- *Grid and guide defaults* defines spacing, offset and tolerance of the grid (see *Guides and Grid* for more details);
- *Layout Paths*: to manage list of custom paths to search print templates.

The *Settings* ► *Keyboard Shortcuts…* menu allows you to use the *shortcuts manager* in the print layout interface.

Menus contextuais

Depending on where you right-click in the print layout dialog, you open a contextual menu with various features:

- Right-click on the menu bar or any toolbar and you get the list of layout panels and toolbars you can enable or disable in one-click.
- Right-click over a ruler and you can *Show Guides*, *Snap to Guides*, *Manage Guides…* opening the *Guides panel* or *Clear Guides*. It's also possible to hide the rulers.
- Clique com o botão direito do mouse na tela de layout de impressão e:
  - You'll be able to *Undo* and *Redo* recent changes, or *Paste* any copied item (only available if no item is selected).
  - If you click over a page, you can additionally access the current *Page Properties* panel or *Remove Page*.
  - If you click on a selected item then you can cut or copy it as well as open the *Item Properties* panel.
  - If more than one item are selected, then you can either group them and/or ungroup if at least one group is already in the selection.
- Right-click inside a text box or spinbox widget of any layout panel provides edit options to manipulate its content.

O Painel de Layout

In the *Layout* panel, you can define the global settings of your print layout.

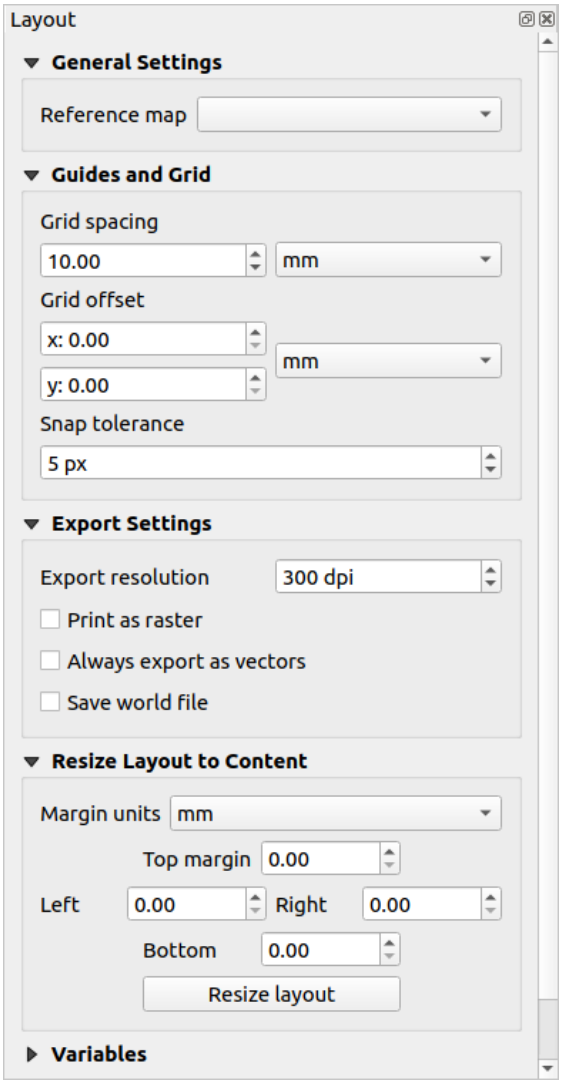


Figura18.3: Configurações de layout no layout de impressão

Configurações Gerais

In a print layout, you can use more than one map item. The *Reference map* represents the map item to use as the layout's master map. It's assigned as long as there's a map item in the layout. The layout will use this map in any of their properties and variables calculating units or scale. This includes exporting the print layout to georeferenced formats.

Moreover, new layout items such as scale bar, legend or north arrow have by default their settings (orientation, displayed layers, scale, …) bound to the map item they are drawn over, and fall back to the reference map if there is no overlapping map.

Guides and Grid

You can put some reference marks on your paper sheet to help you accurately place some items. These marks can be:

- simple horizontal or vertical lines (called **Guides**) put at the position you want (see *O Painel de Guias* for guides creation).
- or regular **Grid**: a network of horizontal and vertical lines superimposed over the layout.

Settings like *Grid spacing* or *Grid offset* can be adjusted in this group as well as the *Snap tolerance* to use for items. The tolerance is the maximum distance below which the mouse cursor is snapped to a grid or a guide, while moving, resizing or creating an item.

Whether grid or guides should be shown is set in *View* menu. There, you can also decide if they might be used to snap layout items. When both a grid line and a guide line are within tolerance of a point, guides will always take precedence - since they have been manually set (hence, assumption that they have been explicitly placed at highly desirable snapping locations, and should be selected over the general grid).

> **Nota**
>
> In the *Settings* ► *Layout Options* menu, you can also set the grid and guides parameters exposed above. However, these options will only apply as defaults to new print layouts.

Definições para exportação

You can define a resolution to use for all exported maps in *Export resolution*. This setting can then be overridden each time you export a map.

Because of some advanced rendering options (*blending mode*, *effects*…), a layout item may need rasterization in order to be exported correctly. QGIS will individually rasterize it without forcing every other item to also be rasterized. This allows printing or saving as PostScript or PDF to keep items as much as possible as vectors, e.g. a map item with layer opacity won't force labels, scale bars, etc to be rasterized too. You can however:

- force all the items to be rasterized checking the *Print as raster* box;
- or use the opposite option, i.e. *Always export as vectors*, to force the export to keep items as vectors when exported to a compatible format. Note that in some cases, this could cause the output to look different to layout.

Where the format makes it possible (e.g., `.TIF`, `.PDF`) exporting a print layout results by default in a georeferenced file (based on the *Reference map* item in the *General settings* group). For other formats, georeferenced output requires you to generate a world file by checking *Save world file*. The world file is created beside the exported map(s), has the name of the page output with the reference map item and contains information to georeference it easily.

Redimensionar o layout para o conteúdo

Using the *Resize page* tool in this group, you create a unique page composition whose extent covers the current contents of the print layout (with some optional *margins* around the cropped bounds).

Note that this behavior is different from the *crop to content* option in that all the items are placed on a real and unique page in replacement of all the existing pages.

Variáveis

The *Variables* lists all the variables available at the layout's level (which includes all global and project's variables).

It also allows the user to manage layout-level variables. Click the button to add a new custom layout-level variable. Likewise, select a custom layout-level variable from the list and click the button to remove it.

More information on variables usage in the *General Tools* section.

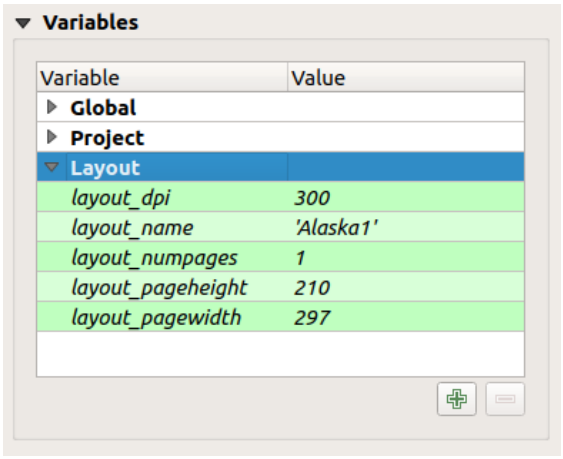


Figura18.4: Editor de Variáveis no Layout de Impressão

Trabalhando com as propriedades da página

A layout can be composed of several pages. For instance, a first page can show a map canvas, and a second page can show the attribute table associated with a layer, while a third one shows an HTML frame linking to your organization website. Or you can add many types of items on each page.

Adicionando uma nova página

Futhermore, a layout can be made using different size and/or orientation of pages. To add a page, select the *Add Pages...* tool from the *Layout* menu or *Layout Toolbar*. The *Insert Pages* dialog opens and you are asked to fill:

- o número de páginas a inserir;
- the position of the page(s): before or after a given page or at the end of the print layout;
- The *Page size*: it could be of a preset format page (`A4`, `B0`, `Legal`, `Letter`, `ANSI A`, `Arch A` and their derivatives as well as a resolution type, such as `1920x1080` or `1024x768`) with associated *Orientation* (Portrait or Landscape).

  The page size can also be of a `custom` format; In that case, you'd need to enter its *Width* and *Height* (with locked size ratio if needed) and select the unit to use among `mm`, `cm`, `px`, `pt`, `in`, `ft`... Conversion of entered values is automatically applied when switching from one unit to another.

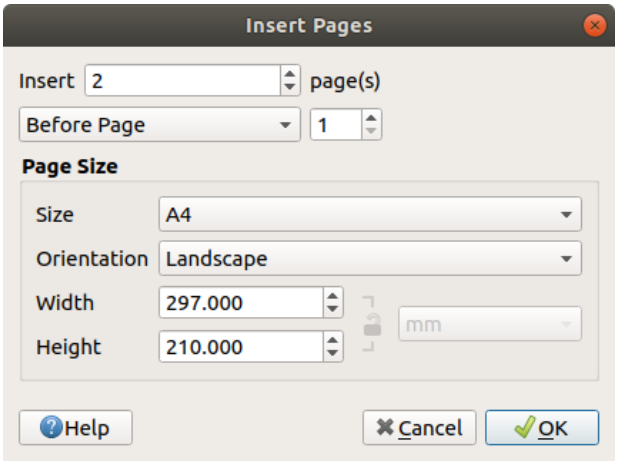


Figura18.5: Criando uma nova página no layout de impressão

Atualizando as propriedades da página

Any page can be later customized through the Page *Item Properties* panel. To access a page's properties, either:

- Left-click on an empty section of the page
- Right-click on a page and select *Page Properties...*.
- Go to *Layout* ► *Page properties...* while hovering over the page

The *Item Properties* panel opens with settings such as:

- for layouts with several pages, it displays the active page number and the count of available pages
- the *Page size* frame described above. You can modify each property using the data defined override options (see *Explore Data-defined override buttons with atlas* for a use case);
- the *Background* of the current page using the *color* or *symbol* you want.
- the *Apply to all Pages* button updates the current page properties to the other existing pages in the layout;
- the *Exclude page from exports* to control whether the current page with its content should be included in the *layout output*;

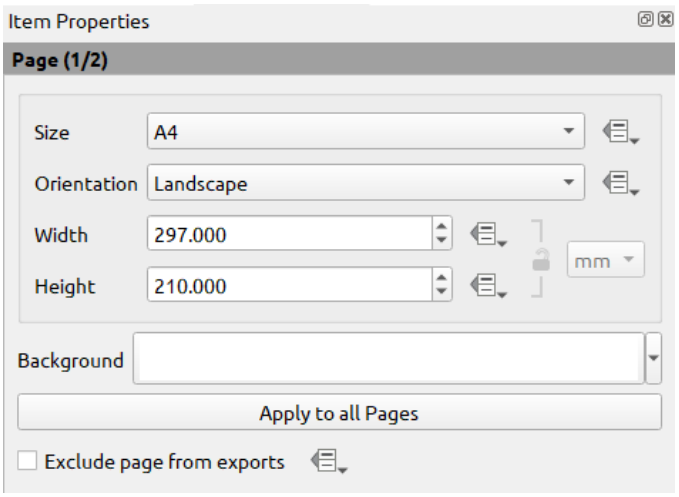


Figura18.6: Page properties dialog

O Painel de Guias

Guides are vertical or horizontal line references you can place on a layout page to assist you on items placement, when creating, moving or resizing them. To be active, guides require the *View* ► *Show Guides* and *View* ► *Snap to Guides* options to be checked. To create a guide, there are two different methods:

- if the *View* ► *Show Rulers* option is set, drag out a ruler and release the mouse button within the page area, at the desired position.
- for more precision, use the *Guides* panel from the *View* ► *Toolbox* ► or by selecting *Manage guides for page…* from the page's contextual menu.

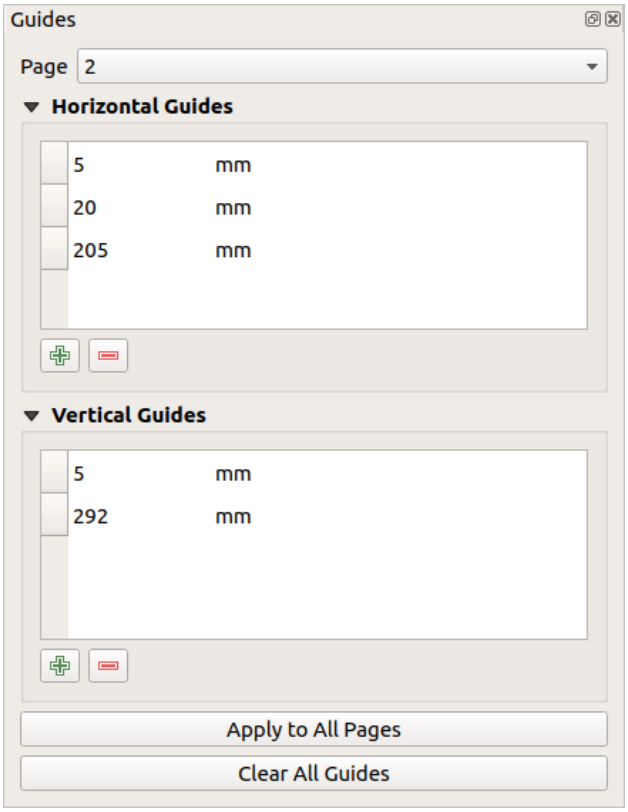


Figura18.7: The Guides panel

The *Guides* panel allows creation of snap lines at specific locations:

1. Select the *Page* you'd like to add the guides to
2. Click the Add new guide button and enter the coordinates of the horizontal or vertical line. The origin is at the top left corner. Different units are available for this.

   The panel also allows adjusting the position of existing guides to exact coordinates: double-click and replace the value.
3. The *Guides* panel lists only the items for the current page. It allows creation or removal of guides only in the current page. However, you can use the *Apply to All Pages* button to replicate the guide configuration of the current page to the other pages in the layout.
4. To delete a guide, select it and press the Remove selected guide button. Use *Clear All Guides* to remove all the guides in the current page.

> **Dica**
>
> **Snapping to existing layout items**
>
> Other than guides and grids, you can use existing items as snapping references when moving, resizing or creating new items; these are called **smart guides** and require *View* ► *Smart Guides* option to be checked. Anytime the mouse pointer is close to an item's bound, a snapping cross appears.

The Items Panel

The *Items* panel offers some options to manage selection and visibility of items. All the items added to the print layout canvas (including *items group*) are shown in a list and selecting an item makes the corresponding row selected in the list as well as selecting a row does select the corresponding item in the print layout canvas. This is thus a handy way to select an item placed behind another one. Note that a selected row is shown as bold. Multiple items selection is also possible holding `Shift` for contiguous items, and `Ctrl` key for non-contiguous ones.

For any available item, you can:

- set it visible or not;
- lock or unlock its position;
- sort its Z position. You can move up and down each item in the list with a click and drag. The upper item in the list will be brought to the foreground in the print layout canvas. By default, a newly created item is placed in the foreground.
- change the item ID by double-clicking the text;
- right-click an item and select whether to copy or delete it or open its *properties panel*.

Once you have found the correct position for an item, you can lock it by ticking the box in column. Locked items are **not** selectable on the canvas. Locked items can be unlocked by selecting the item in the *Items* panel and unchecking the tickbox or you can use the icons on the toolbar.

The Undo History Panel: Revert and Restore actions

During the layout process, it is possible to revert and restore changes. This can be done with the revert and restore tools available in the *Edit* menu, the *Layout* toolbar or the contextual menu any time you right-click in the print layout area:

- Revert last change
- Restore last change

This can also be done by mouse click within the *Undo history* panel (see Fig.18.8). The History panel lists the last actions done within the print layout. Select the point you want to revert to and once you do a new action all the actions done after the selected one will be removed.

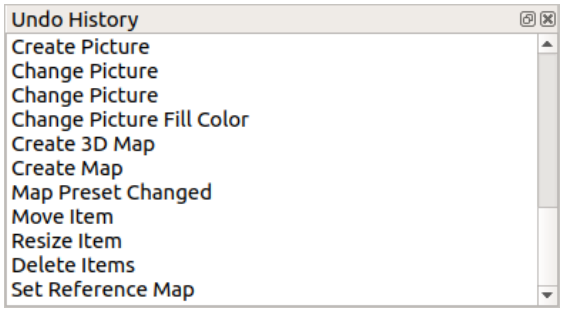


Figura18.8: Undo History in the Print Layout

18.2 Itens de layout

18.2.1 Opções comuns de itens de esquema

O QGIS fornece um grande conjunto de itens para o layout de um mapa. Podem ser de mapa, legenda, barra de escala, figura, tabela, seta norte, tipo de imagem… Entretanto, compartilham algumas opções e comportamentos comuns que são expostos abaixo.

Criando um item esquema

Os itens podem ser criados usando diferentes ferramentas, seja do zero ou com base em itens existentes.

Para criar um item de layout do zero:

1. Selecione a ferramenta correspondente no menu: seleção de menu:*Adicionar Item* ou na barra *Caixa de ferramentas*.
2. Então:
   - Clique na página e preencha as informações de tamanho e posicionamento solicitadas na caixa de diálogo *Propriedades do novo item* que aparece (para detalhes, veja posição_tamanho);

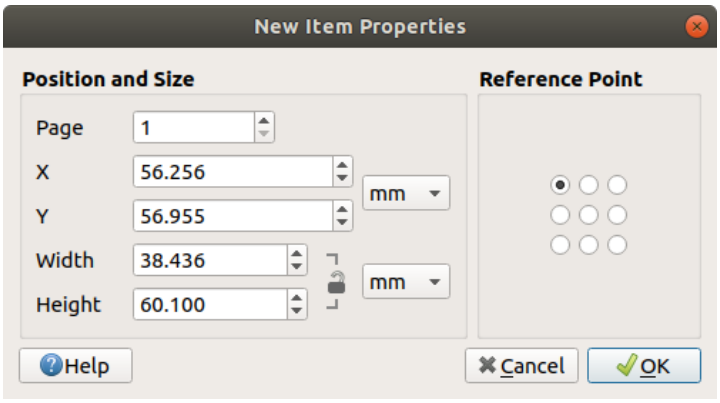


Figura18.9: Caixa de diálogo de propriedades do novo item

   - Ou clique e arraste para definir o tamanho inicial e o posicionamento do item. Você pode confiar em grades e guias 1 para uma melhor posição.

Nota

Como eles podem ter formas específicas, os itens de nó ou seta de desenho não funcionam com métodos de um clique nem clique e arraste; você precisa clicar e colocar cada nó do item. Veja esquema_nó_baseado_shape_item para mais detalhes.

Você também pode:

1. Selecione um item existente com o |selecionar| Botão Selecionar/Mover item da barra de ferramentas *Caixa de ferramentas*
2. Use o menu contextual ou as ferramentas do menu :selecionar menu:*Editar* para copiar/cortar o item e colá-lo na posição do mouse como um novo item.

   Você também pode usar o comando *Colar no local* (`Ctrl+Shift+V`) para duplicar um item de uma página para outra e colocá-lo na nova página nas mesmas coordenadas do original.

Além disso, você pode criar itens usando um modelo de esquema de impressão (para detalhes, veja esquema_gerente) através do comando :seleção de menu:*Esquema –> Adicionar itens do modelo…*.

Dica

Adicione itens de esquema usando o navegador de arquivos

A partir do seu navegador de arquivos ou usando o painel *Navegador*, arraste e solte um modelo de layout de impressão (arquivo:*.qpt*) em uma caixa de diálogo de layout de impressão e o QGIS adiciona automaticamente todos os itens desse modelo ao disposição.

Interagindo com itens de layout

Cada item dentro do layout de impressão pode ser movido e redimensionado para criar um layout perfeito. Para ambas as operações, o primeiro passo é ativar o |selecionar| Selecionar/Mover item e clique no item.

Você pode selecionar vários itens com o |selecionar| Botão Selecionar/Mover item: clique e arraste sobre os itens ou segure o botão `Shift` e clique em cada um dos itens desejados. Para desmarcar um item, clique nele segurando o botão `Shift`.

Cada vez que há uma seleção, a contagem de itens selecionados é exibida na barra de status. Dentro do menu :seleção de menu:*Editar*, você pode encontrar ações para selecionar todos os itens, limpar todas as seleções, inverter a seleção atual e muito mais…

Movendo e redimensionando itens

Unless *View* ► *Show Bounding Boxes* option is unchecked, a selected item will show squares on its boundaries; moving one of them with the mouse will resize the item in the corresponding direction. While resizing, holding `Shift` will maintain the aspect ratio. Holding `Alt` will resize from the item center.

To move a layout item, select it with the mouse and move while holding the left button. If you need to constrain the movements to the horizontal or vertical axis, hold the `Shift` button on the keyboard while moving the mouse. You can also move a selected item using the `Arrow keys` on the keyboard; if the movement is too slow, you can speed it up by holding `Shift`. If you need better precision, use the *Position and size* properties, or grid/guides snapping as explained above for item's creation.

Redimensionar ou mover vários itens de uma vez é feito da mesma forma que para um único item. O QGIS, no entanto, fornece algumas ferramentas avançadas para redimensionar automaticamente uma seleção de itens seguindo regras diferentes:

- a altura de cada item corresponde ao |redimensionar mais alto| mais alto ou o |redimensionar mais curto| item selecionado mais curto;

- a largura de cada item corresponde ao **|redimensionar mais largo|** mais largo ou o **|redimensionar mais estreito|** item selecionado mais restrito;
- resizes items to squares: each item is enlarged to shape a square.

Da mesma maneira, existem *tools* disponíveis para organizar a posição dos vários itens distribuindo equidistantemente:

- bordas (esquerda, direita, superior ou inferior) dos itens;
- Centros de itens horizontalmente ou verticalmente;
- espaçamento entre itens horizontalmente ou verticalmente.

Agrupando itens

Agrupar itens permite manipular um conjunto de itens como um único: você pode facilmente redimensionar, mover, excluir, copiar os itens como um todo.

To create a group of items, select more than one and press the . *Group* button on the *Items* menu or the *Actions* toolbar, or from the right-click menu. A row named `Group` is added to the *Items* panel and can be locked or hidden like any other *Items panel's object*. Grouped items are **not individually** selectable on the canvas; use the Items panel for direct selection and access the item's properties panel.

Bloqueando itens

Once you have found the correct position for an item, you can lock it by using the *Lock selected items* button in the *Items* menu or the *Actions* toolbar, or ticking the box next to the item in the *Items* panel. Locked items are **not** selectable on the canvas.

Itens bloqueados podem ser desbloqueados selecionando o item no painel :seleção de menu:*Itens* e desmarcando a caixa de seleção ou você pode usar os ícones na barra de ferramentas.

Alinhamento e distribuição

Tools for raising or lowering the Z position of items in the layout are inside the Raise selected items pull-down menu. Choose an element on the print layout canvas and select the matching functionality to raise or lower the selected element over the other elements. This order is shown in the *Items* panel. You can also raise or lower objects in the *Items* panel by clicking and dragging an object's label in this list.

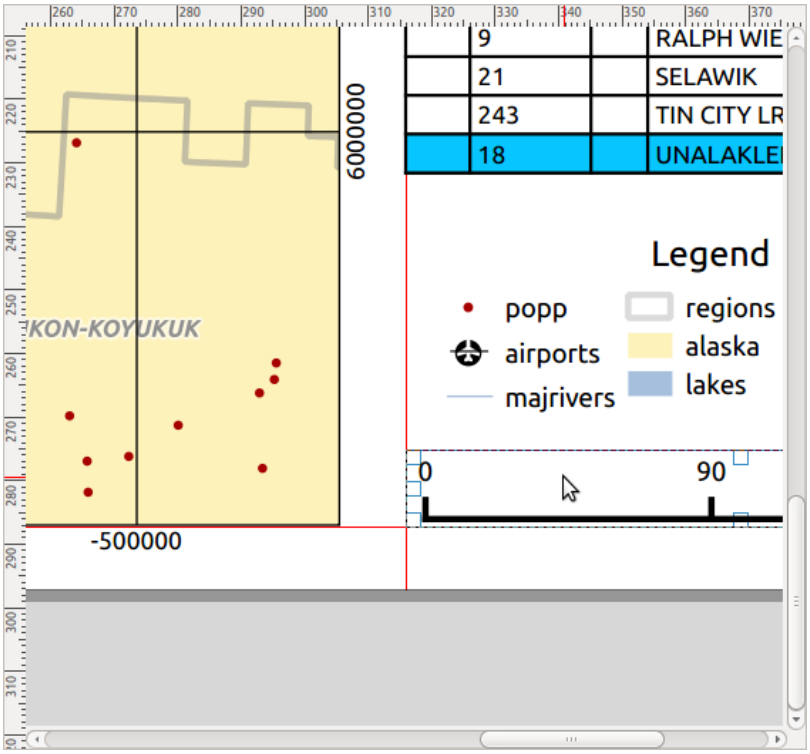


Figura18.10: Linhas auxiliares de alinhamento no layout de impressão

Existem várias opções de alinhamento disponíveis no **|alinhar à esquerda|** Menu suspenso `Alinhar itens selecionados (veja `figura_layout_comum_alinhar`). Para usar uma função de alinhamento, primeiro selecione os elementos e clique em um dos ícones de alinhamento:

- **|alinhar Esquerda|** *Alinhar à esquerda* ou **|alinhar Direita|** *Alinhar à direita*;
- **|alinhar Topo|** *Alinhar Topo* ou **|alinhar Inferior|** *Alinhar Inferior*;
- **|alinhado ao Centro H|** *Centralizar* Horizontalmente ou **|alinhado o centro V|** *Alinhar Centro Vertical*.

Todos os elementos selecionados serão alinhados à sua caixa delimitadora comum. Ao mover itens na tela de layout, as linhas auxiliares de alinhamento aparecem quando as bordas, centros ou cantos são alinhados.

Outra forma de melhorar o posicionamento dos itens de layout é ajustando o espaçamento entre eles, sobre a página de layout. Isso pode ser feito selecionando os itens e pressionando *Distribute Left Edges* no menu suspenso para:

- *Distribute Left Edges* ou *Distribute Right Edges* de itens de forma equidistante
- *Distribute Top edges* ou *Distribute Bottom Edges* de itens de forma equidistante
- *Distribute Horizontal Centers* ou *Distribute Vertical Centers* de itens de forma equidistante
- Adicione espaço igual entre os itens: *Distribute Horizontal Spacing Equally* or *Distribute Vertical Spacing Equally*

Propriedades comuns dos itens

Layout items have a set of common properties you will find at the bottom of the *Item Properties* panel: Position and size, Rotation, Frame, Background, Item ID, Variables and Rendering (see Fig.18.11).

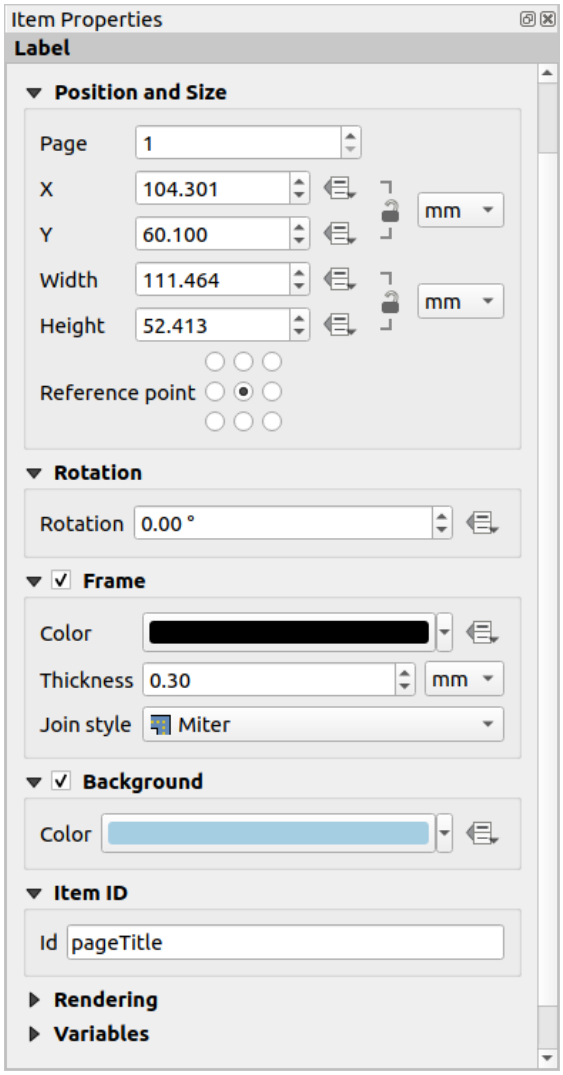


Figura18.11: Grupos de propriedades de itens comuns

Nota

O |Definir dados| O ícone Substituição definida por dados ao lado da maioria das opções significa que você pode associar essa propriedade a uma camada, atributos de feições, geometria ou a qualquer propriedade de outro item de layout, usando expressão 1 ou variáveis 2. Para mais informações veja definir_dados.

- O grupo *Posição e tamanho* permite definir o tamanho e a posição do quadro que contém o item (veja posição_tamanho para mais informações).
- O *Rotação* define a rotação do elemento (em graus).
- A |caixa de seleção| *Frame* mostra ou oculta a moldura ao redor do item. Use as ferramentas *Cor*, *Espessura* e *Estilo de junção* para ajustar essas propriedades.
- In the *Background* group you can create or pick a background *Color*. Transparency can be adjusted through altering the alpha field settings.

- Use o *Item ID* para criar um relacionamento com outros itens de layout de impressão. Isso é usado com o servidor QGIS e outros clientes da web em potencial. Você pode definir um ID em um item (por exemplo, um mapa ou um rótulo) e, em seguida, o cliente da Web pode enviar dados para definir uma propriedade (por exemplo, texto do rótulo) para esse item específico. O comando Obter configurações do projeto listará os itens e IDs que estão disponíveis em um layout.
- O modo *Renderização* ajuda a definir se e como o item pode ser exibido: você pode, por exemplo, aplicar modo de mesclagem 1, ajustar a opacidade do item ou *Excluir item das exportações* .

Posição e tamanho

Estendendo os recursos da caixa de diálogo Propriedades do novo item 1 com recursos definidos por dados, este grupo permite que você coloque os itens com precisão.

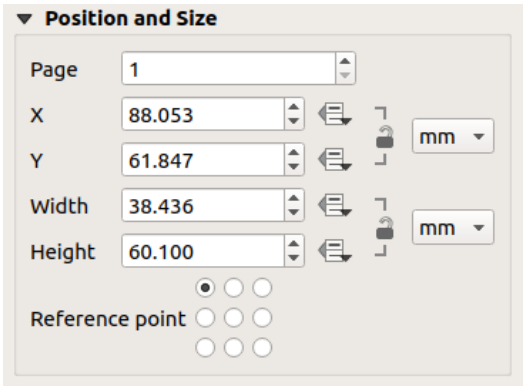


Figura18.12: Posição e tamanho

- o número real da página para colocar o item;
- o ponto de referência do item;
- as coordenadas *X* e *Y* do *Ponto de referência* do item na página escolhida. A relação entre esses valores pode ser bloqueada clicando no botão **|bloqueado cinza|** botão. Alterações feitas em um valor usando o widget ou o A ferramenta Selecionar/Mover item será refletida em ambos;
- o *Largura* e *Altura* da caixa delimitadora do item. Quanto às coordenadas, a relação entre largura e altura pode ser bloqueada.

Modo de renderização

O QGIS permite renderização avançada para itens de layout, como camadas vetoriais e raster.

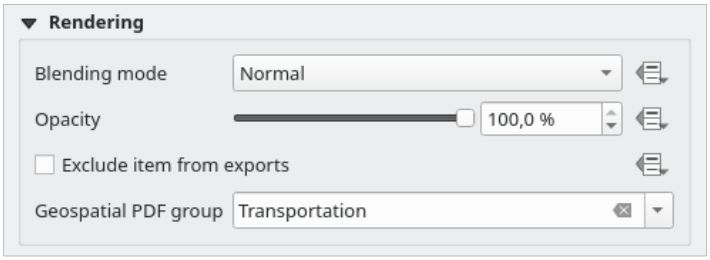


Figura18.13: Modo de renderização

- *Modo de mesclagem*: Com esta ferramenta você pode obter efeitos que de outra forma só seriam alcançados usando software de renderização gráfica. Os pixels de seus itens de sobreposição e subjacência podem ser misturados de acordo com o modo definido (veja modo-mesclagem para descrição de cada efeito).
- *Opacity*: You can make the underlying item in the layout visible with this tool. Use the slider to adapt the visibility of your item to your needs. You can also make a precise definition of the percentage of visibility in the menu beside the slider.
- *Exclude item from exports*: Combined with data-defined expressions, you can decide to make an item invisible in some exports. After activating this checkbox, the item will not be included in export to PDF, print etc..
- *Geospatial PDF group*: When set, a matching layer tree group will be created in the exported Geospatial PDF and the item will only be visible when this group is checked. This allows content to be selectively displayed as a group by viewers of the Geospatial PDF. E.g., it can allow extra layout content such as descriptive labels or legends to only be shown when layers from the group are visible, making the file export much more flexible.

Variáveis

The *Variables* lists all the variables available at the layout item's level (which includes all global, project and layout's variables). Layout map items alhave an additional *Map settings* section for variables that provide easy access to values like the map's scale, extent, and so on.

In *Variables*, it's also possible to manage layout item level variables. Click the button to add a new custom variable. Likewise, select any custom item-level variable from the list and click the button to remove it.

Mais informações sobre o uso de variáveis na seção geral_ferramentas_variáveis.

18.2.2 Ítem do Mapa

O item do mapa é o quadro principal que exibe o mapa que você projetou na tela do mapa. Use o **|add Mapa|** *Add Mapa* seguindo as instruções de criação de itens 1 para adicionar um novo item de mapa que você pode manipular mais tarde da mesma forma como exposto em interagir_layout_item.

Por padrão, um novo item de mapa mostra o status atual do tela do mapa 1 ` com sua extensão e camadas visíveis. Você pode personalizá-lo graças ao painel :guilabel:`Propriedades do item 2. Além das :ref:`propriedades comuns dos itens `, este recurso tem as seguintes funcionalidades:

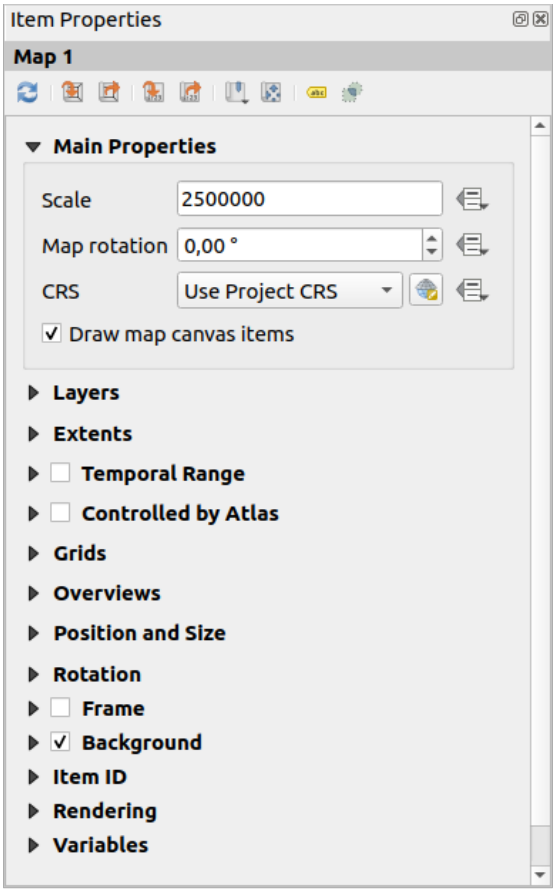


Figura18.14: Painel de propriedades do item de mapa

A Barra de Ferramentas

O painel Mapa *Propriedades do item* incorpora uma barra de ferramentas com as seguintes funcionalidades:

- |atualizar| Atualizar visualização do mapa
- |definir para a extensão da tela| Defina a tela do mapa para corresponder à extensão da tela principal
- |exibir extensão na tela| Ver a extensão atual do mapa na tela principal
- |definir para escala de tela| Defina a escala do mapa para corresponder à escala da tela principal
- |ver Escala na tela|: sup:*Defina a tela principal para corresponder à escala atual do mapa*
- |mostrar favoritos|: sup:*Bookmarks*: define a extensão do item do mapa para corresponder a um espaço existente
- |mover o conteúdo do item|Editar extensão do mapa interativamente: panorâmica e zoom interativamente dentro do item do mapa
- |rotulagem Singular| Configurações de rotulagem: controla o comportamento do rótulo da feição (posicionamento, visibilidade…) na extensão do item do mapa de layout:
  - defina a *Margem da borda do mapa*, uma distância definível de dados dos limites do item do mapa dentro da qual nenhum rótulo deve ser exibido
  - |desmarcado| *Permitir rótulos truncados nas bordas do mapa*: controla se os rótulos que estão parcialmente fora da extensão permitida do item do mapa devem ser renderizados. Se marcado, esses rótulos serão mostrados (quando não houver como colocá-los totalmente dentro da área visível). Se desmarcado, os rótulos parcialmente visíveis serão ignorados.

  - *Itens de bloqueio de rótulos*: permite que outros itens de layout (como barras de escala, setas para o norte, mapas inseridos, etc) sejam marcados como bloqueadores para os rótulos do mapa no item de mapa **ativo**. Isso impede que qualquer rótulo de mapa seja colocado sob esses itens - fazendo com que o mecanismo de rotulagem tente um posicionamento alternativo para esses rótulos ou os descarte completamente.

    Se um *Margem das bordas do mapa* for definido, os rótulos do mapa não serão colocados mais próximos do que a distância especificada dos itens de layout verificados.

  - *Mostrar rótulos não colocados*: pode ser usado para determinar se os rótulos estão faltando no mapa de layout (por exemplo, devido a conflitos com outros rótulos de mapa ou devido a espaço insuficiente para colocar o rótulo) destacando-os em um :ref:` cor predefinida `

- |recorte|: sup:*Configurações de recorte*: permite recortar o item do mapa para o recurso atlas e para moldar e polígonos itens:

  - |caixa de seleção|: guilabel:*Recurso de recorte para atlas*: você pode determinar que o item do mapa de layout será cortado automaticamente no :ref:`recurso de atlas `.

    Existem diferentes modos de recorte disponíveis:

    * *Recortar ao renderizar apenas*: aplica um recorte e baseado em pintor, de modo que partes dos recursos vetoriais que ficam fora do recurso atlas se tornem invisíveis
    * *Recurso de recorte antes de renderizar*: aplica o clipe antes de renderizar os recursos, então as bordas dos recursos que ficam parcialmente fora do atlas ainda estarão visíveis no limite do atlas
    * *Renderizar feições de interseção inalteradas*: renderiza todas as feições que cruzam a feição atual do atlas, mas sem cortar sua geometria.

    You can *Force labels inside atlas feature*. If you don't want to *Clip all layers* to the atlas feature you can use the *Clip selected layers* option.

  - |caixa de seleção| *Recortar para o item*: é possível alterar a forma do item do mapa usando um item shape 1 ou polígono 2 do layout de impressão. Quando você habilita esta opção, o mapa será automaticamente recortado na forma selecionada na caixa de combinação. Novamente, os modos de recorte mencionados acima estão disponíveis e os rótulos podem ser forçados a serem exibidos apenas dentro da forma de recorte.

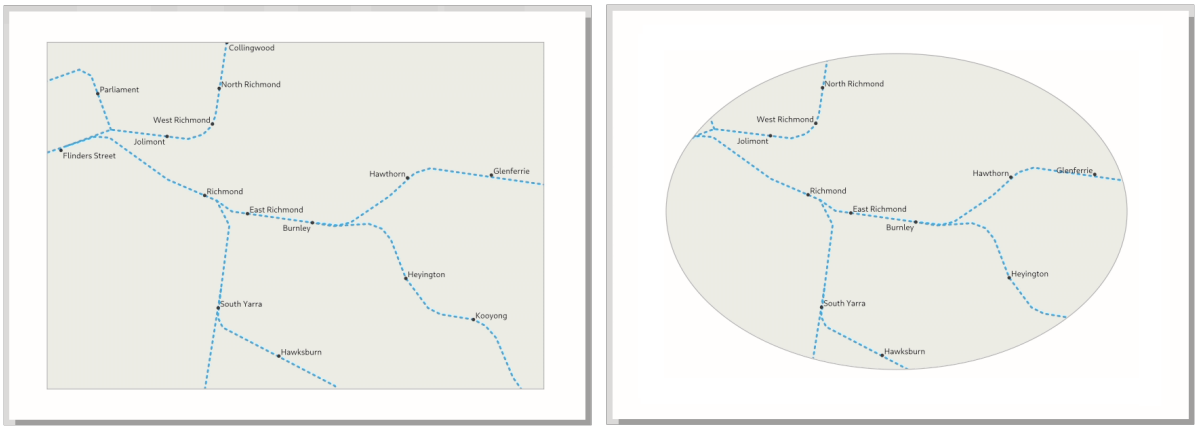


Figura18.15: Clipping a layout map item to shapes

Propriedades principais

No grupo *Propriedades principais* (veja `figura_layout_mapa`) do painel mapa *Propriedades do item*, as opções disponíveis são:

- O botão *Atualizar visualização* para atualizar a renderização do item do mapa se a visualização na tela do mapa tiver sido modificada. Observe que, na maioria das vezes, a atualização do item do mapa é acionada automaticamente pelas alterações;
- O *Escala* para definir manualmente a escala do item do mapa;
- O *Rotação do mapa* permite que você gire o conteúdo do item do mapa no sentido horário em graus. A rotação da tela do mapa pode ser imitada aqui;
- O *CRS* permite que você exiba o conteúdo do item do mapa em qualquer CRS 1 `. O padrão é ``Usar projeto CRS`;
- **|caixa de seleção|** *Desenhar itens da tela do mapa* permite que você mostre no layout de impressão anotações 1 que são colocadas na tela principal do mapa.

Camadas

Por padrão, a aparência do item do mapa é sincronizada com a renderização da tela do mapa, o que significa que alternar a visibilidade das camadas ou modificar seu estilo no *Painel de Camadas* é aplicado automaticamente ao item do mapa. Porque, como qualquer outro item, você pode querer adicionar vários itens de mapa a um layout de impressão, é necessário quebrar essa sincronização para permitir mostrar diferentes áreas, combinações de camadas, em diferentes escalas… O *Camadas `grupo de propriedades (veja :numref:`figura_layout_mapa_camadas`)* ajuda você a fazer isso.

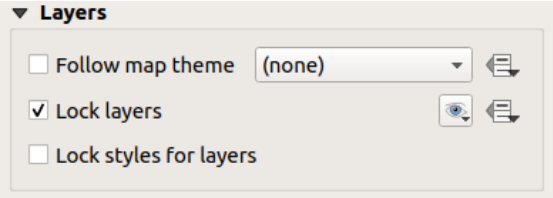


Figura18.16: Grupo de Camadas de Mapa

Se você quiser manter o item do mapa consistente com um tema do mapa ` existente, marque |caixa de seleção| :guilabel:`Siga o tema do mapa e selecione o tema desejado na lista suspensa. Quaisquer alterações aplicadas ao tema na janela principal do QGIS (usando a função substituir tema) afetarão automaticamente o item do mapa. Se um tema de mapa for selecionado, a opção *Bloquear estilos para camadas* será desabilitada porque *Seguir tema de mapa* também atualiza o estilo (simbologia, rótulos, diagramas) das camadas.

Para bloquear as camadas mostradas em um item do mapa para a visibilidade da tela do mapa atual, marque **|caixa de seleção|** *Bloquear camadas*. Quando esta opção está habilitada, qualquer mudança na visibilidade das camadas na janela principal do QGIS não afetará o item de mapa do layout. No entanto, o estilo e os rótulos das camadas bloqueadas ainda são atualizados de acordo com a janela principal do QGIS. Você pode evitar isso usando *Bloquear estilos para camadas*.

Em vez de usar a tela de mapa atual, você também pode bloquear as camadas do item de mapa para aquelas de um tema de mapa existente: selecione um tema de mapa em **|mostrar predefinições|** Definir lista de camadas de um tema de mapa, e a **|caixa de seleção|** *Bloquear camadas* está ativado. O conjunto de camadas visíveis no tema do mapa é agora usado para o item do mapa até que você selecione outro tema do mapa ou desmarque a **|caixa de seleção|** :guilabel:opção *Bloquear camadas*. Você pode precisar atualizar a visualização usando o comando **|atualizar|** atualizar| view da barra de ferramentas *Navegação* ou o botão *Atualizar visualização* visto acima.

Observe que, diferentemente da opção *Siga o tema do mapa*, se a opção :guilabel:`Bloquear camada` estiver habilitada e configurada para um tema de mapa, as camadas no item de mapa não serão atualizadas mesmo que o tema do mapa seja atualizado (usando a função substituir tema) na janela principal do QGIS.

Camadas bloqueadas no item de mapa também podem ser definido por dados1, usando o **|Dados definidos|** ícone ao lado da opção. Quando usado, isso substitui a seleção definida na lista suspensa. Você precisa passar uma lista de camadas separadas pelo caractere `|`. O exemplo a seguir bloqueia o item do mapa para usar apenas as camadas `camada 1` e `camada 2`:

```
concat ('layer 1', '|', 'layer 2')
```

Extensões

O grupo *Extensões* do painel de propriedades do item do mapa fornece as seguintes funcionalidades (veja :num ref:*figura_layout_mapa_extenções*):

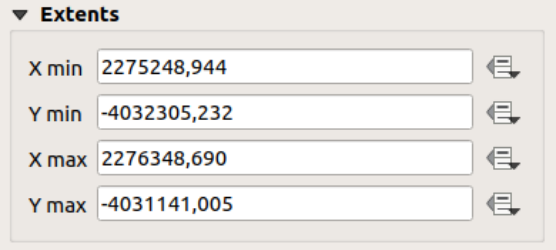


Figura18.17: Grupo de Extensões do Mapa

A área **Extensões** exibe as coordenadas `X` e `Y` da área mostrada no item do mapa. Cada um desses valores pode ser substituído manualmente, modificando a área de tela do mapa exibida e/ou o tamanho do item do mapa. A extensão também pode ser modificada usando ferramentas na parte superior do painel de itens do mapa, como:

- **|definir para a extensão da tela|** Defina a tela do mapa para corresponder à extensão da tela principal
- **|definir para escala de tela|** Defina a escala do mapa para corresponder à escala da tela principal

Você também pode alterar a extensão de um item de mapa usando o **|mover o conteúdo do item|** :sup:Ferramenta *Mover o conteúdo do item*: clique e arraste dentro do item do mapa para modificar sua visualização atual, mantendo a mesma escala. Com o **|mover o conteúdo do item|** ferramenta habilitada, use a roda do mouse para aumentar ou diminuir o zoom, modificando a escala do mapa mostrado. Combine o movimento com a tecla `Ctrl` pressionada para ter um zoom menor.

Elevation range

The *Elevation range* setting in the map item properties allows you to filter the content of certain layers based on a specific elevation range. While all layers remain visible, those that support elevation filtering (currently point clouds and raster DEMs) will have their data filtered. Only the parts of these layers that fall within the elevation range set by the *Lower* and *Upper* values will be displayed.

The *Elevation range* can be data-defined. This means you can set different elevation ranges for different features in the *atlas* or *report*.

Intervalo temporal

O grupo *Intervalo temporal* do painel de propriedades do item do mapa fornece as opções para controlar a renderização das camadas no item do mapa com base em um intervalo temporal. Somente as camadas cujas propriedades temporais se sobrepõem ao intervalo de tempo definido pelas datas *Começo* e *Fim* são exibidas no item do mapa.

As ferramentas definidas por dados associados ajudam a tornar o intervalo de tempo dinâmico e permitem a saída de atlas 1 temporais, ou seja, mapas automatizados com extensão espacial fixa e cujos conteúdos variam com base no tempo. Por exemplo, usando como camada de cobertura um arquivo csv com um par de campos inicial e final e várias linhas representando intervalos de datas, habilite o intervalo temporal e o controle por atlas nas propriedades do item do mapa e clique na exportação de atlas.

Controlado por atlas

A **|caixa de seleção|** As propriedades do grupo *Controlado por atlas* estão disponíveis somente se um atlas 1 estiver ativo no layout de impressão. Marque esta opção se desejar que o item do mapa seja regido pelo atlas; ao iterar sobre a camada de cobertura, a extensão do item do mapa é panorâmica/ampliada para o recurso de atlas a seguir:

- **|botão de rádio ligado|** *Margem ao redor das feições*: amplia a feição na melhor escala, mantendo ao redor de cada uma margem representando uma porcentagem da largura ou altura do item do mapa. A margem pode ser a mesma para todas as feições ou definir variável, por exemplo, dependendo da escala do mapa;
- **|botão de rádio desligado|** *Escala predefinida (melhor ajuste)*: amplia a feição no projeto :ref:`escala predefinida ` onde a feição do atlas se encaixa melhor;
- **|botão de rádio desligado|** *Escala fixa*: as feições do atlas são movidas de uma para outra, mantendo a mesma escala do item do mapa. Ideal ao trabalhar com recursos do mesmo tamanho (por exemplo, uma grade) ou para destacar as diferenças de tamanho entre os recursos do atlas.

Grades

Com as grades, você pode adicionar, sobre seu mapa, informações relativas à sua extensão ou coordenadas, seja na projeção do item do mapa ou em outra. O grupo *Grades* oferece a possibilidade de adicionar várias grades a um item do mapa.

- Com a **|Adicionar simbologia|** e **|Remover simbologia|** botões você pode adicionar ou remover uma grade selecionada;
- Com a **|seta para cima|** e **|seta para baixo|** botões você pode mover para cima e para baixo uma grade na lista, portanto, mova-a para cima ou para baixo de outra, sobre o item do mapa.

Clique duas vezes na grade adicionada para renomeá-la.

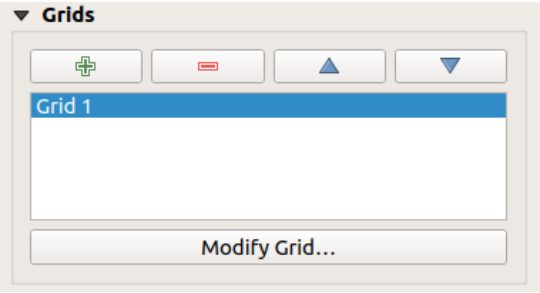


Figura18.18: Diálogo Mapear Grades

Para modificar uma grade, selecione-a e pressione o botão *Modificar grade...* para abrir o painel *Propriedades da Grade do Mapa* e acessar suas opções de configuração.

Aparência da grade

No painel *Propriedades da Grade do Mapa*, marque **|caixa de seleção|** *Grade ativada* para mostrar a grade no item do mapa.

Como tipo de grade, você pode especificar o uso de:

- **Solid**: shows a line across the grid frame. The *Line style* can be customized using *color* and *symbol* selector widget;
- **Cross**: displays segment at the grid lines intersection for which you can set the *Line style* and the *Cross width*;
- **Markers**: only displays customizable markers symbol at grid lines intersection;
- or **Frame and annotations only**.

Além do tipo de grade, você pode definir:

- the *CRS* of the grid: by default, it will follow the map item CRS. Press Select CRS button to set it to a different CRS.
- the *Interval* type to use for the grid references:
  - **Map Units**: you set a distance within the map (in the unit of the grid CRS) between consecutive grid references in the *X* and *Y* directions. The number of grid ticks will vary depending on the map scale.
  - choosing **Fit Segment Width** will dynamically select the grid interval based on the map extent to a "pretty" interval. That optimal interval is calculated within a range of distances whose *Minimum* and *Maximum* values can be customized.
  - With **Millimeters** or **Centimeters**, you set a distance on the paper between consecutive grid references in the *X* and *Y* directions. The number of grid ticks will be the same whatever the map scale.
- the *Offset* from the map item edges, in the *X* and/or the *Y* direction
- e o *Modo de mistura* da grade (veja Modo-de-misturas) quando compatível.

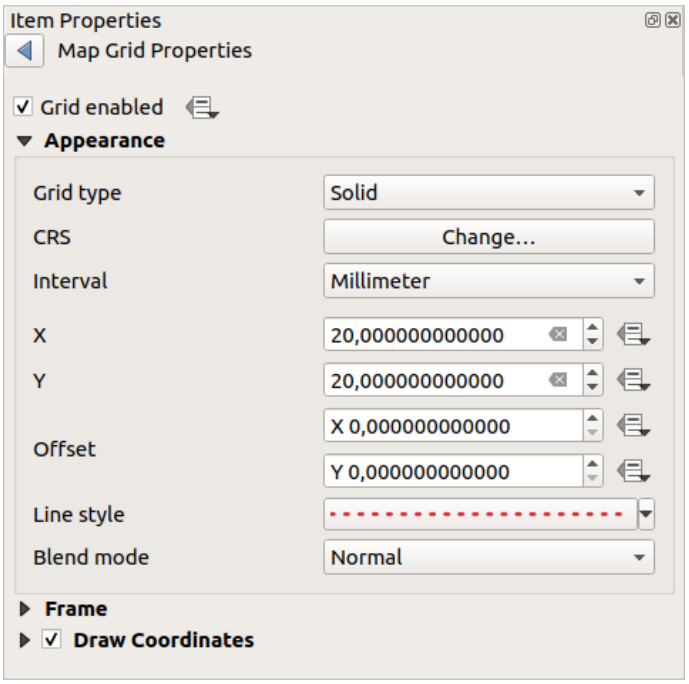


Figura18.19: Diálogo de Aparência da Grade

Quadro de grade

Existem diferentes opções para estilizar o quadro que contém o mapa. As seguintes opções estão disponíveis: `Nenhum quadro`, `Zebra`, `Zebra (náutica)`, `Ticks interiores`, `Ticks exterior`, `Interior e Exterior ticks`, `Borda da linha` e ``Borda da linha (náutica)`.

Quando compatível, é possível definir o *Tamanho da moldura*, a *Margem da moldura*, a *Espessura da linha da moldura* com a cor associada e a *Cores de preenchimento da moldura*.

Usando os valores `Latitude/Y apenas` e `Longitude/X apenas` na seção de divisões, você pode evitar uma combinação de coordenadas de latitude/Y e longitude/X exibidas em cada lado ao trabalhar com mapas girados ou grades reprojetadas. Além disso, você pode optar por definir visível ou não cada lado do quadro de grade.

When the map item extent is rotated (from the *Main properties* group) or the grid has a different CRS applied, grid lines may not be orthogonal to the map item sides. This can result in bad looking of the grid when styled with interior and/or exterior ticks. Checking *Follow grid rotation* will align the ticks with grid lines. Moreover, you can adjust some more properties:

- *Ticks alignment*: The interior and/or exterior ticks will be parallel to their corresponding grid line. Their alignment can be:
  - **Orthogonal**: ticks on the same side end at one line, parallel to the side. This can result e.g. in some ticks getting longer when with a low angle to the frame.
  - **Fixed length**: all ticks have the same length, so they may not align
- *Skip below angle*: prevents displaying ticks for grid lines intersecting the frame border below a specified threshold
- *Margin from map corner*: prevents displaying ticks too close to the map corners, because they could overlap and/or be out of bounds.

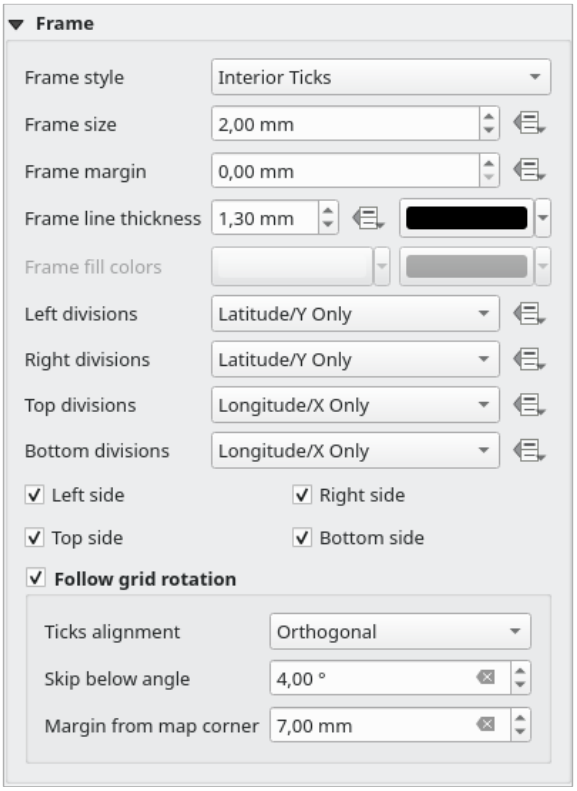


Figura18.20: Caixa de Diálogo de Quadro de Grade

Coordenadas

The *Draw coordinates* checkbox allows you to add coordinates to the map frame. Displayed values relate to the chosen *grid interval* unit. You can choose the annotation numeric format, the options range from decimal to degrees, minute and seconds, with or without suffix, aligned or not and a custom format using the expression dialog.

For each of the *Left*, *Right*, *Top* and *Bottom* sides of the grid frame, you can indicate:

- whether to render the coordinates: **Show all**, **Show latitude/Y only**, **Show longitude/X only**, **Disabled**. Showing only Latitude/Y or Longitude/X values in the divisions helps prevent a mix of latitude/Y and longitude/X coordinates showing on each side when working with rotated maps or reprojected grids.
- the relative position of the text to the grid frame: **Outside frame** or **Inside frame**
- the placement and orientation of the annotation:
  - **Horizontal**
  - **Vertical ascending**, **Vertical descending**
  - **Boundary direction**
  - **Above tick**, **On tick**, **Under tick** when a tick-based frame is used

You can also define the *Font properties* (font, size, color, buffer,…) the *Distance to the map frame* and the *Coordinate precision* (number of decimals) for the drawn annotations.

Follow grid rotation: available when the map extent is rotated or the grid is reprojected, it helps you adjust the annotations placement. Depending on the selected placement mode, the annotations are also rotated:

- *Annotations alignment*: it can be **Orthogonal** or of **Fixed length**
- *Skip below angle*: prevents displaying annotations for grid lines intersecting the frame border below a specified threshold
- *Margin from map corner*: prevents displaying annotations too close to the map corners, because they could overlap and/or be out of bounds.

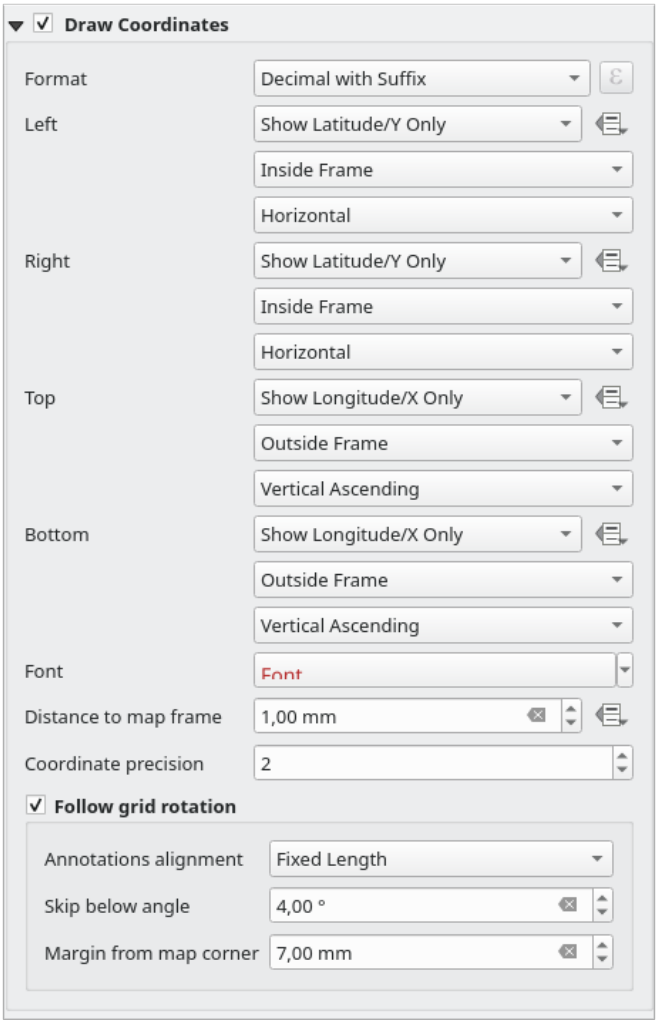


Figura18.21: Caixa de diálogo Coordenadas de desenho de grade

Visões gerais

Às vezes você pode ter mais de um mapa no layout de impressão e gostaria de localizar a área de estudo de um item do mapa em outro. Isso pode ser, por exemplo, para ajudar os leitores do mapa a identificar a área em relação ao seu contexto geográfico maior mostrado no segundo mapa.

O grupo *Visões gerais* do painel do mapa ajuda você a criar o link entre duas extensões de mapas diferentes e fornece as seguintes funcionalidades:

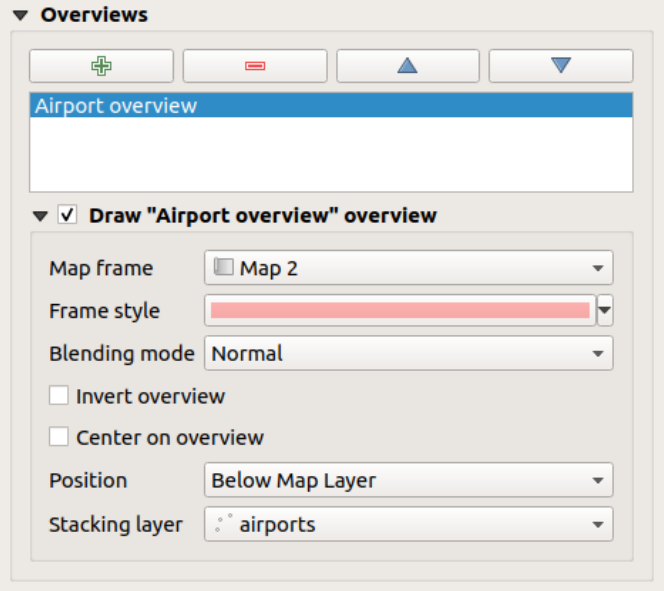


Figura18.22: Grupo de visões gerais do mapa

Para criar uma visão geral, selecione o item do mapa no qual deseja mostrar a extensão do outro item do mapa e expanda a opção *Visão geral* no painel *Propriedades do item*. Em seguida, pressione a **|Adicionar simbologia|** botão para adicionar uma visão geral.

Inicialmente esta visão geral é chamada de 'Visão Geral 1' (veja `Figura_layout_mapa_visão geral`). Você pode:

- Renomeie-o com um clique duplo
- Com a **|simbologia Adicionar|** e **|simbologia Remover|** botões, adicionar ou remover visões gerais
- Com a **|seta para cima|** e **|seta para baixo|** botões, move uma visão geral para cima e para baixo na lista, colocando-a acima ou abaixo de outras visões gerais no item do mapa (quando estão na mesma posição de pilha 1).

Em seguida, selecione o item de visão geral na lista e marque a **|caixa de seleção|** *Desenhe "1" visão geral* para permitir o desenho da visão geral no quadro de mapa selecionado. Você pode personalizá-lo com:

- O :guilabel:` Moldura do mapa ` seleciona o item do mapa cujas extensões serão mostradas no item do mapa atual.
- O *Estilo de moldura* usa as propriedades do símbolo 1 para renderizar o quadro de visão geral.
- O *Modo de mesclagem* permite definir diferentes modos de mesclagem de transparência.
- A **|caixa de seleção|** *Inverter visão geral* cria uma máscara ao redor das extensões quando ativado: as extensões do mapa referenciadas são mostradas claramente, enquanto o resto do item do mapa é misturado com a cor de preenchimento do quadro (se uma cor de preenchimento for usada).
- A **|caixa de seleção|** *Centro na visão geral* move o conteúdo do item do mapa para que o quadro de visão geral seja exibido no centro do mapa. Você só pode usar um item de síntese para centralizar, quando houver várias sínteses.
- O *Posição* controla exatamente onde na pilha de camadas do item do mapa a visão geral será colocada, por exemplo permitindo que uma extensão de visão geral seja desenhada abaixo de algumas camadas de feição, como estradas, enquanto a desenha acima de outras camadas de fundo. As opções disponíveis são:
  - *Abaixo do mapa*
  - *Abaixo da camada do mapa* e *Acima da camada do mapa*: coloque o quadro de visão geral abaixo e acima das geometrias de uma camada, respectivamente. A camada é selecionada na opção *Camada de empilhamento*.

- *Abaixo dos rótulos do mapa*: dado que os rótulos são sempre renderizados acima de todas as geometrias de feição em um item do mapa, coloca o quadro de visão geral acima de todas as geometrias e abaixo de qualquer rótulo.
- *Acima dos rótulos do mapa*: coloca o quadro de visão geral acima de todas as geometrias e rótulos no item do mapa.

18.2.3 O item de mapa 3D

O item Mapa 3D é usado para exibir uma visualização de mapa 3D `. Use o ladd Mapa 3D| :guilabel:`Adicionar Mapa 3D, e siga as instruções de criação de itens ` para adicionar um novo item de Mapa 3D que você pode manipular mais tarde da mesma forma como demonstrado em :ref:`interagir_layout_item.

Por padrão, um novo item de mapa 3D está vazio. Você pode definir as propriedades da visualização 3D e personalizá-la no painel *Propriedades do item*. Além das propriedades comuns `, este recurso possui as seguintes funcionalidades (:numref:`figura_layout_3dmapa_prop):

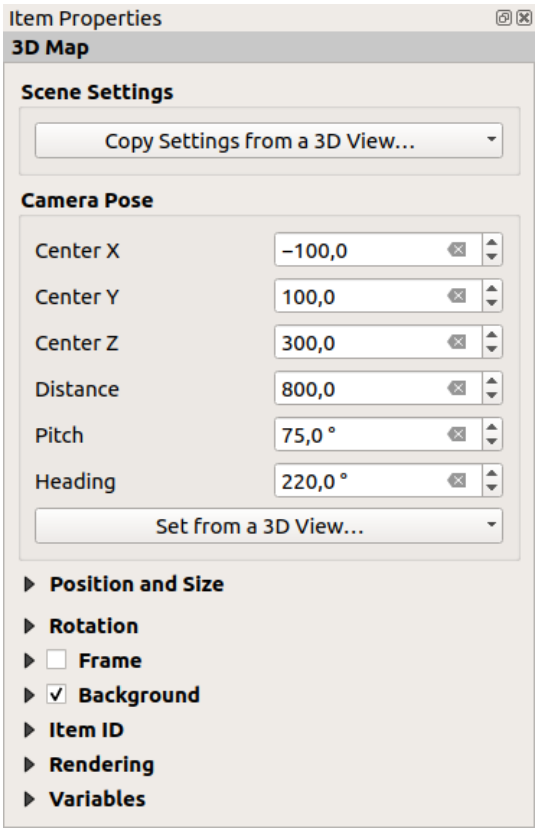


Figura18.23: Propriedades do item de mapa 3D

Configurações de cena

Pressione *Copiar configurações de uma vista 3D…* para escolher a visualização do mapa 3D a ser exibida.

A visualização do mapa 3D é renderizada com sua configuração atual (camadas, terreno, luzes, posição da câmera e ângulo…).

pose da câmera

- *Centro X* define a coordenada X do ponto para o qual a câmera está apontando
- *Centro Y* define a coordenada Y do ponto para o qual a câmera está apontando
- *Centro Z* define a coordenada Z do ponto para o qual a câmera está apontando
- *Distância* define a distância do centro da câmera até o ponto para o qual a câmera está apontando
- *Pitch* define a rotação da câmera em torno do eixo X (rotação vertical). Valores de 0 a 360 (graus). 0°: terreno visto de cima; 90°: horizontal (de lado); 180°: direto de baixo; 270°: horizontal, de cabeça para baixo; 360°: direto de cima.
- *Cabeçalho* define a rotação da câmera em torno do eixo Y (rotação horizontal - 0 a 360 graus). 0°/360°: norte; 90°: oeste; 180°: sul; 270°: leste.

O menu suspenso *Definir a partir de uma vista 3D…* permite preencher os itens com os parâmetros de uma visualização 3D.

18.2.4 A rotulagem de item

o item:guilabel:*Label 'é uma ferramenta que ajuda a decorar seu mapa com textos que ajudam a entendê-lo; pode ser o título, autor, fontes de dados ou qualquer outra informação…Você pode adicionar um rótulo com o | label | : guilabel: `Adicionar etiqueta* ferramenta a seguir: ref:` itens instruções de criação <create_layout_item>`e manipule-a da mesma maneira como exposto em: ref:` interact_layout_item

Por padrão, o item de rótulo fornece um texto padrão que você pode personalizar usando seu painel *Propriedades do item*. Além das Propriedades comuns dos itens 1, este recurso tem as seguintes funcionalidades (veja :número ref:*figura_layout_rótulo*):

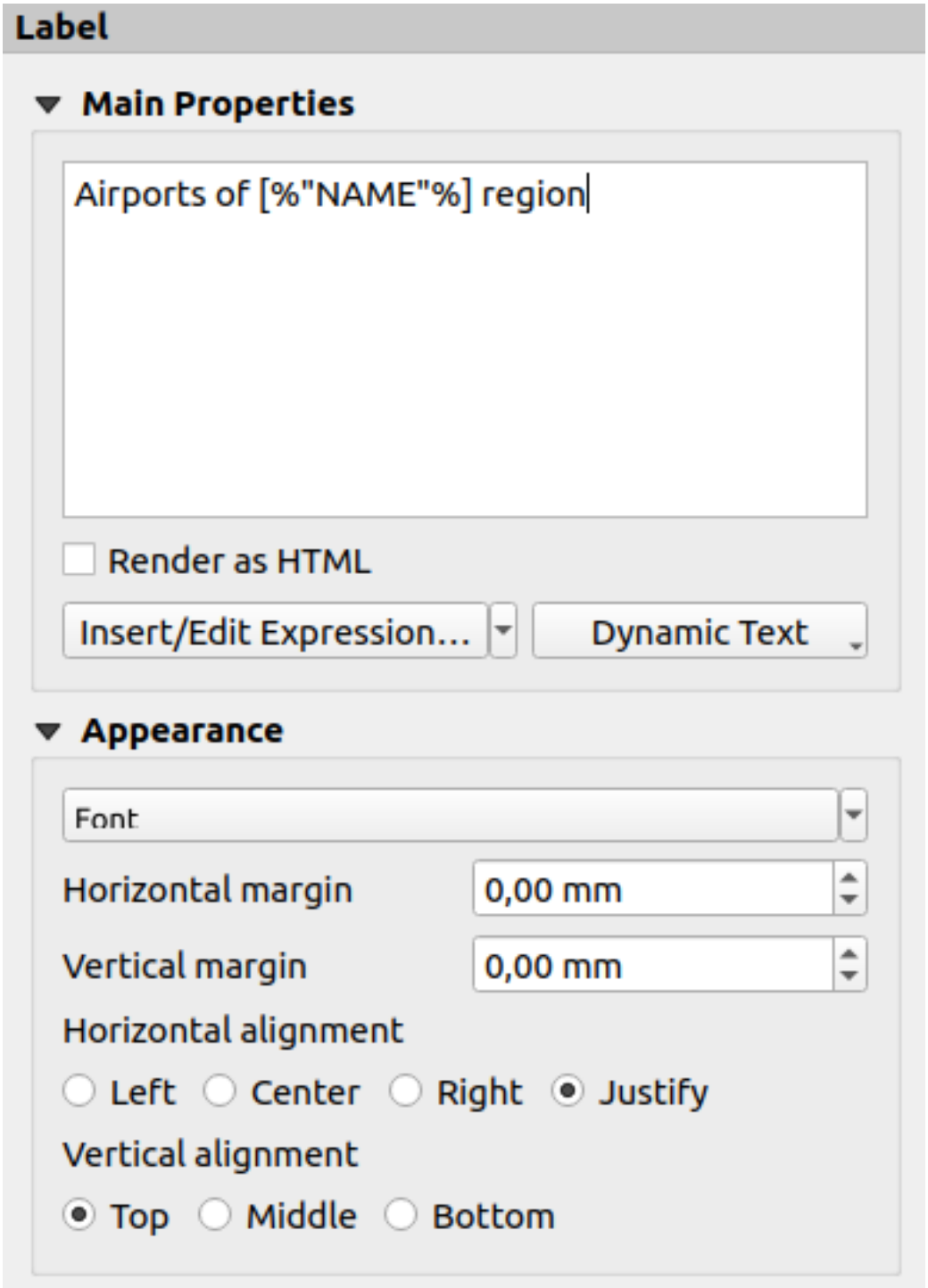


Figura18.24: Painel de propriedades do item de etiqueta

Propriedades principais

O grupo *Principais propriedades* é o local para fornecer o texto do rótulo. O texto pode ser estático, dinâmico com funções e variáveis :ref:*expressão 1*, e/ou formatado com HTML. Partes dinâmicas de um rótulo precisam ser cercadas por `[%` e `%]` para serem interpretadas e avaliadas como tal.

- Para usar expressões em rótulos, você pode clicar no botão *Inserir/Editar Expressão…*, escrever sua fórmula como de costume e quando a caixa de diálogo for aplicada, o QGIS adiciona automaticamente os caracteres ao redor.

  > **Dica**
  >
  > When you click the *Insert/Edit Expression…* button with no selection made in the textbox, the new expression will be appended to the existing text. However, if you want to modify an existing expression, the behavior changes based on your selection: If some text is selected and the selection is within an expression (between '[%' and '%]'), or if no text is selected but the cursor is inside an expression, the whole expression will be selected. In all other cases, the selection will remain as is.

  Como os mapas geralmente são preenchidos com algumas informações textuais comuns (data, autor, título, número da página, …), o QGIS fornece acesso direto às expressões ou variáveis correspondentes: pressione o botão *Texto dinâmico* para selecionar e insira-os em sua etiqueta.

  > **Dica**
  >
  > O menu superior: seleção de menu: *Adicionar Item –> Adicionar texto dinâmico –>* pode ser usado para criar um novo item de etiqueta preenchido com a expressão predefinida selecionada.

  É possível transformar um rótulo dinâmico em estático: pressione a seta suspensa ao lado do botão *Inserir/Editar Expressão…* e selecione *Converter para estático*. Quaisquer partes dinâmicas do conteúdo do rótulo serão avaliadas e substituídas por seus valores atuais. Você pode ajustar manualmente o texto resultante quando necessário.

- As etiquetas podem ser interpretadas como código HTML: verificar **|caixa de seleção|** *Renderizar como HTML*. Agora você pode inserir marcações ou estilos HTML, URL, uma imagem clicável com link para uma página da web ou algo mais complexo…

O código a seguir combina renderização HTML com expressões, para uma rotulagem avançada e produzirá :número ref:*figura_layout_rótulo_html*:

```
<html>
 <head>
   <style>
      /* Define some custom styles, with attribute-based size */
      name {color:red; font-size: [% ID %]px; font-family: Verdana; text-shadow:␣
↪grey 1px 0 10px;}
      use {color:blue;}
   </style>
 </head>

 <body>
   <!-- Information to display -->
   <u>Feature Information</u>
   <ul style="list-style-type:disc">
     <li>Feature Id: [% ID %]</li>
     <li>Airport: <name>[% NAME %]</name></li>
     <li>Main use: <use>[% USE %]</use></li>
   </ul>
   Last check: [% concat( format_date( "control_date", 'yyyy-MM-dd'), ' by <b><i>',
↪ @user_full_name, '</i></b>' ) %]
```

(continua na próxima página)

(continuação da página anterior)

```
  <!-- Insert an image -->
  <p align=center><img src="path/to/logos/qgis-logo-made-with-color.svg" alt=
↪"QGIS icon" style="width:80px;height:50px;"</p>
 </body>
</html>
```

Figura18.25: Aproveitando um rótulo com estilo HTML

Aparência

- Define font and style of the text by clicking on the *Font* button. In the *Label Font* menu you can use some of the options for *Formatting the label text*.
- Você pode especificar diferentes margens horizontais e verticais em `mm`. Esta é a margem da borda do item de layout. O rótulo pode ser posicionado fora dos limites do rótulo, por exemplo. para alinhar itens de etiqueta com outros itens. Nesse caso, você deve usar valores negativos para a margem.
- Usar o alinhamento de texto é outra maneira de posicionar seu rótulo. Pode ser:
  - *Esquerda*, *Centro*, *Direita* ou *Justificar* para *Alinhamento horizontal*
  - and *Top*, *Vertical Center*, *Bottom* for *Vertical alignment*.

Dica

Resize the layout label item to fit its contents

By default, the size of a layout label item does not depend on the contained text. It may be too large with useless blank space or too small, hiding part of the text. To automatically resize a label so it fits the text inside it perfectly, double-click one of the square handles on the label. The label will adjust its size to match the text, either expanding or shrinking as needed, while the opposite handle of the label remains fixed in place. This allows you to quickly ensure the label fits the text without manually resizing it. So e.g., double-clicking the left-middle handle causes the left side of the label to move.

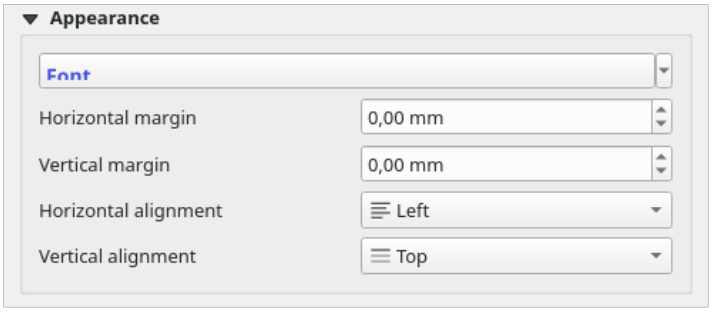


Figura18.26: Label Item Appearance Properties

Explorando expressões em um item de etiqueta

Abaixo alguns exemplos de expressões que você pode usar para preencher o item do rótulo com informações interessantes - lembre-se que o código, ou pelo menos a parte calculada, deve estar entre `[%` e `%]` no :guilabel :*Quadro das propriedades principais*:

- Exiba um título com o valor atual do recurso do atlas em “campo1”:

```
'This is the map for ' || "field1"
```

  ou, escrito na seção *Principais propriedades*:

```
This is the map for [% "field1" %]
```

- Adicione uma paginação para recursos de atlas processados (por exemplo, `` Página 1/10 “)

```
concat( 'Page ', @atlas_featurenumber, '/', @atlas_totalfeatures )
```

- Retorne o nome dos aeroportos do recurso atual da região do atlas, com base em seus atributos comuns:

```
aggregate( layer := 'airports',
           aggregate := 'concatenate',
           expression := "NAME",
           filter := fk_regionId = attribute( @atlas_feature, 'ID' ),
           concatenator := ', '
         )
```

  Ou, se uma relação de atributos 1 for definida:

```
relation_aggregate( relation := 'airports_in_region_relation',
                    aggregate := 'concatenate',
                    expression := "NAME",
                    concatenator := ', '
                  )
```

- Retorne o nome dos aeroportos contidos na feição atual da região do atlas, com base em sua relação espacial:

```
aggregate( layer := 'airports',
           aggregate := 'concatenate',
           expression := "NAME",
           filter := contains( geometry( @parent ), $geometry ),
           concatenator := ', '
         )
```

  OU:

```
array_to_string( array:= overlay_contains( layer := 'airports',
                                           expression := "NAME" ),
                 delimiter:= ', '
               )
```

- Retorne a coordenada X inferior da extensão do item `Mapa 1`:

```
x_min( map_get( item_variables( 'Map 1' ), 'map_extent' ) )
```

- Recupere o nome das camadas no item `Mapa 1` do layout atual e formate em um nome por linha:

```
array_to_string(
 array_foreach(
  map_get( item_variables( 'Map 1' ), 'map_layers' ), -- retrieve the layers␣
↪list
  layer_property( @element, 'name' ) -- retrieve each layer name
```

(continua na próxima página)

(continuação da página anterior)

```
 ),
 '\n' -- converts the list to string separated by breaklines
)
```

- Exiba a lista de camadas com seus strings de licença (direitos de uso) em um item de layout `Mapa 1`. Você precisa preencher as propriedades Metadados de acesso 1 das camadas primeiro.

```
array_to_string( map_credits( 'Map 1', true ) )
```

18.2.5 O Item Legenda

O item *Legenda' é uma caixa ou tabela que explica o significado os símbolos usados no mapa. A legenda está então ligada a um item de mapa. Você pode adicionar um item de legenda com a ferramenta \addLegend\ :guilabel:`Adicionar legenda* seguindo as *instruções de criação de items* e manipular ela da mesma maneira que exposto em *Interagindo com itens de layout*.

Por padrão, o item de legenda mostra todas as camadas disponíveis e pode ser refinada usando seu painel de *Propriedades do item*. Diferente das *propriedades comuns de itens*, está ferramenta possui as seguintes funcionalidades (veja Fig.18.27):

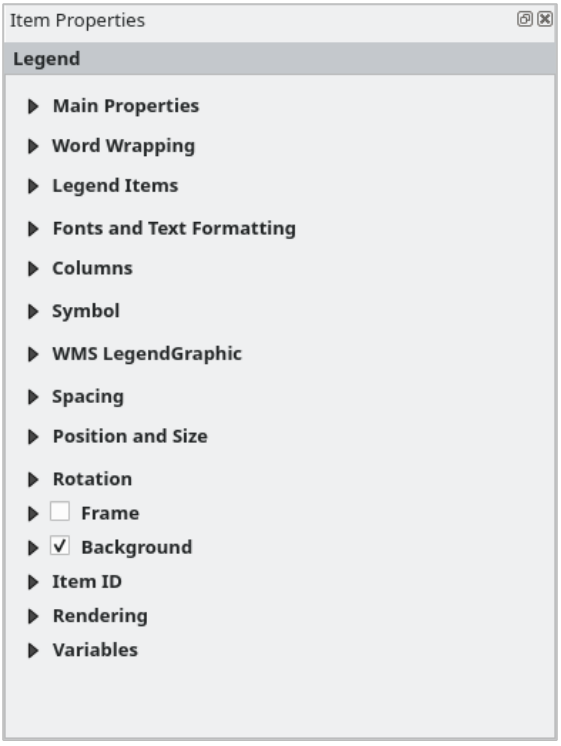


Figura18.27: Painel de Propriedades do Item Legenda

Propriedades principais

O grupo:guilabel:*Propriedades principais* do painel lengenda:guilabel:*Propriedades do item* fornece as seguintes funcionalidades (veja: número ref:*figura_layout_legenda_ppt*):

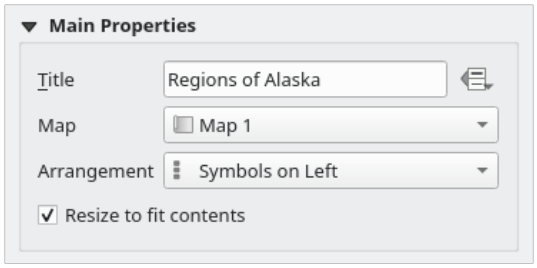


Figura18.28: Grupo de propriedades principais da legenda

Em Propriedades principais você pode:

- Altere o:guilabel:*Título* da legenda. Ele pode ser tornado dinâmico usando a configuração :ref:`substituição definida por dados 1 `, útil por exemplo ao gerar um atlas;
- Escolha a qual item:guilabel:*Mapa* a legenda atual irá se referir. Por padrão, o mapa sobre o qual o item de legenda é desenhado é selecionado. Se nenhum, então volta para o mapa de referência 1.

> **Nota**
>
> Variáveis do item de mapa vinculado (@mapa_id, @mapa_escala, @map_extemder…) também são acessíveis a partir de propriedades definidas por dados da legenda.

- Defina os símbolos e o posicionamento do texto na legenda: o *Arranjo* pode ser *Símbolos à esquerda* ou *Símbolos à direita*. O valor padrão depende da localidade em uso (da direita para a esquerda ou não).
- Use |caixa de seleção| *Redimensionar para ajustar o conteúdo* para controlar se uma legenda deve ou não ser redimensionada automaticamente para caber em seu conteúdo. Se desmarcada, a legenda nunca será redimensionada e, em vez disso, ficará no tamanho que o usuário tiver definido. Qualquer conteúdo que não se ajuste ao tamanho é cortado.

Word wrapping

The *Word wrapping* group of the legend *Item Properties* panel provides the following functionalities:

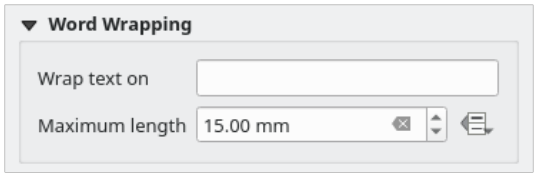


Figura18.29: Legend's word wrapping group

- Envolva o texto da legenda em um determinado caractere: cada vez que o caractere aparece, ele é substituído por uma quebra de linha;
- Allow legend text to be automatically wrapped after a set line length (in millimeters), preventing very wide auto-generated legends. The *Maximum length* can be set as a static or data-defined value, convenient for dynamic layouts which adjust legend size based on for example page orientation or displayed features, or atlases where you may want to tweak the legend appearance on different pages.

Examples:

- Control legend width based on layout page orientation

```
IF( @layout_pageheight >= @layout_pagewidth, 35, 80 )
```

- Set the legend column width when a specific layer is displayed in the linked map

```
CASE WHEN
  array_contains(
    map_get( item_variables('legend_1'), 'map_layer_ids' ),
    'a_specific_layer_id' )
THEN 60
END
```

Itens da legenda

O grupo:guilabel:*Itens de legenda* do painel legenga *Propriedades do item* fornece as seguintes funcionalidades (veja: número ref:*figura_layout_legenda_itens*):

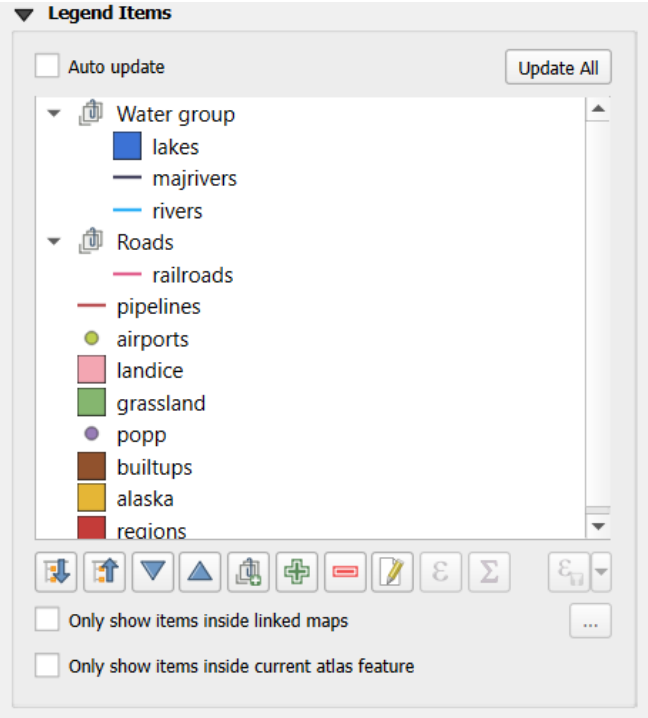


Figura18.30: Grupo de itens da legenda

- A legenda será atualizada automaticamente se |caixa de seleção| *Atualização automática* está marcada. Quando *Atualização automática* está desmarcado, isso lhe dará mais controle sobre os itens da legenda. Todos os ícones abaixo da lista de itens da legenda serão ativados.
- A janela de itens de legenda lista todos os itens de legenda e permite alterar a ordem dos itens, agrupar camadas, remover e restaurar itens na lista, editar nomes de camadas e simbologia e adicionar um filtro.
  - Use the Expand all and Collapse all buttons to respectively expand or collapse all the groups and subgroups in the legend tree. Ensure that the *Auto update* is unchecked to use these buttons.
  - A ordem do item pode ser alterada usando o |seta para cima| e |seta para baixo| botões ou com a funcionalidade 'arrastar e soltar'. A ordem não pode ser alterada para gráficos de legenda WMS.
  - Use o |adicionar grupo| botão para adicionar um grupo de legendas.

- Use o |Adicionar simbologia| botão para adicionar camadas e |Remover simbologia| botão para remover grupos, camadas ou classes de símbolos.
- O botão |Editar simbologia| é usado para editar a camada, nome do grupo ou título. Primeiro você precisa selecionar o item da legenda. Clicar duas vezes no item também abre a caixa de texto para renomeá-lo.
- A |expressão| botão usa expressões para personalizar cada rótulo de símbolo da camada selecionada (veja legenda_itens_dados_definidos)
- O botão |soma| adiciona uma contagem de recursos para cada classe de camada vetorial.
- O |Filtro expressão| Filtrar legenda por expressão ajuda a filtrar quais dos itens de legenda de uma camada serão exibidos, ou seja, usando uma camada que possui diferentes itens de legenda (por exemplo, de uma simbologia baseada em regras ou categorizada), você pode especificar um expressão booleana para remover da árvore de legendas os estilos que não possuem nenhum recurso que satisfaça uma condição. Observe que as feições são mantidas e mostradas no item do mapa de layout.

Embora o comportamento padrão do item de legenda seja imitar a árvore do painel *Camadas*, exibindo os mesmos grupos, camadas e classes de simbologia, clicar com o botão direito do mouse em qualquer item oferece opções para ocultar o nome da camada ou elevá-la como um grupo ou subgrupo. Caso você tenha feito algumas alterações em uma camada, você pode revertê-las escolhendo *Redefinir para os padrões* no menu contextual da entrada da legenda.

Após alterar a simbologia na janela principal do QGIS, você pode clicar em *Atualize tudo* para adaptar as alterações no elemento de legenda do layout de impressão.

- With the *Only show items inside linked maps*, only the legend items visible in the linked map will be listed in the legend. If you have more than one map you can click on … and select other maps from your layout. This tool remains available when *Auto-update* is active.
- Ao gerar um atlas com recursos de polígono, você pode filtrar itens de legenda que estão fora do recurso de atlas atual. Para fazer isso, marque a |caixa de seleção| :guilabel:Opção *Mostrar apenas itens dentro do recurso atual do atlas*.

Dados-defina os rótulos das legendas

|expressão| permite adicionar expressões 1 a cada rótulo de símbolo de uma determinada camada. Novas variáveis (`@rsimbolo_rótulo`, `@simbolo_id` e ``@simbolo_contar`) ajudam você a interagir com a entrada da legenda.

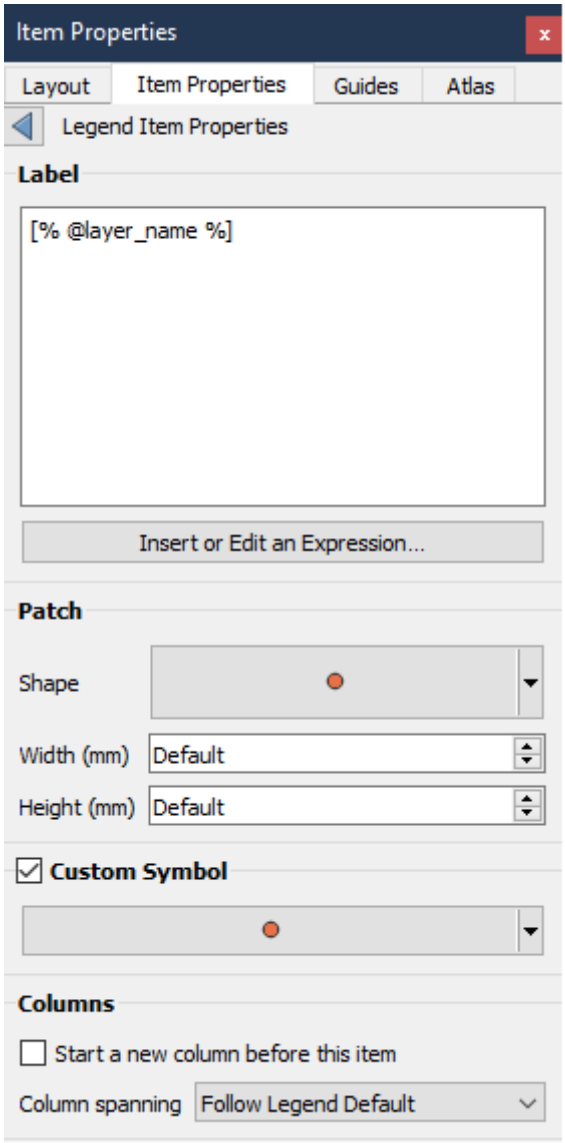
Por exemplo, dada uma camada `regiões` categorizada por seu campo `tipo`, você pode anexar a cada classe na legenda seu número de feições e área total, por exemplo. `Bairro (3) - 850ha`:

1. Selecione a entrada da camada na árvore de legendas
2. Pressione a |expressão| botão, abrindo a caixa de diálogo *Construção de Fragmentos de Expressões*
3. Digite a seguinte expressão (*supondo que os rótulos dos símbolos não tenham sido editados*):

```
format( '%1 (%2) - %3ha',
        @symbol_label,
        @symbol_count,
        round( aggregate(@layer, 'sum', $area, filter:= "type"=@symbol_label)/
↪10000 )
      )
```

4. Pressione *OK*

Personalizando itens de legenda



Itens de legenda também podem ser personalizados individualmente em *Propriedades de Itens de Legenda*. Mas essas customizações só podem ser feitas com **|caixa de seleção|** *Atualização automática* desabilitada.

Clicando duas vezes em um item ou pressionando **|Editar simbologia|** Editar propriedades do item selecionado permite maior personalização.

Rótulo

Para todos os tipos de itens, permite modificar o texto da etiqueta digitando ou inserindo expressões usando a **|expressão|** *Inserir ou editar uma expressão*. As expressões também podem ser adicionadas diretamente em qualquer lugar no rótulo do item usando a notação [% expressão %].

Colunas

A propriedade Legenda do item também permite que você controle o comportamento da divisão da coluna, forçando a divisão da coluna a ocorrer após um item específico ou todos os símbolos de uma camada. A divisão automática de uma camada e seu filho também pode ser permitida ou bloqueada por camada neste ferramenta.

Correção

Para itens com um símbolo, a propriedade Item de Legenda permite especificar a altura e a largura máximas que um símbolo pode ocupar.

Para símbolos vetoriais, você pode especificar uma forma personalizada para o símbolo. As formas geralmente são definidas por uma expressão para representar a geometria em um plano simples, mas esses símbolos também podem ser salvos no gerenciador de estilos e importados posteriormente. O símbolo padrão para cada tipo de geometria também pode ser controlado através do gerenciador de estilos.

Símbolo Personalizado

Um símbolo personalizado também pode ser especificado para símbolos vetoriais. Isso pode ser útil para ajustar a renderização de um símbolo específico, aprimorá-lo na legenda ou ter um símbolo independente de sua visualização de símbolo real. Este símbolo personalizado substituirá o símbolo da legenda, mas levará em conta o símbolo *Correção* especificado.

Fonts and text formatting

The *Fonts and text formatting* group of the legend *Item Properties* panel provides the following functionalities:

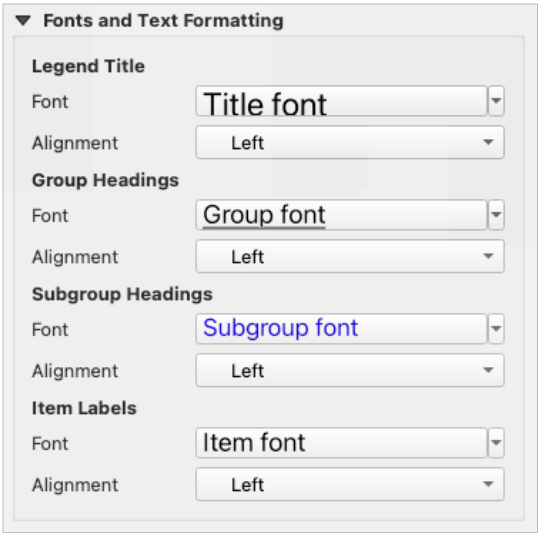


Figura18.31: Propriedades das fontes da legenda

- You can change the font of the legend title, group, subgroup and item (feature) in the legend item using the *font selector* widget, providing all the capabilities of *text formatting* (font spacing control, mixed HTML formatting, coloring, blending, background, text buffer, shadow, …)
- Para cada um desses níveis você pode definir o texto *Alinhamento*: pode ser *Esquerda* (padrão para localidades baseadas da esquerda para a direita), *Centro* ou *Direita* (padrão para localidades baseadas da direita para a esquerda).

Colunas

Sob o grupo *Colunas* do painel legenda *Propriedades do item*, os itens da legenda podem ser organizados em várias colunas:

- Defina o número de colunas no *Contar* **|selecione o número|** campo. Este valor pode ser tornado dinâmico, por exemplo, seguindo as características do atlas, o conteúdo da legenda, o tamanho do quadro…
- **|caixa de seleção|** *Larguras de coluna iguais* define como as colunas da legenda devem ser ajustadas.
- A **|caixa de seleção|** A opção *Dividir camadas* permite que uma legenda de camada categorizada ou graduada seja dividida entre colunas.

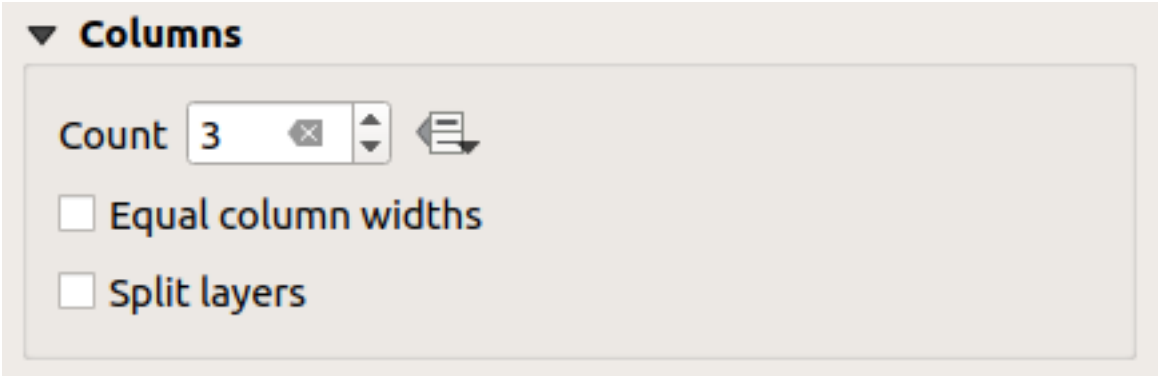


Figura18.32: Configurações das colunas da legenda

Símbolo

O grupo *Símbolo* do painel *Propriedades do item* da legenda configura o tamanho dos símbolos exibidos ao lado dos rótulos da legenda. Você pode:

- Defina *Largura do símbolo* e *Altura do símbolo*
- Defina os marcadores' *Tamanho mínimo do símbolo* e *Tamanho máximo do símbolo*: `0.00mm` significa que não há nenhum valor definido.
- |caixa de seleção| *Desenhar traço para símbolos raster*: adiciona um contorno ao símbolo representando a cor da faixa da camada raster; você pode definir *Cor do traço* e *Espessura*.

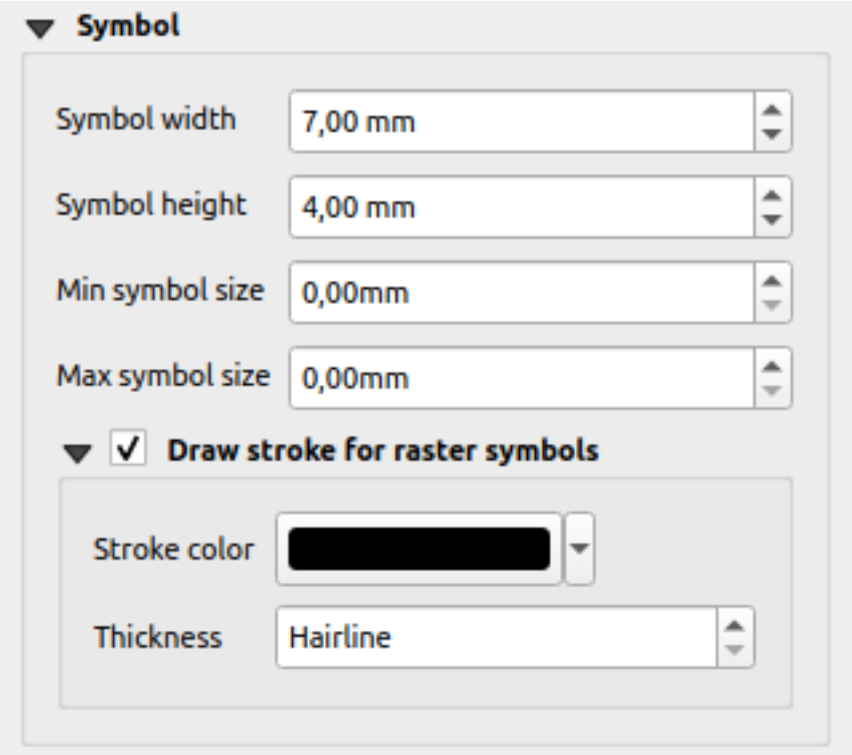


Figura18.33: Configuração dos símbolos da legenda

Gráfico de legenda do WMS

A seção *WMS Gráfico de legenda* do painel legenda *Propriedades do Item* fornece as seguintes funcionalidades (veja :número ref:*figura_layout_legenda_wms*):

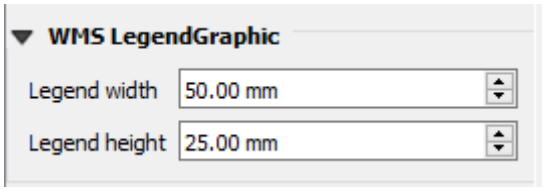
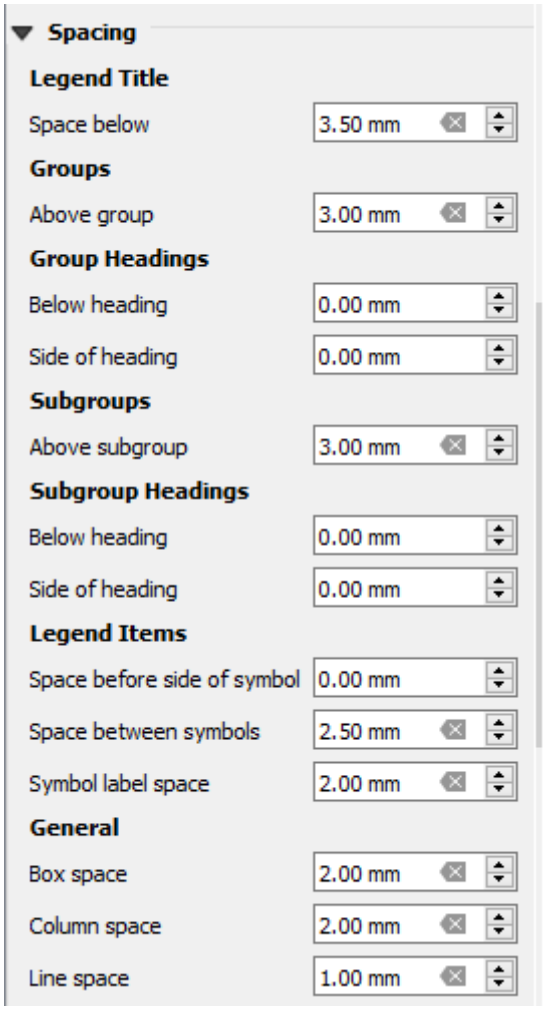


Figura18.34: Gráfico de legenda do WMS

Quando você adiciona uma camada WMS e insere um item de legenda, uma solicitação será enviada ao servidor WMS para fornecer uma legenda WMS. Esta legenda só será mostrada se o servidor WMS fornecer o recurso Obter gráfico de legenda. O conteúdo da legenda do WMS será fornecido como uma imagem raster.

WMS Gráfico de legenda é usado para ajustar a *Largura da legenda* e a *Altura da legenda* da imagem raster da legenda do WMS.

Espaçamento



A seção *Espaçamento* permite que você personalize o espaçamento dentro da legenda. O espaçamento pode ajudar muito a denotar o agrupamento de itens na legenda e sua relação.

Espaçamento ao redor e antes do título, grupos, subgrupos, símbolos, rótulos, caixas, colunas e linhas podem ser personalizados através deste diálogo.

18.2.6 Item da Barra de Escala

As barras de escala fornecem uma indicação visual do tamanho das feições e da distância entre as feições no item do mapa. Um item de barra de escala requer um item de mapa. Use a **|Barra de escala|** *Adicionar Barra de Escala* seguindo as instruções de criação de itens 1 ` para adicionar um novo item da barra de escala que você pode manipular mais tarde da mesma forma como exposto em :ref:`interagir_layout_item.

Por padrão, um novo item da barra de escala mostra a escala do item do mapa sobre o qual é desenhado. Se não houver nenhum item de mapa abaixo, o mapa de referência 1 ` é usado. Você pode personalizá-lo no painel :guilabel:`Propriedades do Item. Além das propriedades comuns dos itens 2, este recurso tem as seguintes funcionalidades (veja: número ref:*figura_layout_barra_de_escala*):

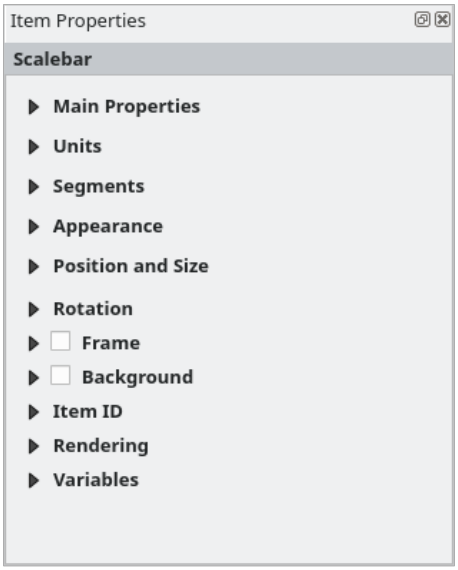


Figura18.35: Painel de Propriedades do Item da Barra de Escala

Propriedades principais

O grupo:guilabel:*Principais propriedades* do painel da barra de escala: guilabel: *Propriedades do item* fornece as seguintes funcionalidades (veja: número ref:*figura_layout_barra_de_escala_ppt*):

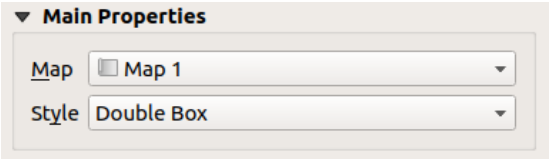


Figura18.36: Grupo de propriedades principal da barra de escala

1. Primeiro, escolha o mapa ao qual a barra de escala será anexada
2. Em seguida, escolha o estilo da barra de escala. Os estilos disponíveis são:
   - Estilos **Caixa única** e **Caixa dupla**, que contêm uma ou duas linhas de caixas alternando cores;

- Marcações de linha **Médio**, **Para cima** ou **Para baixo**;
- Estilo **Linha escalonada** que desenha uma representação de linha escalonada de uma barra de escala
- Estilo **oca** que desenha uma única caixa com cores alternadas para os segmentos, com linhas horizontais nos segmentos alternados
- **Numérico**, onde a proporção de escala é impressa (por exemplo, `1:50000`).

3. Defina as propriedades conforme apropriado

Unidades

O grupo:guilabel:*Unidades* da barra de escala: guilabel:*Item Properties* fornece as funcionalidades para definir as unidades de exibição e alguma formatação de texto (veja: número ref:*figura_layout_barra_de_escala_unidades*):

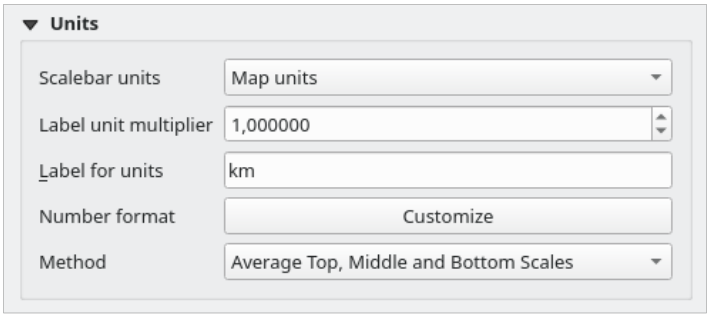


Figura18.37: Grupo de unidades da barra de escala

- Selecione as unidades que deseja usar com:guilabel:*Unidades_barra_de_escala*. Há muitas opções possíveis: **Unidades de mapa** (a padrão), **Metros**, **Pés**, **Milhas** ou **Milhas náuticas**… e alguns derivados. A conversão de unidades é feita automaticamente.
- O: guilabel:*Multiplicador de unidade de rótulo* especifica quantas unidades de barra de escala por unidade rotulada. Por exemplo, se as unidades da barra de escala estiverem definidas para “metros”, um multiplicador de 1000 resultará nos rótulos da barra de escala em “quilômetros”.
- O campo:guilabel:*Etiqueta para unidades* define o texto usado para descrever as unidades da barra de escala, por exemplo, `m` ou `km`. Isso deve ser combinado para refletir o multiplicador acima.
- Pressione *Customizar* ao lado de *Formato numérico* para ter controle sobre todas as propriedades de formatação dos números na barra de escala, incluindo separadores de milhar, casas decimais, notação científica, etc. (veja :ref:` formatação_número` para mais detalhes). Muito útil no caso de fazer mapas para audiências fora do local atual do QGIS, ou quando você deseja variar o estilo dos padrões do local (por exemplo, adicionar separadores de milhares quando o padrão do local é ocultá-los).
- Select the *Method* for scale bar calculation: Depending on the map CRS and extent, the distance reported in the scale bar may not be the same measured over the map. This option, which defaults to the *project's scale calculation method*, helps you specify the one you would like to use for this particular scale bar.

Segmentos

O grupo *Segmentos* da barra de escala *Propriedades do item* fornece as funcionalidades para configurar o número e tamanho dos segmentos e subdivisões (veja :número ref:*figura_layout_barra_de_escala_segmentos*):

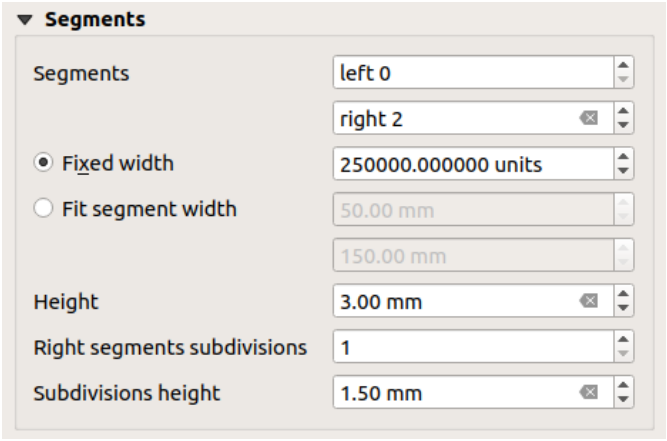


Figura18.38: Grupo de segmentos da barra de escala

- Você pode definir o número de *Segmentos* que serão desenhados nos lados esquerdo e direito do `0` da barra de escala:
  - número de subdivisões de um segmento único no lado:guilabel:*Esquerdo*
  - número de segmentos no lado:guilabel:*direita*
- You can set the width of a segment or a range for the total length of the scale bar:
  - Set how long a segment will be in your scale bar units (*Fixed width*)
  - Or limit the total scale bar size in `mm` with *Fit segment width* option. In the latter case, each time the map scale changes, the scale bar is resized (and its label updated) to fit between the upper and lower range set.
- *Altura* é usado para definir a altura da barra.
- *Subdivisões do segmento direito* é usado para definir o número de seções que os segmentos do lado direito da barra de escala podem ter (para estilos de barra de escala *Marcas de linha para baixo*, *Marcas de linha no meio* e *Linhas para cima*) .
- *Altura da subdivisão* é usado para definir a altura do segmento de subdivisão.

Aparência

The *Appearance* group of the scale bar *Item Properties* panel provides the following functionalities:

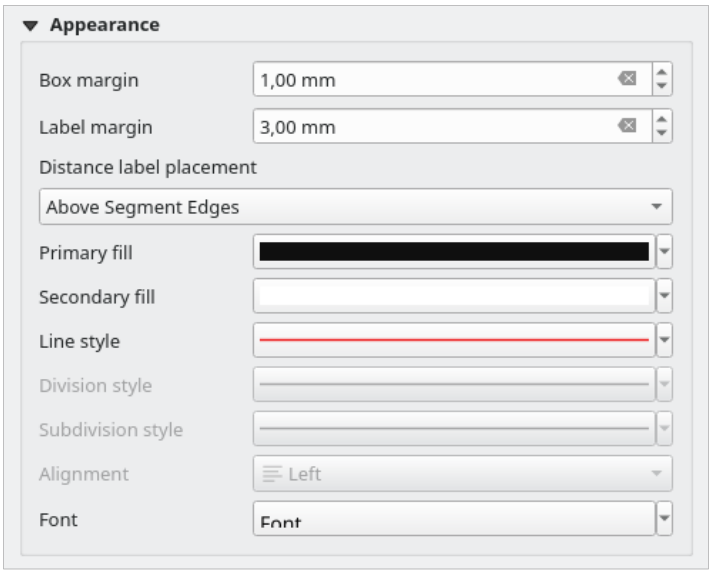


Figura18.39: Scale Bar Appearance group

Você pode definir como a barra de escala será exibida em seu quadro.

- *Margem da caixa* : espaço entre o texto e as bordas do quadro
- *Margem da etiqueta* : espaço entre o texto e o desenho da barra de escala
- *Distance label placement* defines the position of the text relative to the scale bar segments: *Above Segment Edges*, *Above Segment Centers*, *Below Segment Edges* or *Below Segment Centers*
- *Preenchimento primário* e *Preenchimento secundário* do desenho da barra de escala usando preencher as propriedades dos símbolos 1 (cor, opacidade, padrões, efeitos…) — para *Caixa Simples* , estilos *Caixa dupla* e *Oca*
- *Estilo de linha* do desenho da barra de escala usando propriedades de símbolos de linha 1 (cor, traço, junção, estilo cap, padrões, efeitos…) — para todos, exceto o estilo *Numérico*
- *Estilo de divisão* e *Estilo de subdivisão* respectivamente para segmentos de divisão e subdivisão nos estilos de barra de escala *Linhas para cima*, *Linhas para o meio* e *Linha para baixo* usando as propriedades de símbolos de linha :ref:` 1` (cor, traço, junção, estilo cap, padrões, efeitos…)
- *Alinhamento* coloca o texto no lado esquerdo, central ou direito do quadro (somente para o estilo de barra de escala *Numérico*)
- Português *Font* para definir as propriedades 1 (tamanho, fonte, cor, espaçamento entre letras, sombra, fundo…) do rótulo da barra de escala.

Como a maioria das propriedades de exibição da barra de escala depende de símbolos cujas propriedades podem ser definidas por dados, é possível renderizar barras de escala definidas por dados.

Exemplo: o código a seguir aplicado à propriedade negrito dos rótulos de escala exibirá números em negrito quando forem múltiplos de 500:

```
-- returns True (or 1) if the value displayed on the bar
-- is a multiple of 500

@scale_value % 500 = 0
```

18.2.7 Os itens da tabela

Você pode usar os itens da tabela para decorar e explicar o seu mapa:

- *Attribute table*: automatically extracts a subset of the attributes of a layer, based on predefined rules
- *Fixed table*: allows for creation of tables with contents manually entered (i.e. spreadsheet style), so that you can create completely custom tables.

Item da tabela de atributos

Qualquer camada no projeto pode ter seus atributos mostrados no layout de impressão. Use a ferramenta *Adicionar Tabela de Atributos* seguindo as instruções de criação dos itens. <create_layout_item>Para adicionar um novo item tabela você pode manipular depois da mesma forma como exposto em “interagir com item de layout”

Por padrão, um novo item da tabela de atributos mostra colunas da primeira camada (listada alfabeticamente), com todos os campos. Você pode entretanto personalizar a tabela graças ao painel *Propriedades do Item*. Diferente das *propriedades comuns de itens*, esta opção tem as seguintes funções (veja Fig.18.40):

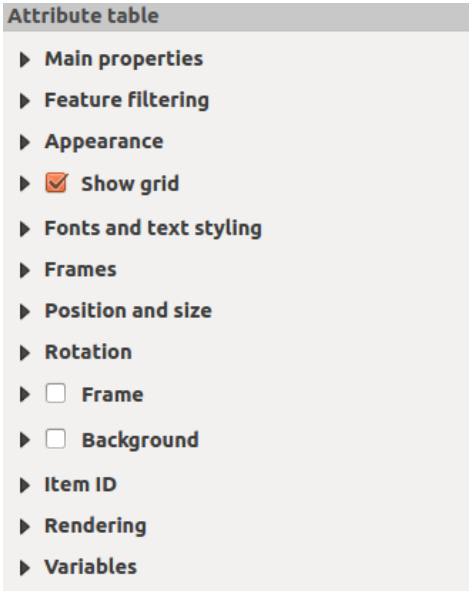


Figura18.40: Tabela de atributos no Painel de Propriedades do Item

Propriedades principais

A tabela do grupo de atributos *Main properties* oferece as seguintes funcionalidades (veja Fig.18.41):

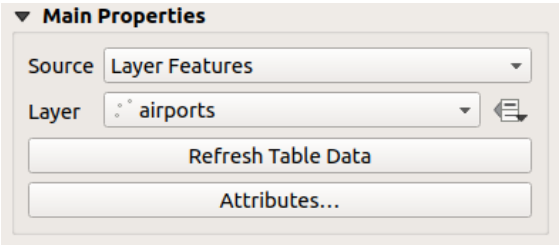


Figura18.41: Tabela de atributos Grupo de propriedades principais

- Para *Source* você pode por pardrão apenas selecionar **Feições de camada** permitido que você selecione uma *Layer* das camadas de vetores carregadas no projeto.

O |dados Definir| O botão Substituição definida por dados próximo à lista de camadas permite que você altere dinamicamente a camada que é usada para preencher a tabela, ex. você pode preencher a tabela de atributos com diferentes atributos de camada por página do atlas. Observe que a estrutura da tabela usada (:numref:`figura_layout_tabela_selecione`) é a da camada mostrada na lista suspensa *Camada* e é deixada intacta, o que significa que definir uma tabela definida por dados para uma camada com campos diferentes resultarão em colunas vazias na tabela.

Caso você ative a opção |caixa de seleção|*Gerar um atlas* no painel *Atlas* (veja atlas_geração), existem dois *Fonte* adicionais possíveis:

- **Recurso atlas atual** (veja `figura_layout_tabela_atlas`): você não verá nenhuma opção para escolher a camada, e o item da tabela mostrará apenas uma linha com os atributos do recurso atual da camada de cobertura do atlas .
- e **Relação filhos** (veja `figura_layout_tabela_relação`): uma opção com os nomes das relações aparecerá. Este recurso só pode ser usado se você tiver definido uma rellação 1 usando sua camada de cobertura do atlas como pai, e a tabela mostrará as linhas filhas do recurso atual da camada de cobertura do atlas.

- O botão *Atualizar dados da tabela* pode ser usado para atualizar a tabela quando o conteúdo real da tabela for alterado.

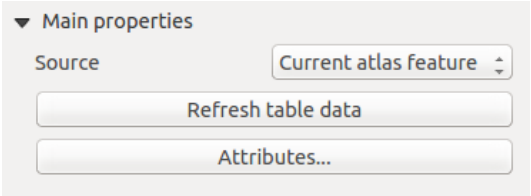


Figura18.42: Tabela de atributos Propriedades principais para 'Recurso de atlas atual'

Figura18.43: Tabela de atributos Propriedades principais para 'Relação filhos'

- O botão *Atributos…* inicia o diálogo *Selecionar atributos*, (veja `figura_layout_tabela_selecionar`) que pode ser usado para alterar o conteúdo visível da tabela. A parte superior da janela mostra a lista dos atributos a serem exibidos e a parte inferior ajuda a classificar os dados.

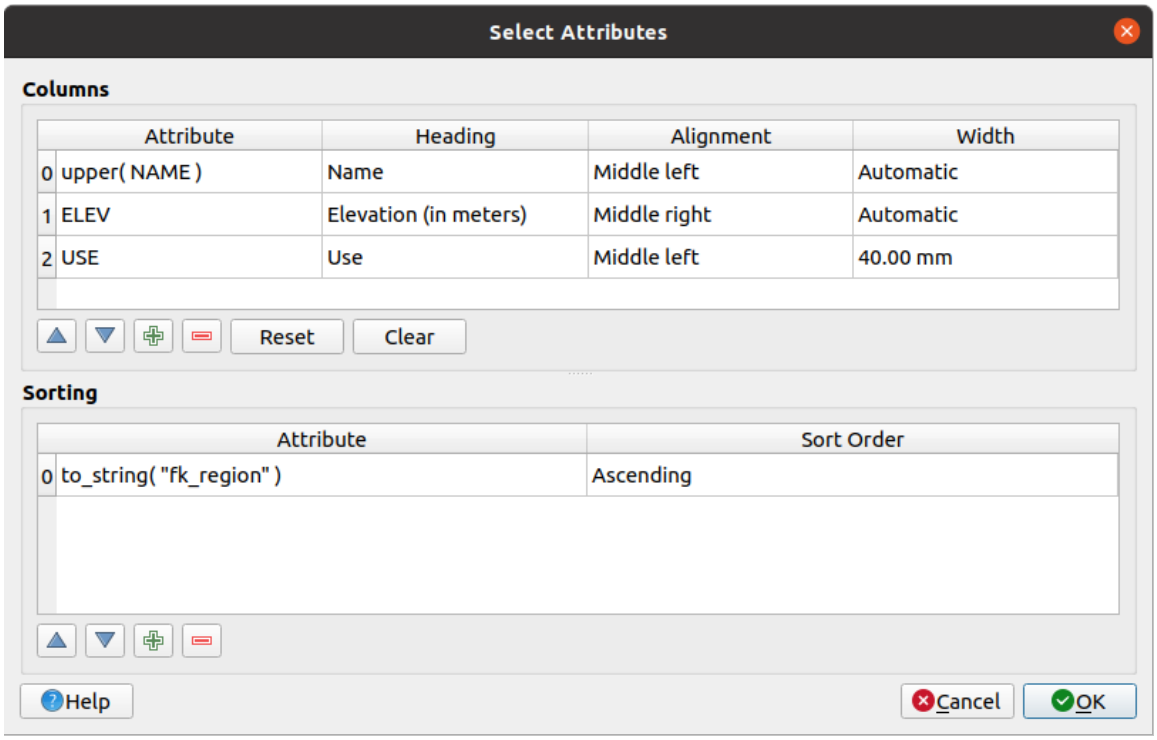


Figura18.44: Tabela de atributos Diálogo de seleção de atributos

Na seção *Colunas* você pode:

- Mova os atributos para cima ou para baixo na lista selecionando as linhas e usando a |seta para cima| e |seta para baixo| botões para deslocar as linhas. Várias linhas podem ser selecionadas e movidas a qualquer momento.
- Adicione um atributo com o |Adicionar simbologia| botão. Isso adicionará uma linha vazia na parte inferior da tabela, onde você poderá selecionar um campo para ser o valor do atributo ou criar um atributo por meio de uma expressão regular.
- Remova um atributo com o |Remover simbologia| botão. Várias linhas podem ser selecionadas e removidas a qualquer momento.
- Redefina a tabela de atributos de volta ao seu estado padrão com o botão *Redefinir*.
- Limpe a tabela usando o botão *Limpar*. Isso é útil quando você tem uma tabela grande, mas deseja mostrar apenas um pequeno número de atributos. Em vez de remover manualmente cada linha, pode ser mais rápido limpar a tabela e adicionar as linhas necessárias.
- Os títulos das células podem ser alterados adicionando o texto personalizado na coluna *Cabeçalho*.
- O alinhamento da célula pode ser gerenciado com a coluna *Alinhamento* que ditará a posição dos textos dentro da célula da tabela.
- A largura da célula pode ser gerenciada manualmente adicionando valores personalizados à coluna *largura*.

Na seção *Ordenação* você pode:

- Adicione um atributo para classificar a tabela com: pressione o |Adicionar simbologia| botão e uma nova linha vazia é adicionada. Insira um campo ou uma expressão na coluna *Atributo* e defina a *Ordem de classificação* como **Ascendente** ou **Descendente**.
- Selecione uma linha na lista e use |seta para cima| e |seta para baixo| botões para alterar a prioridade de classificação no nível do atributo. Selecionar uma célula na coluna *Ordem de classificação* ajuda a alterar a ordem de classificação do campo de atributo.
- Use a |Remover simbologia| botão para remover um atributo da lista de classificação.

Filtragem de recursos

O grupo *Filtragem de recursos* da tabela de atributos fornece as seguintes funcionalidades (veja `figura_layout_tabela_filtro`):

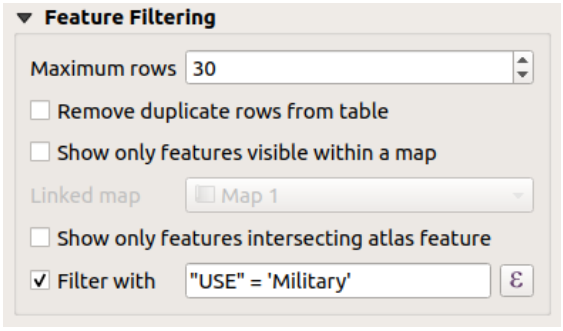


Figura18.45: Tabela de atributos Grupo de filtragem de recursos

Você pode:

- Defina o *Máximo de linhas* a ser exibido.
- Ative |caixa de seleção| *Remova as linhas duplicadas da tabela* para mostrar somente registros únicos.
- Ative |caixa de seleção| *Mostrar apenas feições visíveis dentro de um mapa* e selecione o *Mapa vinculado* cujos atributos de feições visíveis serão exibidos.
- Ative |caixa de seleção| *Mostrar apenas feições que cruzam a feição do Atlas* só está disponível quando |caixa de seleção| *Gerar um atlas* está ativado. Quando ativado, ele mostrará uma tabela apenas com os recursos que cruzam o recurso atual do atlas.
- Ative |caixa de seleção| *Filtrar com* e forneça um filtro digitando na linha de entrada ou insira uma expressão regular usando a |expressão| botão de expressão. Alguns exemplos de instruções de filtragem que você pode usar ao carregar a camada de aeroportos do conjunto de dados de amostra:
  - `ELEVADO > 500`
  - `NOME = 'ANIAK'`
  - `` NOME NÃO GOSTA DE 'AN%'``
  - `ExpressãoRegular_corresponde(atributo($recursoatual, 'USE') , '[i]')`

  A última expressão regular incluirá apenas os aeroportos que possuem a letra 'i' no campo de atributo 'USE'.

Aparência

O grupo *Aparência* da tabela de atributos fornece as seguintes funcionalidades (veja `figura_layout_tabela_aparência`):

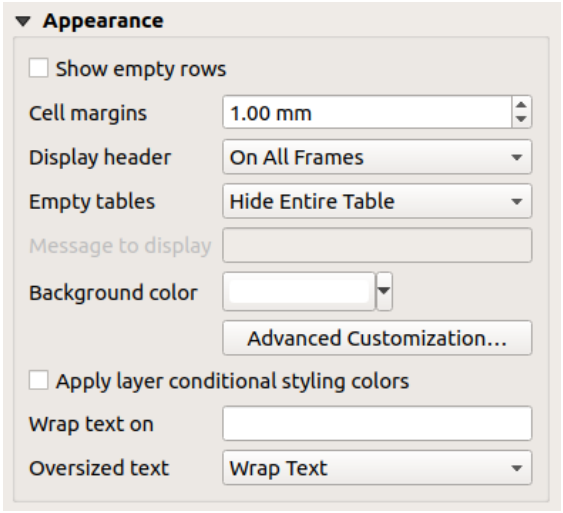


Figura18.46: Grupo de aparência da tabela de atributos

- Clique em |caixa de seleção| *Mostrar linhas vazias* para preencher a tabela de atributos com células vazias. Esta opção também pode ser usada para fornecer células vazias adicionais quando você tiver um resultado para mostrar!
- Com *Margens da célula* você pode definir a margem ao redor do texto em cada célula da tabela.
- Com *Exibir cabeçalho* você pode selecionar de uma lista uma de 'No primeiro quadro', a opção padrão 'Em todos os quadros' ou 'Sem cabeçalho'.
- A opção *Tabela vazia* controla o que será exibido quando a seleção do resultado estiver vazia.
  - **Desenhar apenas cabeçalhos**, apenas desenhará o cabeçalho, exceto se você tiver escolhido 'Sem cabeçalho' para *Exibir cabeçalho*.
  - **Ocultar a tabela inteira**, desenhará apenas o plano de fundo da tabela. Você pode ativar |caixa de seleção| *Não desenhe o fundo se o quadro estiver vazio* em *Quadros* para esconder completamente a tabela.
  - **Mostrar mensagem definida**, desenhará o cabeçalho e adicionará uma célula abrangendo todas as colunas e exibirá uma mensagem como 'Sem resultado' que pode ser fornecida na opção *Mensagem a ser exibida*
- A opção *Mensagem a ser exibida* só é ativada quando você selecionou **Mostrar mensagem definida** para *Tabela Vazia*. A mensagem fornecida será mostrada na tabela na primeira linha, quando o resultado for uma tabela vazia.
- Com *Cor de fundo* você pode definir a cor de fundo da tabela usando a ferramenta seletor de cores `. A opção :guilabel:`Personalização avançada ajuda a definir diferentes cores de fundo para cada célula (veja `figura_layout_tabela_fundo`)

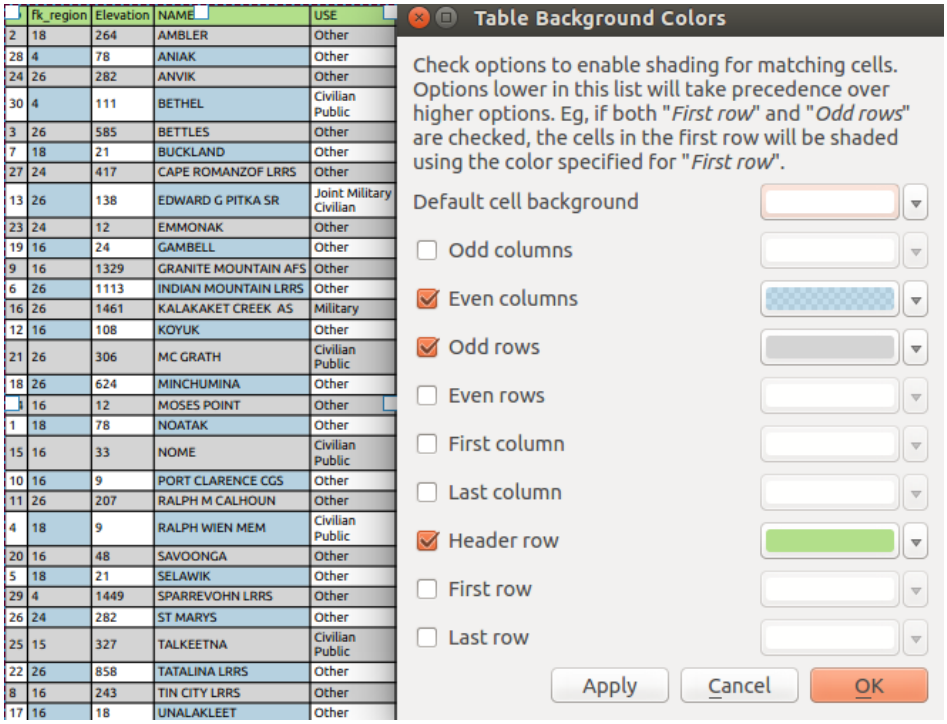


Figura18.47: Caixa de diálogo de fundo avançado da tabela de atributos

- *Apply layer conditional styling colors*: the *conditional table formatting* present in the layer is applied inside the layout attribute table (background color, font family and properties such as bold, italic, strikeout, underline, color, …). Conditional formatting rules take precedence over other layout table formatting settings, e.g. they will override other cell background color settings such as alternating row colors.
- Com a opção *Quebre o texto em*, você pode definir um caractere no qual o conteúdo da célula será encapsulado cada vez que for encontrado
- Com *Texto superdimensionado* você define o comportamento quando a largura definida para uma coluna é menor que o comprimento do seu conteúdo. Pode ser **Quebrar texto** ou **Truncar texto**.

Nota

Mais propriedades do item da tabela de atributos estão descritas na seção tabelas _propriedades_comuns.

O item de tabela fixa

Informações adicionais sobre o mapa podem ser inseridas manualmente em uma tabela escolhendo **ladicionar tabela manuall** *Adicionar tabela fixa* e seguindo as instruções de criação de itens 1 para adicionar um novo item de tabela que você pode manipular mais tarde da mesma forma como exposto em interagir_layout_item.

Por padrão, uma tabela vazia com duas colunas e linhas minimizadas aparece no layout do mapa. Você tem que personalizar a tabela no painel *Propriedades do Item*. Além das propriedades comuns dos itens 1, este recurso tem as seguintes funcionalidades:

Propriedades principais

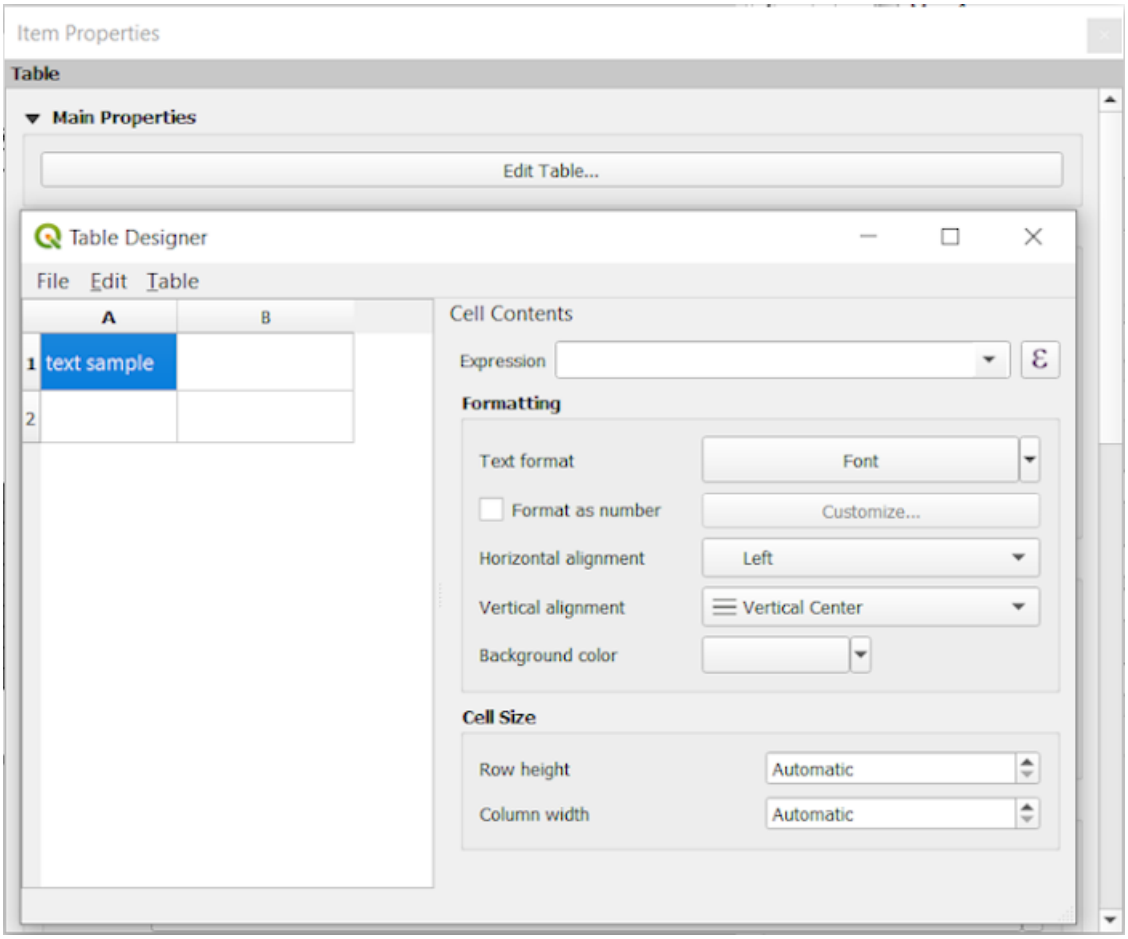


Figura18.48: Painel de propriedades do item de tabela fixa com designer de tabela

In *Main properties*, press *Edit table…* to open the *Table designer* dialog where you can build your table (double-clicking the table item will also open the editor dialog):

- On the left hand of the dialog, a table frame displays the active table item with its content and settings. Click or double-click on a cell and write a text to modify its contents. Press `Ctrl+Enter` to add linebreak (for a multiline text). The logic for selecting multiple cells in the table is the same as in the vector layer *attribute table*. Grab and move a row or column separator to resize the row above or the column before. A double-click on the separator will resize the row above or the column before to fit their contents. A right-click over column or row headers shows a contextual menu to insert columns or rows around the selection or to delete them.
- Through the menus on top of the dialog, it is possible to:
  - *File* ► *Import Content From Clipboard*: it overrides contents of the table.
  - *File* ► *Close editor*: it closes the *Table Designer* dialog.
  - From the *Edit* menu, you can work with selection functionalities for rows and columns:
    - *Select all* cells in the table
    - *Select columns* or *Select Rows*: when cells are selected in the table, you can extend the selection respectively to their column(s) or row(s)
    - *Clear Cells*: deletes the content of the selected cells
  - The *Table* menu is the place where you design the structure of the table. You can:

    - *Insert rows* ► above or below the selection
    - *Insert columns* ► before or after the selection
    - *Delete Rows* or *Delete Columns* of selected cells
    - *Merge selected cells*: select multiple cells in a rectangular shape and you can merge them and concatenate their contents into the top left cell (whose styling is also applied to the merged cell, unless there is a last row/column cell in the selection).
    - *Split selected cells*: select a merged cell and you can split it back to its individual cells. The current text is kept in the top left cell, and the other cells are filled with their contents before they get merged. They are also reapplied their original styling.
    - *Include Header Row* whose styling can only be controlled from the *Fontes e estilo de texto* widget
- Select the cells and, on the right of the table frame, you can provide the *Cell Contents*:
  - Either manually enter the contents of each cell, or use an *Expression* to automatically populate it.
  - Under the *Formatting* group, set specific options that apply to the selection and take precedence over the *global table content styling*:
    - escolhendo o *Formato de texto*
    - por **|caixa de seleção|** *Formatar como número* (vários formatos estão disponíveis)
    - definindo o *Alinhamento horizontal* e o *Alinhamento vertical*
    - escolhendo uma *Cor de fundo*
  - Defina o *Tamanho da célula* com *Altura da linha* e *Largura da coluna*.

Aparência

O grupo *Aparência* da tabela fixa fornece as seguintes funcionalidades:

- Clique em **|caixa de seleção|** *Mostrar linhas vazias* para preencher a tabela de atributos com células vazias.
- Com *Margens da célula* você pode definir a margem ao redor do texto em cada célula da tabela.
- Com *Exibir cabeçalho* você pode selecionar de uma lista uma de 'No primeiro quadro', a opção padrão 'Em todos os quadros' ou 'Sem cabeçalho'.
- Com *Cor de fundo* você pode definir a cor de fundo da tabela usando a ferramenta seletor de cores1. A opção *Personalização avançada* ajuda a definir diferentes cores de fundo para cada célula.
- Com *Texto superdimensionado* você define o comportamento quando a largura definida para uma coluna é menor que o comprimento do seu conteúdo. Pode ser **Quebrar texto** ou **Truncar texto**.

> **Nota**
>
> Mais propriedades do item de tabela fixa são descritas na seção :ref:`tabelas_propriedades_comuns `.

Funcionalidades comuns das tabelas

Mostre as grades

O grupo *Mostre as grades* dos itens da tabela fornece as seguintes funcionalidades (veja `figura_layout_tabela_grade`):

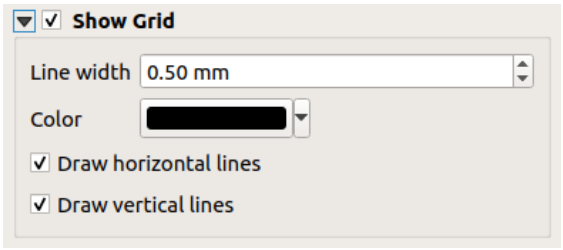


Figura18.49: Tabela de atributos Mostrar grupo de grade

- Ative |caixa de seleção| *Mostrar grade* quando quiser exibir a grade, os contornos das células da tabela. Você também pode selecionar *Desenhar linhas horizontais* ou *Desenhar linhas verticais* ou ambos.
- Com *Espessura da linha* você pode definir a espessura das linhas usadas na grade.
- O *Cor* da grade pode ser definido usando a ferramenta de seleção de cores.

Fontes e estilo de texto

O grupo *Fontes e estilo de texto* dos itens da tabela fornece as seguintes funcionalidades (veja `figura_layout_tabela_fontes`):

Figura18.50: Tabela de atributos Grupo de fontes e estilo de texto

- Você pode definir as propriedades *Fonte* para *Cabeçalho da Tabela* e *Conteúdo da tabela*, usando a ferramenta avançada configurações de texto 1 (com buffer, sombra, efeitos de pintura, transparência, fundo, coloração, …). Observe que essas alterações não afetam as células que têm fonte personalizada atribuída, seja na seção *Aparência* ou na caixa de diálogo *Designer de Tabela*. Apenas as células com a renderização padrão são substituídas.
- For *Table heading* you can additionally set the *Alignment* to `Follow column alignment` or override this setting by choosing `Left`, `Center` or `Right`. The column alignment is set using the *Select Attributes* dialog (see Fig.18.44).

Quadros

O grupo *Quadros* das propriedades do item da tabela fornece as seguintes funcionalidades (veja `figura_layout_tabela_quadros`):

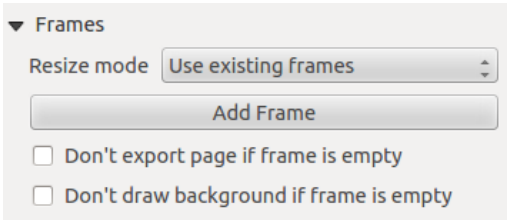


Figura18.51: Grupo de quadros da tabela de atributos

- Com *Modo de redimensionamento* você pode selecionar como renderizar o conteúdo da tabela de atributos:
  - `Usar quadros existentes` exibe o resultado apenas no primeiro frame e nos quadros adicionados.
  - `Estender para a próxima página` criará quantos frames (e páginas correspondentes) forem necessários para exibir a seleção completa da tabela de atributos. Cada quadro pode ser movido no layout. Se você redimensionar um quadro, a tabela resultante será dividida entre os outros quadros. O último quadro será aparado para caber na tabela.
  - `Repetir até terminar` também criará tantos quadros quanto a opção *Estender para a próxima página*, exceto que todos os quadros terão o mesmo tamanho
- Use o botão *Adicionar Quadro* para adicionar outro quadro com o mesmo tamanho do quadro selecionado. O resultado da tabela que não couber no primeiro quadro continuará no próximo quadro quando você usar o modo Redimensionar `Usar quadros existentes`.
- Ative **|caixa de seleção|** *Não exportar página se o quadro estiver vazio* impede que a página seja exportada quando o quadro da tabela não tiver conteúdo. Isso significa que todos os outros itens de layout, mapas, barras de escala, legendas etc. não serão visíveis no resultado.
- Ative **|caixa de seleção|** *Não desenhe o fundo se o quadro estiver vazio* impede que o fundo seja desenhado quando o quadro da tabela não tiver conteúdo.

18.2.8 Os itens de marcador, imagem e seta do norte

- *O Item de Imagem*
  - *Propriedades principais*
  - *Tamanho e posicionamento*
  - *Rotação da imagem*
- *O Item de Seta Norte*
- *O item marcador*

Junto com os itens de mapa ou legenda no layout de impressão, você pode decorar sua realização com imagens ou anotações. O QGIS fornece diferentes ferramentas para conseguir isso:

- o item de imagem: decora o layout com uma imagem raster ou arquivo SVG (por exemplo, logotipos, fotos, setas norte, …)
- o :ref:`item de seta norte `: um item de imagem predefinido com uma imagem de seta norte

- o item marcador: decora o layout com o vetor QGIS símbolos. Ele pode ser usado para colocar marcadores sobre um item do mapa ou para a criação de legendas personalizadas avançadas.

O Item de Imagem

Você pode adicionar uma imagem arrastando-a do gerenciador de arquivos para a tela, colando-a diretamente no layout usando `Ctrl+V` ou :seleção de menu:*Editar –> Colar* e usando o **|adicionar imagem|** Adicionar Imagem, seguindo instruções de criação de itens `. Então você pode manipulá-lo, conforme explicado em :ref:`interagir_layout_item.

Ao usar **|adicionar imagem|** Adicionar imagem, o item de imagem será um quadro em branco que você pode personalizar usando seu painel *Propriedades do item*. Além das :ref:`propriedades comuns dos itens `, este recurso tem as seguintes funcionalidades:

Propriedades principais

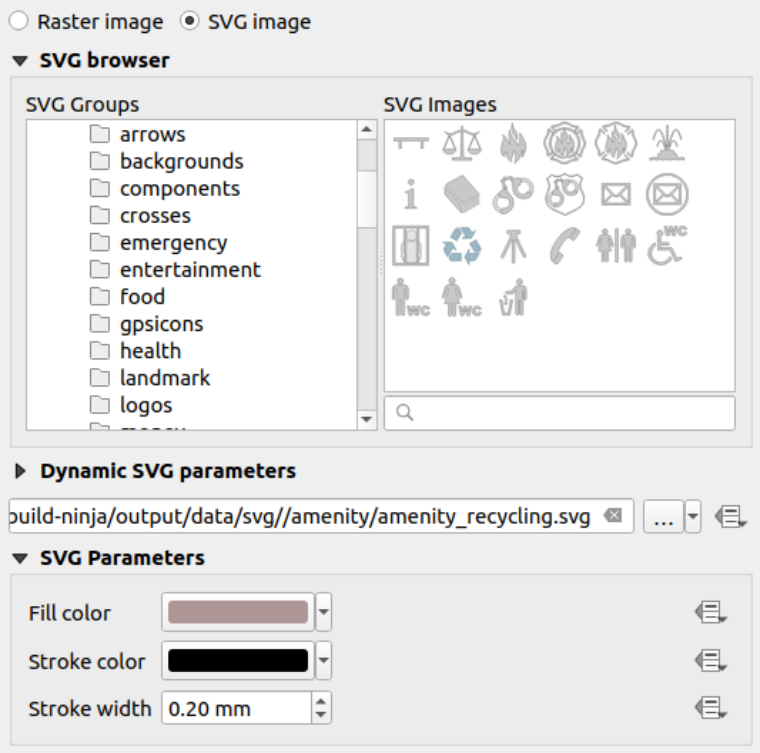


Figura18.52: Painel de Propriedades do Item de Imagem

O item de imagem suporta dois tipos de imagens:

- *Imagem Raster*: uma ferramenta seletora de arquivos pode ser usada para buscar os dados. Use o botão … Navegar para selecionar um arquivo em seu computador ou digite o caminho diretamente no campo de texto. Você pode até fornecer um URL remoto que aponte para uma imagem. A imagem associada também pode ser :ref:`embutida ` no layout.

  Use o **|Definir dados|** sobreposição definida por dados para definir a origem da imagem a partir de um atributo de feição ou usando uma expressão regular.

- *Imagem SVG*: usando por padrão as bibliotecas SVG fornecidas em :seleção de menu:*Configurações –> Opções –>Sistema –> Caminhos SVG*. No entanto, você pode usar qualquer outro arquivo, e a seleção do arquivo segue as mesmas regras da imagem raster. Os parâmetros SVG também podem ser definidos como dinâmicos.

  Os arquivos QGIS fornecidos (padrão) : arquivo:*.SVG* são personalizáveis, o que significa que você pode facilmente aplicar outros *Cor de preenchimento*, *Cor do traço* (incluindo opacidade) e *Largura do traçado ` do que*

o original, usando seu recurso correspondente no grupo :guilabel:`Parâmetros SVG. Essas propriedades também podem ser definidas por dados 1.

Se você adicionar um arquivo:.*SVG* que não habilite essas propriedades, pode ser necessário adicionar as seguintes marcações ao arquivo para adicionar suporte, por exemplo. para transparência:

- *preencher-opacidade="parâmetro(preencher-opacidade)"*
- *opacidade-do-traço="parâmetro(contorno-opacidade)"*

Mais detalhes em svg_símbolo.

Nota

Arrastar e soltar um arquivo de imagem (raster ou SVG) na página de layout criará um item de imagem de layout com as configurações correspondentes.

Tamanho e posicionamento

Figura18.53: Tamanho das imagens de layout e propriedades de posicionamento

Com a opção *Modo de redimensionamento*, você pode definir como a imagem é exibida quando o quadro é redimensionado:

- *Zoom`*: amplia/reduz a imagem para o quadro, mantendo a proporção da imagem
- `Esticar`: estica a imagem para caber dentro do quadro
- `Grampo`: use este modo apenas para imagens raster, ele define o tamanho da imagem para o tamanho original da imagem sem escala, e o quadro é usado para recortar a imagem. Assim, apenas a parte da imagem que está dentro do quadro ficará visível.
- `Ampliar e redimensionar o quadro`: amplia a imagem para caber no quadro e, em seguida, redimensiona o quadro para ajustar as dimensões da imagem resultante
- `Redimensionar o quadro para o tamanho da imagem`: define o tamanho do quadro para corresponder ao tamanho original da imagem (sem dimensionamento)

Dependendo do *Modo de redimensionamento* selecionado, as opções *Colocação* e *Rotação da imagem* podem ser desabilitadas. *Placement* permite que você selecione a posição da imagem dentro de seu quadro (superior/meio/inferior e esquerda/centro/direita).

Rotação da imagem

As imagens podem ser giradas com o campo *Rotação da imagem*. Ativando a |caixa de seleção| A caixa de seleção *Sincronizar com o mapa* sincroniza a rotação da imagem com a rotação aplicada ao item do mapa selecionado. Este é um recurso conveniente para fazer qualquer imagem se comportar como uma seta para o norte. O *Alinhamento Norte* pode ser:

- **Grade norte**: a direção de uma linha de grade paralela ao meridiano central da grade nacional/local
- **Norte verdadeiro**: direção de um meridiano de longitude.

Você também pode aplicar uma declinação *Deslocamento* à rotação da imagem.

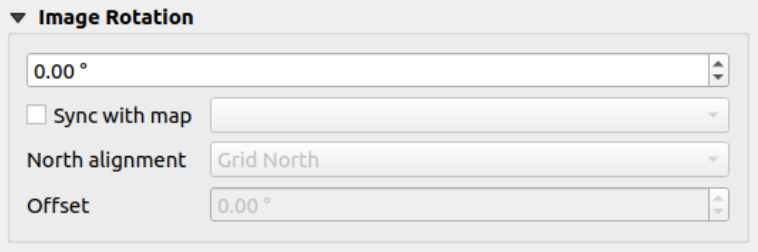


Figura18.54: Propriedades de rotação de imagem de imagens de layout

O Item de Seta Norte

Você pode adicionar uma seta norte com a |seta norte| Adicionar seta norte, seguindo instruções de criação de itens 1 e manipule-o da mesma forma como exposto em interagir_layout_item.

Como as setas do norte são imagens, o item *Seta do Norte* tem as mesmas propriedades que o item de imagem 1. As principais diferenças são:

- Uma seta norte padrão é usada ao adicionar o item, em vez de um quadro em branco
- O item de seta norte é sincronizado com um item de mapa por padrão: a propriedade *Sincronizar com o mapa* é o mapa sobre o qual o item de seta norte é desenhado. Se nenhum, ele volta para o mapa de referência 1.

> **Nota**
>
> Muitas das setas do norte não têm um 'N' adicionado na seta do norte. Isso é feito de propósito, pois existem idiomas que não usam um 'N' para Norte.

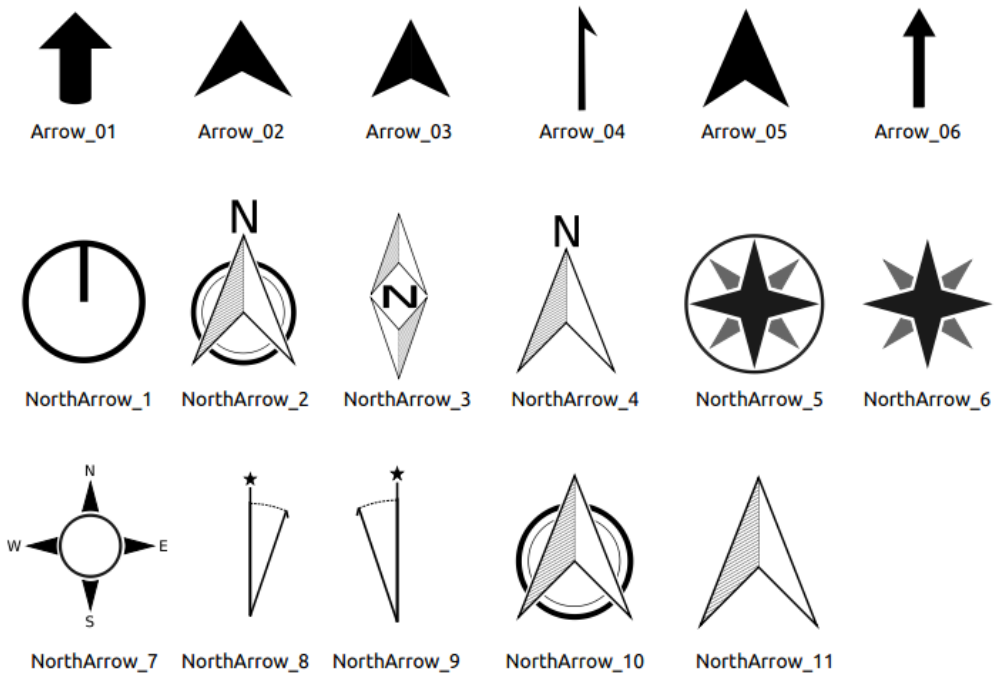


Figura18.55: Setas do norte disponíveis para seleção na biblioteca SVG fornecida

O item marcador

Para adicionar um item de marcador, selecione o **|adicionar marcador|** Adicionar Marcador, e clique na página. Um símbolo de marcador de ponto padrão é adicionado. Então você pode manipulá-lo, conforme explicado em interagir_layout_item. Mas observe que, diferentemente da maioria dos outros itens, você redimensiona o item, pois seu tamanho é controlado pelas propriedades dos símbolos incorporados.

O item marcador pode ser personalizado a partir do painel *Propriedades do Item*. Além das :ref:`propriedades comuns dos itens 1 `, você também pode:

- modifique o *Símbolo*, contando com todos os símbolos :ref:`capacidades do ferramenta 1 `
- sincronize a rotação do item do marcador com a do mapa (veja layout_imagens_rotação), agindo como uma seta para o norte. A rotação do mapa é adicionada a qualquer rotação de nível de símbolo de marcador existente (por exemplo, se você tiver que girar o marcador de triângulo 90° para que ele aponte para cima, ele ainda funcionará bem no modo de seta para o norte!)

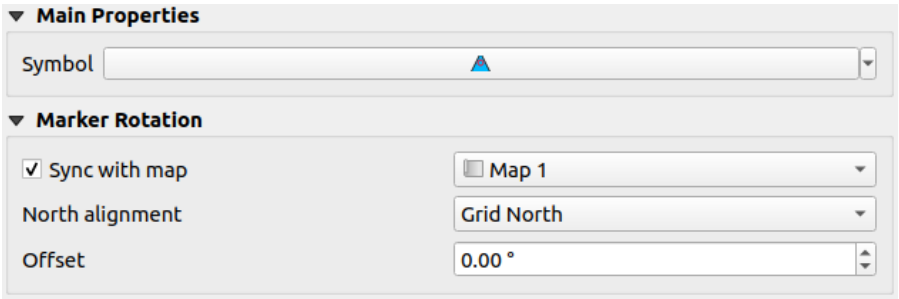


Figura18.56: As propriedades personalizadas do item marcador

18.2.9 The Elevation Profile Item

The Elevation Profile item is used to display an *elevation profile view* in a layout. Use the *Add Elevation Profile* button, and follow *items creation instructions* to add a new elevation profile item that you can later manipulate the same way as demonstrated in *Interagindo com itens de layout*.

A new elevation profile item has default settings that make it render an empty chart. You can customize its properties in the *Item Properties* panel. In addition to the *common properties*, this feature has the following functionalities:

The Elevation profile *Item Properties* panel embeds a top toolbar with the following functionalities:

- Update elevation profile to refresh the item rendering
- Copy from elevation profile: a drop-down menu to select an elevation profile view from. The view settings are applied to the layout elevation profile item, and can be later modified.

Camadas

Under the *Layers* group, check in the tree view the layers you would like to render in the profile item. Remember to properly configure the *Elevation* properties of the selected layers.

Profile curve

- *Controlled by atlas*: the *profile curve* will be taken from the current atlas feature and the elevation profile view updated as you walk through the atlas features. This is currently supported for an active layout atlas or report, using a line geometry type coverage layer.
- The *Tolerance* distance, which can be data-defined, helps you control how far a feature of the visible layers should be from the profile curve in order to display in the layout elevation view. Only point features are currently returned.

Show Subsections Indicator

It is possible to highlight the position of the vertices of the curve on the Elevation Profile. Select the *Show Subsections Indicator* to display vertical lines at the distances of the vertices. Use the *Lines* drop-down list to set the symbol, color, opacity and width of the Subsections Indicator lines.

Chart ranges

A layout elevation profile item does not necessarily display the full extent of the elevation profile view it is based on. You can limit the area to render, providing:

- on the X axis, the *Minimum distance* and *Maximum distance* from the profile curve starting point
- On the Y axis, the *Minimum elevation* and *Maximum elevation*

Distance and elevation axes

The *Distance axis* and *Elevation axis* groups give options to tweak the grid over the elevation profile item, respectively on the X and Y axes:

- the distance display *Unit*, allowing to override the map canvas unit
- the graduation on the axis with both a *Major interval* and *Minor interval*
- the line symbols to apply to the corresponding *Major grid lines* and *Minor grid lines*
- how regular the graduation items should be labeled (*Label interval*) as well as their *Label format* and *Label font*
- the *Distance labels*: configures whether the distance unit symbol should be placed next to *Every value*, *First value*, *Last value*, *First and last values*, or skipped (*None*)

Chart area

Under *Chart area*, you can configure the rendering of the area in which the elevation profile plot is actually displayed:

- a *Background* fill symbol
- a *Border* line symbol
- the margins from the elevation profile item border

18.2.10 O quadro com item em HTML

É possível adicionar um quadro que exiba o conteúdo de um site ou até mesmo criar e estilizar sua própria página HTML e exibi-la! Você pode adicionar uma imagem com o **|adicionar HTML|** *Adicione HTML* seguindo as instruções de criação de itens ` e manipule-o da mesma forma como exposto em :ref:`interagir_layout_item. Observe que a escala HTML é controlada pela resolução de exportação do layout no momento em que o quadro HTML é criado.

O item HTML pode ser personalizado usando seu painel *Propriedades do item*. Além das **:referência:`propriedades comuns dos itens `, este recurso tem as seguintes funcionalidades (veja :número ref:`figura_layout_html`**):

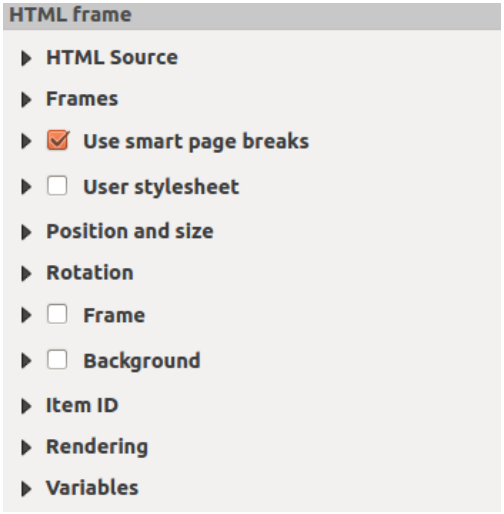


Figura18.57: Quadro HTML, o Painel de Propriedades do Item

Origem HTML

O grupo *Origem HTML* do painel HTML quadro *Propriedades do item* fornece as seguintes funcionalidades (veja :número ref:*figure_layout_html_ppt*):

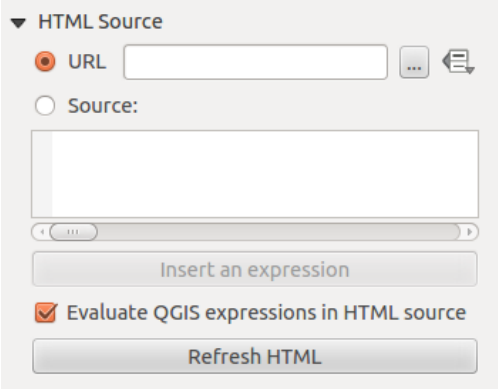


Figura18.58: quadro HTML, as propriedades da fonte HTML

- Em *URL* você pode inserir a URL de uma página da web que você copiou do seu navegador de Internet ou selecionar um arquivo HTML usando o botão ... Procurar. Há também a opção de usar o **|Definir dados|** :sup: Botão *Substituição definida por dados*, para fornecer uma URL do conteúdo de um campo de atributo de uma tabela ou usando uma expressão regular.
- Em *Fonte* você pode inserir texto na caixa de texto com algumas marcações HTML ou fornecer uma página HTML completa.
- O botão *Inserir ou editar uma expressão...* pode ser usado para adicionar uma expressão como `[%Ano($now)%]` na caixa de texto Fonte para exibir o ano atual. Este botão só é ativado quando o botao de radio *Fonte* é selecionado. Após inserir a expressão, clique em algum lugar na caixa de texto antes de atualizar o quadro HTML, caso contrário você perderá a expressão.
- Ative **|caixa de seleção|** *Avaliar expressões QGIS em código HTML* para ver o resultado da expressão que você incluiu, caso contrário você verá a expressão.
- Use o botão *Atualizar HTML* para atualizar o(s) quadros(s) HTML e ver o resultado das alterações.

Quadros

O grupo *Quadros* do painel HTML frame *Propriedades do item* fornece as seguintes funcionalidades (veja :número ref:*figura_layout_html_quadros*):

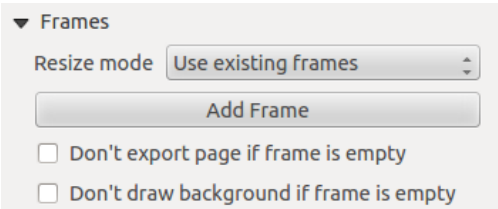


Figura18.59: quadro HTML, as propriedades dos quadros

- Com *Modo de redimensionamento* você pode selecionar como renderizar o conteúdo HTML:
  - `Usar quadros existentes` exibe o resultado apenas no primeiro frame e nos quadros adicionados.
  - `Estender para a próxima página` criará quantos frames (e páginas correspondentes) forem necessários para renderizar a altura da página web. Cada quadro pode ser movido no layout. Se você redimensionar um quadro, a página da Web será dividida entre os outros quadros. O último quadro será cortado para caber na página da web.

  - `Repetir em cada página` irá repetir o canto superior esquerdo da página web em cada página em quadros do mesmo tamanho.
  - `Repetir até terminar` também criará tantos quadros quanto a opção `Estender para a próxima página`, exceto que todos os quadros terão o mesmo tamanho.
- Use o botão *Adicionar quadro* para adicionar outro quadro com o mesmo tamanho do quadro selecionado. Se a página HTML não couber no primeiro frame, ela continuará no próximo frame quando você usar *Modo de redimensionamento* ou *Usar quadros existentes*.
- Ative **|caixa de seleção|** *Não exportar página se o quadro estiver vazio* impede que a página seja exportada quando o quadro não tem conteúdo HTML. Isso significa que todos os outros itens de layout, mapas, barras de escala, legendas etc. não serão visíveis no resultado.
- Ative **|caixa de seleção|** *Não desenhe o fundo se o quadro estiver vazio* impede que o quadro HTML seja desenhado se o quadro estiver vazio.

Use quebras de página inteligentes e folha de estilo do usuário

A caixa de diálogo *Usar quebras de página inteligentes* e a caixa de diálogo *Folha de estilo do usuário* do painel HTML *Propriedades do item* fornecem as seguintes funcionalidades (veja :número ref:*figura_layout_html_quebras*):

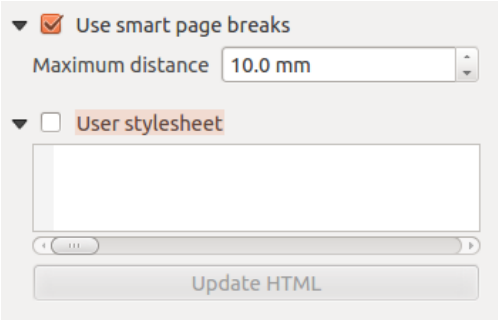


Figura18.60: Quadro HTML, Usar quebras de página inteligentes e Propriedades da folha de estilo do usuário

- Ative **|caixa de seleção|** *Use quebras de página inteligentes* para evitar que o conteúdo do quadro html quebre no meio de uma linha de texto para que continue agradável e suave no próximo quadro.
- Defina a *Distância máxima* permitida ao calcular onde colocar as quebras de página no html. Essa distância é a quantidade máxima de espaço vazio permitida na parte inferior de um quadro após calcular o local ideal de quebra. Definir um valor maior resultará em uma melhor escolha do local da quebra de página, mas mais espaço desperdiçado na parte inferior dos quadros. Isso só é usado quando *Usar quebras de página inteligentes* está ativado.
- Ative **|caixa de seleção|** *Folha de estilo do usuário* para aplicar estilos HTML que geralmente são fornecidos em folhas de estilo em cascata. Um exemplo de código de estilo é fornecido abaixo para definir a cor da marcação de cabeçalho `1` para verde e definir a fonte e o tamanho da fonte do texto incluído nas marcações de parágrafo `2`.

```
h1 {color: #00ff00;
}
p {font-family: "Times New Roman", Times, serif;
   font-size: 20px;
}
```

- Use o botão *Atualizar HTML* para ver o resultado das configurações da folha de estilo.

18.2.11 Itens do shape

Qgis proporciona um leque de ferramentas para desenhar formas complexas ou regulares sobre o layout de impressão.

> **Nota**
>
> Ao contrário de outros itens do layout de impressão, você não pode definir o estilo do quadro, nem da cor de fundo do quadro delimitador de formas (definido como transparente por padrão).

Os Itens regulares do Shape

O item *Shape* é uma ferramenta que ajuda a decorar seu mapa com formas regulares como triângulo, retângulo, elipse… Você pode adicionar uma forma regular usando o |add shape básico| :sup:'Add Shape' que dá acesso a ferramentas específicas como |add retângulo básico| Adicionar Retângulo, |add círculo básico| Add Elipse e |add triângulo básico| Adicionar Triângulo. Depois de selecionar a ferramenta apropriada, você pode desenhar o item seguindo as instruções de criação de itens `. Como outros itens de layout, uma forma regular pode ser manipulada da mesma forma que exposta em :ref:`interagir_layout_item.

> **Nota**
>
> Manter pressionada a tecla `Shift` enquanto desenha a forma básica com o método de clicar e arrastar ajuda a criar um quadrado, círculo ou triângulo perfeito.

O item de forma padrão pode ser personalizado usando seu painel *Propriedades do item 1*. Além das propriedades comuns dos itens `, este recurso tem as seguintes funcionalidades (veja :numref:`figura_layout_basico_shape):

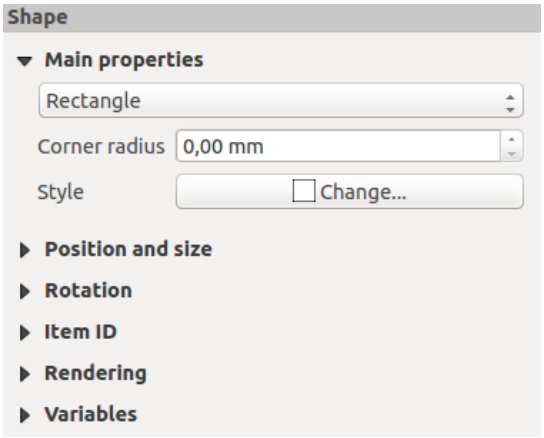


Figura18.61: Painel propriedades do Item Shape

O grupo *Principais propriedades* mostra e permite que você alterne o tipo do item de forma (**Elipse**, **Retângulo** ou **Triângulo**) dentro do quadro fornecido.

Você pode definir o estilo da forma usando a ferramenta seletora avançado símbolo 1 e cor 2…

Para a forma de retângulo, você pode definir em unidades diferentes o valor de *Ângulo de raio* para arredondar os cantos.

Os itens de forma baseados em nós

Enquanto o |adicionar Forma Básica| A ferramenta *Add Shape* fornece uma maneira de criar um item geométrico simples e predefinido, o |adicionar Forma de Nós| A ferramenta *Adicionar item de nó* ajuda você a criar um item geométrico personalizado e mais avançado. Para polilinhas ou polígonos, você pode desenhar quantas linhas ou lados quiser e os vértices dos itens podem ser manipulados de forma independente e direta usando o |editar nós de Shape| *Editar item de nós*. O próprio item pode ser manipulado conforme exposto em interagir_layout_item.

editar a forma dos nós

1. Clique em |adicionar forma de nós| :sup:'Adicionar item de nó' ícone
2. Selecione |add polígono| Adicionar Polígono ou |add Polilinha| :sup:Ferramenta *Adicionar polilinha*
3. Execute cliques esquerdos consecutivos para adicionar nós do seu item. Se você mantiver pressionada a tecla `Shift` enquanto desenha um segmento, ele é restringido a seguir uma orientação múltipla de 45|graus|.
4. Quando terminar, clique com o botão direito do mouse para encerrar a forma.

Você pode personalizar a aparência da forma no painel *Propriedades do item*.

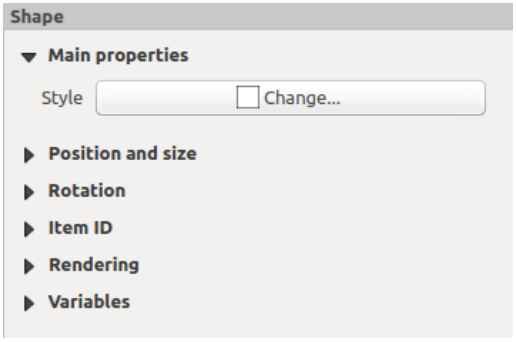


Figura18.62: Painel de Propriedades do Item de Forma de Nó de Polígono

Nas *Propriedades principais*, você pode definir o estilo da forma usando a ferramenta seletora avançada símbolo 1 e cor 2…

Para itens de nó de polilinha, você também pode parametrizar o *Marcadores de linha*, ou seja, adicionar:

- marcadores de início e/ou fim com opções:
  - *Nenhum*: desenha uma polilinha simples.
  - *Seta*: adiciona uma ponta de seta triangular regular que você pode personalizar.
  - *SVG* marcador: usa um arquivo: *SVG* como ponta de seta do item.
- personalize a cabeça da seta:
  - *Cor do traço da seta*: define a cor do traço da ponta da seta.
  - *Cor de preenchimento da seta*: Defina a cor de preenchimento da seta.
  - *Largura do contorno da seta* : Defina a largura do contorno da seta.
  - *Espessura da ponta da seta*: Defina a espessura da seta.

As imagens SVG são giradas automaticamente com a linha. As cores de traçado e preenchimento de imagens SVG predefinidas do QGIS podem ser alteradas usando as opções correspondentes. SVG personalizado pode exigir algumas tags seguindo esta instrução 1.

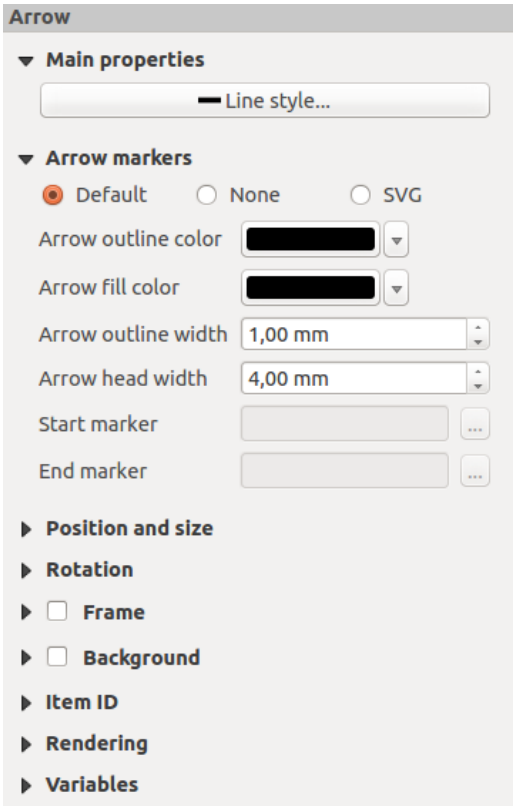


Figura18.63: Painel propriedades do Item Shape Nó Polilinha

Item da Seta

A |adicionar seta| A ferramenta Adicionar seta é um atalho para criar uma polilinha habilitada para seta por padrão e, portanto, tem as mesmas propriedades e comportamento de um :ref:`item de nó de polilinha `.

Na verdade, o item de seta pode ser usado para adicionar uma seta simples, por exemplo, para mostrar a relação entre dois itens de layout de impressão diferentes. No entanto, para criar uma seta de norte, o imagem item 1 deve ser considerado primeiro, pois dá acesso a um conjunto de setas de norte no formato **:arquivo:`.SVG`** que você pode sincronizar com um item de mapa para que ele gira automaticamente com ele.

Editar um ítem nó na geometria

Uma ferramenta específica é fornecida para editar formas baseadas em nós através de **|editar a forma dos nós|** Editar item de nós. Dentro deste modo, você pode selecionar um nó clicando nele (um marcador é exibido no nó selecionado). Um nó selecionado pode ser movido arrastando-o ou usando as teclas de seta. Além disso, neste modo, você pode adicionar nós a uma forma existente: clique duas vezes em um segmento e um nó é adicionado no local em que você clicar. Finalmente, você pode remover o nó selecionado pressionando a tecla `Del`.

18.3 Criando um arquivo de Saída

Fig.18.64 shows an example print layout including all the types of layout items described in the previous section.

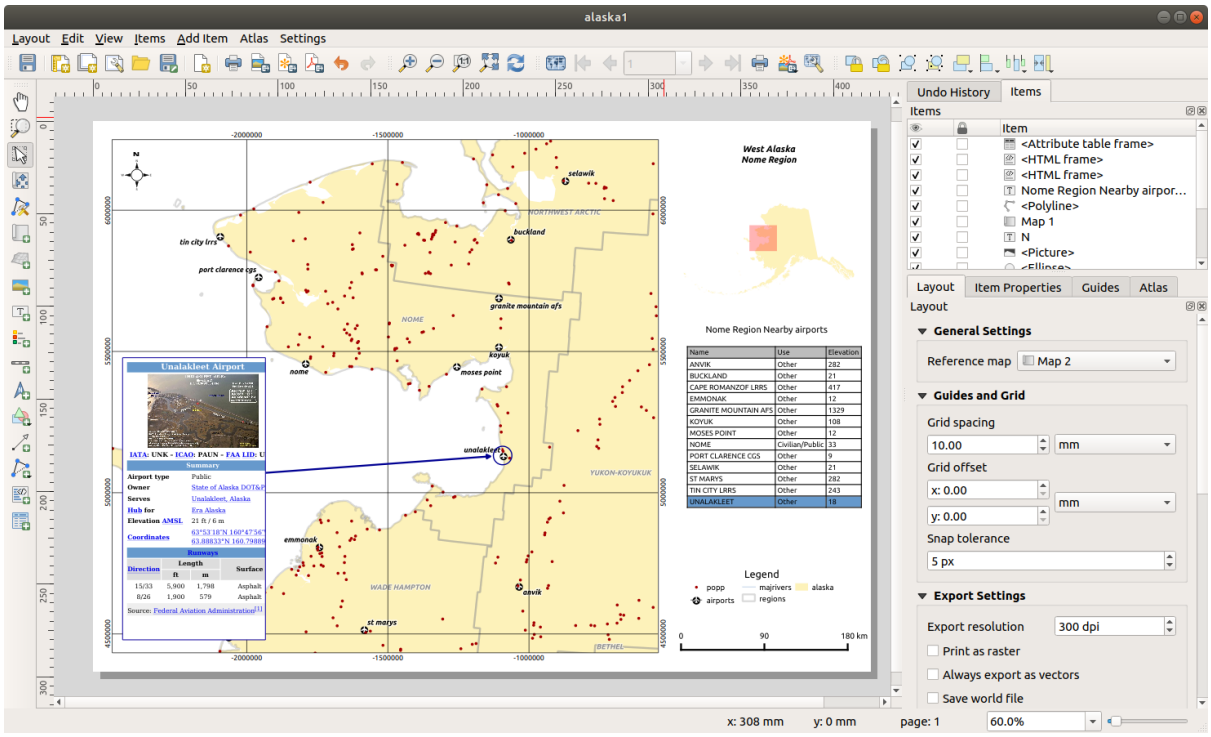


Figura18.64: Print Layout with map view, legend, image, scale bar, coordinates, text and HTML frame added

From the *Layout* menu or toolbar, you can output the print layout to different file formats, and it is possible to modify the resolution (print quality) and paper size:

- The Print icon allows you to print the layout to a connected printer or a PostScript file, depending on the installed printer drivers.
- The Export as image icon exports the print layout image formats such as `PNG`, `BMP`, `TIF`, `JPG`, and many others…
- The Export as SVG icon saves the print layout as an `SVG` (Scalable Vector Graphic).
- O ícone Exportar como PDF salva o layout de impressão definido diretamente como um arquivo `PDF` (Portable Document Format).

18.3.1 Definições para exportação

Whenever you export a print layout, there are a selection of export settings QGIS needs to check in order to produce the most appropriate output. These configurations are:

- The *Export settings* of the *Layout* panel, such as *Export resolution*, *Print as raster* *Always export as vectors* or *Save world file*
- *Exclude page from exports* in the *page item properties* panel
- *Exclude item from exports* in the *item properties* panel

Moreover, a number of predefined checks are automatically applied to the layout. Currently these checks include testing that scalebars are correctly linked to map items, and that map overview items are also correctly linked to a map. If the checks fail, you are shown a nice warning advising you of the issue.

18.3.2 Exportar como Imagem

Para exportar um layout como uma imagem:

1. Clique no ícone Exportar como imagem
2. Select the image format, the folder and filename (e.g. `myill.png`) to use. If the layout contains more than one page, each page will be exported to a file with the given filename with the page number appended (e.g. `myill_2.png`).
3. Na próxima caixa de diálogo (*Opções de Exportação de Imagem*):
   - You can override the print layout *Export resolution* and the exported page dimensions (as set in *Layout* panel).
   - For JPEG and JPG image formats, you can set the *JPEG Quality* of the output image.
   - Image rendering can also be improved with the *Enable antialiasing* option.
   - If you want to export your layout as a **georeferenced image** (e.g., to share with other projects), check the *Generate world file* option, and an *ESRI World File* with the same name as the exported image, but a different extension (`.tfw` for TIFF, `.pnw` for PNG, `jgw` for JPEG, …) will be created when exporting. This option can also be checked by default in the *layout panel*.

     > **Nota**
     >
     > For multi-page output, only the page that contains the *reference map* will get a world file (assuming that the *Generate world file* option is checked).

   - By checking *Crop to content* option, the image output by the layout will include the minimal area enclosing all the items (map, legend, scale bar, shapes, label, image…) of each page of the composition:
     - Se a composição incluir uma única página, a saída será redimensionada para incluir TUDO na composição. A página pode ser reduzida ou estendida a todos os itens, dependendo da sua posição (na, acima, abaixo, esquerda ou direita da página).
     - In case of a multi-page layout, each page will be resized to include items in its area (left and right sides for all pages, plus top for the first page and bottom for the last page). Each resized page is exported to a separate file.

     The *Crop to content* dialog also lets you add margins around the cropped bounds.
   - By checking *Open file after exporting* the exported file will automatically open in the default image viewer.

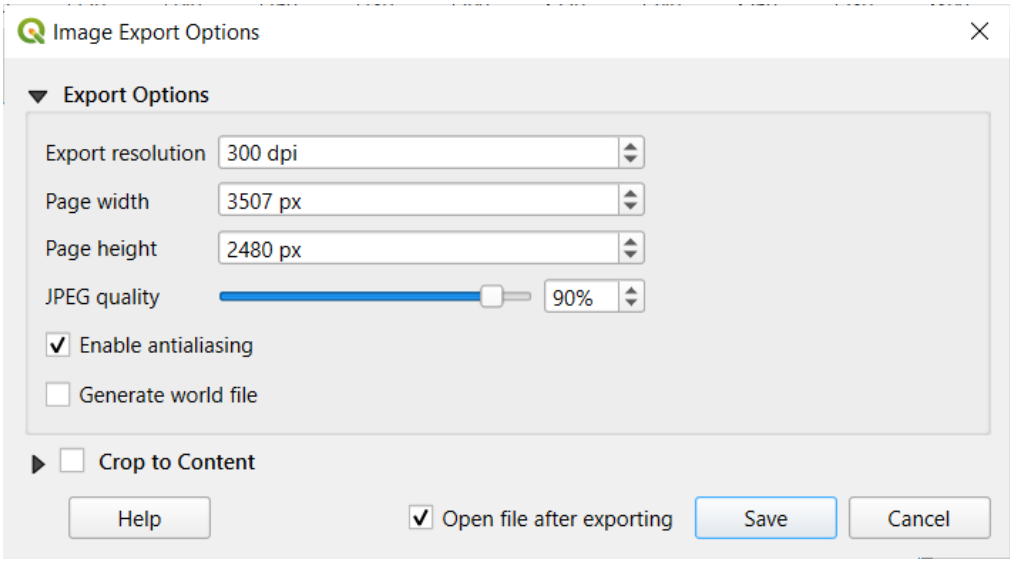


Figura18.65: Image Export Options

Dica

Use formatos de imagem compatíveis com transparência quando os itens ultrapassarem a extensão do papel

Layout items may be placed outside the paper extent. When exporting with the *Crop to content* option, the resulting image may therefore extend beyond the paper extent. Since the background outside of the paper extent will be transparent, for image formats that do not support transparency (e.g. `BMP` and `JPG`) the transparent background will be rendered as full black, "corrupting" the image. Use transparency-compatible formats (e.g. `TIFF` and `PNG`) in such cases.

Nota

When supported by the format (e.g. `PNG`) and the underlying Qt library, the exported image may include *project metadata* (author, title, date, description…)

18.3.3 Exportar como SVG

Para exportar um layout como SVG:

1. Clique no ícone Exportar como SVG
2. Fill in the path and filename (used as a base name for all the files in case of multi-page composition, as for image export)
3. In the next *SVG Export Options* dialog, you can override the layout default *export settings* or configure new ones:
   - *Export map layers as SVG groups*: exported items are grouped within layers whose name matches the layer names from QGIS, making it much easier to understand the contents of the document.
   - *Always export as vectors*: some rendering options require items to be rasterized for a better rendering. Check this option to keep the objects as vectors with the risk that the appearance of the output file may not match the print layout preview (for more details, see *Definições para exportação*).
   - *Exportar metadados RDF* do documento, como título, autor, data, descrição…

- *Simplify geometries to reduce output file size*: this avoids exporting ALL geometry vertices, which can result in a ridiculously complex and large export file size that could fail to load in other applications. Geometries will be simplified while exporting the layout in order to remove any redundant vertices which are not discernably different at the export resolution (e.g. if the export resolution is `300 dpi`, vertices that are less than `1/600 inch` apart will be removed).
- Set the *Text export*: controls whether text labels are always or preferably exported as *text or outline objects*.
- Apply *Crop to content* *option*
- *Disable tiled raster layer exports*: When exporting files, QGIS uses a built-in raster layer tiled rendering that saves memory. Sometimes, this can cause visible "seams" in the rasters for generated files. Checking this option would fix that, at the cost of a higher memory usage during exports.
- By checking *Open file after exporting* the exported file will automatically open in the default SVG viewer.

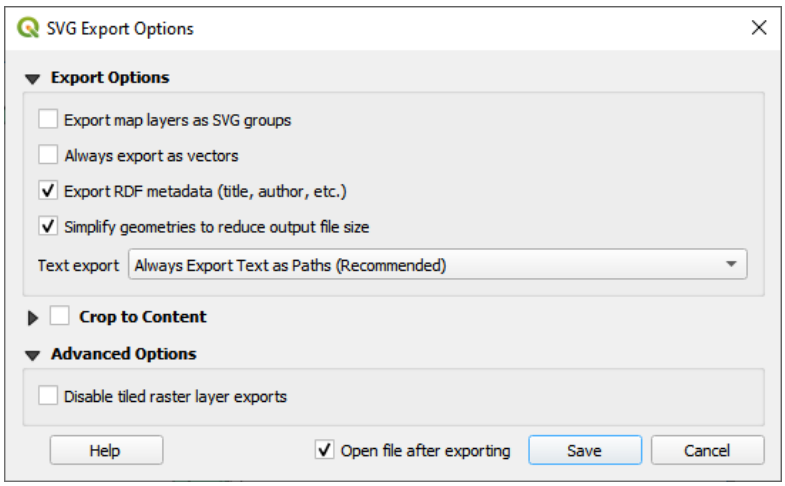


Figura18.66: Opções de Exportação SVG

Nota

Currently, the SVG output is very basic. This is not a QGIS problem, but a problem with the underlying Qt library. This will hopefully be sorted out in future versions.

18.3.4 Exportar como PDF

Para exportar um layout como PDF:

1. Click the Export as PDF icon
2. Preencha o caminho e o nome do arquivo: diferente da exportação de imagem e SVG, todas as páginas do layout são exportadas para um único arquivo PDF.
3. In the next *PDF Export Options* dialog, you can override the layout default *export settings* or configure new ones:
   - *Always export as vectors*: some rendering options require items to be rasterized for a better rendering. Check this option to keep the objects as vectors with the risk that the appearance of the output file may not match the print layout preview (for more details, see *Definições para exportação*).

> **Nota**
>
> **Mind opacity settings for vector layers**
>
> When you set a layer-wide opacity (or opacity for any vector symbology) to less than 100%, it forces a rasterized rendering of those objects during PDF exports. Depending on the size of the layer, this may considerably increase the PDF file size.

- *Append georeference information*: available only if the *reference map*, from which the information is taken, is on the first page.
- *Exportar metadados RDF* do documento, como título, autor, data, descrição…
- Set the *Text export*: controls whether text labels are always or preferably exported as *text or outline objects*.
- Control the PDF *Image compression* using:
  - *Lossy (JPEG)*, which is the default compression mode
  - or *Lossless*, which creates bigger files in most cases, but is much more suitable for printing outputs or for post-production in external applications (requires Qt 5.13 or later).
- *Create Geospatial PDF*: Generate a georeferenced PDF file.
- *Disable tiled raster layer exports*: When exporting files, QGIS uses tiled based rendering that saves memory. Sometimes, this can cause visible "seams" in the rasters for generated files. Checking this option would fix that, at the cost of a higher memory usage during exports.
- *Simplify geometries to reduce output file size*: Geometries will be simplified while exporting the layout by removing vertices that are not discernably different at the export resolution (e.g. if the export resolution is `300 dpi`, vertices that are less than `1/600 inch` apart will be removed). This can reduce the size and complexity of the export file (very large files can fail to load in other applications).
- By checking *Open file after exporting* the exported file will automatically open in the default PDF viewer.

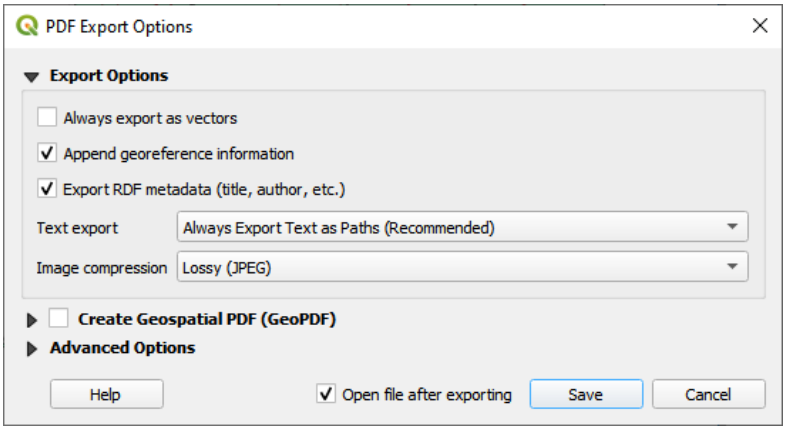


Figura18.67: Opções de exportação de PDF

> **Nota**
>
> Geospatial PDF export is supported, and a number of Geospatial PDF specific options are available:
>
> - *Format* (Geospatial PDF format - there are some variations),
> - *Include multiple map themes* (specify map themes to include),

- *Include vector feature information* (choose the layers and group them into logical PDF groups).

Nota

A exportação de um layout de impressão para formatos que suportam georreferenciamento (por exemplo `PDF` e `TIFF`) cria uma saída georreferenciada por padrão.

Nota

When exporting maps or layouts to vector formats such as PDF or SVG, line pattern fills are rendered using a line-by-line vector-based approach instead of rasterized tiles. This change results in smaller and higher-quality output files, with individual pattern lines remaining fully editable in external vector graphics software.

18.3.5 Gerar um Atlas

Atlas functions allow you to create map books in an automated way. Atlas uses the features of a table or vector layer (*Coverage layer*) to create an output for each feature (**atlas feature**) in the table / layer. The most common usage is to zoom a map item to the current atlas feature. Further use cases include:

- um item de mapa mostrando, para outra camada, apenas feições que compartilham o mesmo atributo que a feição atlas ou estão dentro de sua geometria.
- um rótulo ou item HTML cujo texto é substituído à medida que as feições são iteradas
- a table item showing attributes of associated *parent or children* features of the current atlas feature…

Para cada feição, a saída é processada para todas as páginas e itens de acordo com suas configurações de exportação.

Dica

Usar variáveis para maior flexibilidade

QGIS provides a large panel of functions and *variables*, including atlas related ones, that you can use to manipulate the layout items, but also the symbology of the layers, according to atlas status. Combining these features gives you a lot of flexibility and helps you easily produce advanced maps.

To enable the generation of an atlas and access atlas parameters, refer to the *Atlas* panel. This panel contains the following (see Fig.18.68):

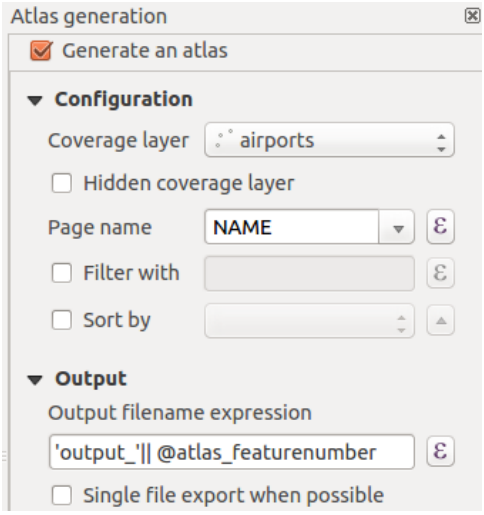


Figura18.68: Atlas Panel

- *Gerar um atlas* habilita ou desabilita a geração de atlas.
- *Configuração*
  - A *Coverage layer* combo box that allows you to choose the table or vector layer containing the features to iterate over.
  - An optional *Hidden coverage layer* that, if checked, will hide the coverage layer (but not the other layers) during the generation.
  - An optional *Page name* combo box to specify the name for the feature page(s). You can select a field of the coverage layer or set an *expression*. If this option is empty, QGIS will use an internal ID, according to the filter and/or the sort order applied to the layer.
  - An optional *Filter with* text area that allows you to specify an expression for filtering features from the coverage layer. If the expression is not empty, only features that evaluate to `True` will be processed.
  - An optional *Sort by* that allows you to sort features of the coverage layer (and the output), using a field of the coverage layer or an expression. The sort order (either ascending or descending) is set by the two-state *Sort direction* button that displays an up or a down arrow.
- *Output* - this is where the output of the atlas can be configured:
  - An *Output filename expression* textbox that is used to generate a filename for each atlas feature. It is based on expressions. is meaningful only for rendering to multiple files.
  - A *Single file export when possible* that allows you to force the generation of a single file if this is possible with the chosen output format (`PDF`, for instance). If this field is checked, the value of the *Output filename expression* field is meaningless.
  - Uma lista drop-down *Formato de exportação de imagem* para selecionar o formato de saída quando usar o botão Exportar atlas como Imagens ….

Control map by atlas

The most common usage of atlas is with the map item, zooming to the current atlas feature, as iteration goes over the coverage layer. This behavior is set in the *Controlled by atlas* group properties of the map item. See *Controlado por atlas* for different settings you can apply on the map item.

Customizar rótulos com expressão

Para adaptar rótulos ao recurso em que o atlas repete, você pode incluir expressões. Certifique-se de colocar a parte da expressão (incluindo funções, campos ou variáveis) entre `[%` e `%]` (veja *A rotulagem de item* para mais detalhes).

For example, for a city layer with fields `CITY_NAME` and `ZIPCODE`, you could insert this:

```
The area of [% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ',
format_number($area/1000000, 2) ) %] km2
```

ou outra combinação:

```
The area of [% upper(CITY_NAME)%],[%ZIPCODE%] is
[%format_number($area/1000000,2) %] km2
```

The information `[% concat( upper(CITY_NAME), ',', ZIPCODE, ' is ', format_number($area/1000000, 2) ) %]` is an expression used inside the label. Both expressions would result in the following type of label in the generated atlas:

```
The area of PARIS,75001 is 1.94 km2
```

Explore Data-defined override buttons with atlas

There are several places where you can use a Data defined override button to override the selected setting. This is particularly useful with atlas generation. See *Data defined override setup* for more details on this widget.

For the following examples the `Regions` layer of the QGIS sample dataset is used and selected as *Coverage layer* for the atlas generation. We assume that it is a single page layout containing a map item and a label item.

When the height (north-south) of a region extent is greater than its width (east-west), you should use *Portrait* instead of *Landscape* orientation to optimize the use of paper. With a Data Defined Override button you can dynamically set the paper orientation.

Right-click on the page and select *Page Properties* to open the panel. We want to set the orientation dynamically, using an expression depending on the region geometry, so press the button of field *Orientation*, select *Edit…* to open the *Expression string builder* dialog and enter the following expression:

```
CASE WHEN bounds_width(@atlas_geometry) > bounds_height(@atlas_geometry)
THEN 'Landscape' ELSE 'Portrait' END
```

Now if you *preview the atlas*, the paper orients itself automatically, but item placements may not be ideal. For each Region you need to reposition the location of the layout items as well. For the map item you can use the button of its *Width* property to set it dynamic using the following expression:

```
@layout_pagewidth - 20
```

Likewise, use the button of the *Height* property to provide the following expression to constrain map item size:

```
@layout_pageheight - 20
```

To ensure the map item is centered in the page, set its *Reference point* to the upper left radio button and enter `10` for its *X* and *Y* positions.

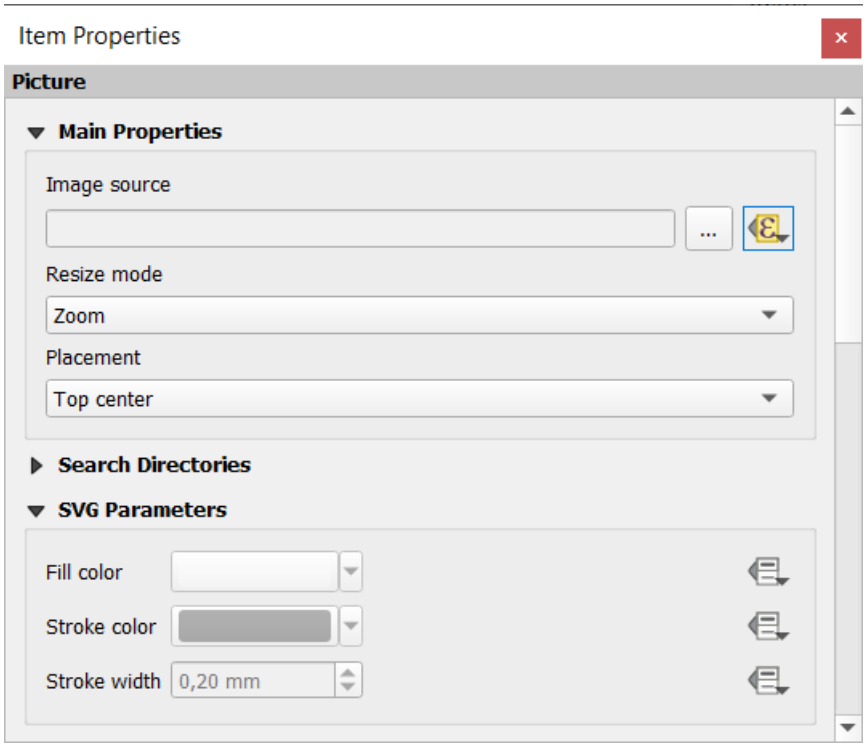
Let's add a title above the map in the center of the page. Select the label item and set the horizontal alignment to *Center*. Next move the label to the right position, choose the middle button for the *Reference point*, and provide the following expression for field *X*:

```
@layout_pagewidth / 2
```

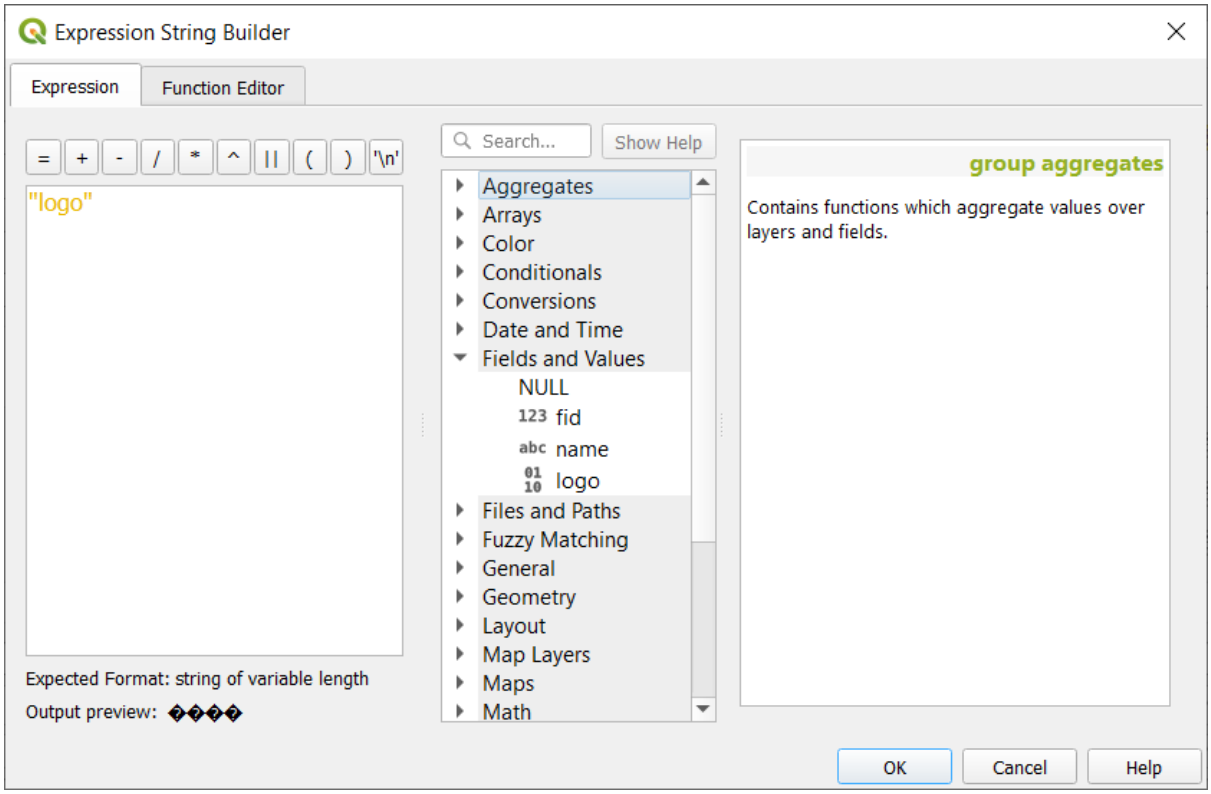
For all other layout items you can set the position in a similar way so they are correctly positioned both for portrait and landscape. You can also do more tweaks such as customizing the title with feature attributes (see *Customizar rótulos com expressão* example), changing images, resizing the number of legend columns number according to page orientation, …

The information provided here is an update of the excellent blog (in English and Portuguese) on the Data Defined Override options Multiple_format_map_series_using_QGIS_2.6 .

Another example for using data-defined override buttons is the usage of a dynamic picture. For the following examples we use a geopackage layer containing a BLOB field called `logo` with the field type binary (see *Criando uma nova camada GeoPackage*). For every feature there is defined a different picture so that the atlas can iterate over as described in *Pré-visualizar e gerar um atlas*. All you need to do is add a picture in the print layout and go to its *Item properties* in the atlas context. There you can find a data-defined override button in the *Image source* section of the *Main Properties*.



In the following window choose *Edit* so that the *Expression String Builder* opens. From the *Fields and values* section you can find the BLOB field that was defined in the geopackage layer. Double-click the field name `logo` and click *OK*.



The atlas iterates over the entries in the BLOB field provided that you choose the geopackage layer as *Coverage layer* (further instructions you can find in *Pré-visualizar e gerar um atlas*).

Estes são apenas dois exemplos de como você pode usar algumas configurações avançadas com o atlas.

Pré-visualizar e gerar um atlas

Figura18.69: Barra de ferramentas de pré-visualização do Atlas

Once the atlas settings have been configured, and layout items (map, table, image…) linked to it, you can create a preview of all the pages by choosing *Atlas* ► *Preview Atlas* or clicking the Preview Atlas icon. You can then use the arrows to navigate through all the features:

- First feature
- Previous feature
- Next feature
- Last feature

You can also use the combo box to select and preview a specific feature. The combo box shows atlas feature names according to the expression set in the atlas *Page name* option.

As for simple compositions, an atlas can be generated in different ways (see *Criando um arquivo de Saída* for more information - just use tools from the *Atlas* menu or toolbar instead of the *Layout* menu.

This means that you can directly print your compositions with *Atlas* ► *Print Atlas*. You can also create a PDF using *Atlas* ► *Export Atlas as PDF…*: You will be asked for a directory to save all the generated PDF files, except if the *Single file export when possible* has been selected. In that case, you'll be prompted to give a filename.

With *Atlas* ► *Export Atlas as Images…* or *Atlas* ► *Export Atlas as SVG…* tool, you're also prompted to select a folder. Each page of each atlas feature composition is exported to the image file format set in *Atlas* panel or to SVG.

Nota

With multi-page output, an atlas behaves like a layout in that only the page that contains the *Configurações Gerais* will get a world file (for each feature output).

Dica

Imprimir uma feição específica do atlas*

If you want to print or export the composition of only one feature of the atlas, simply start the preview, select the desired feature in the drop-down list and click on *Layout* ► *Print* (or *Export…* to any supported file format).

Utilizar as relações definidas pelo projeto para a criação do atlas

For users with HTML and Javascript knowledge it is possible to operate on GeoJSON objects and use project defined relations from the QGIS project. The difference between this approach and using expressions directly inserted into the HTML is that it gives you a full, unstructured GeoJSON feature to work with. This means that you can use existing Javascript libraries and functions that operate on GeoJSON feature representations.

The following code includes all related child features from the defined relation. Using the JavaScript `setFeature` function it allows you to make flexible HTML which represents relations in whatever format you like (lists, tables, etc). In the code sample, we create a dynamic bullet list of the related child features.

```
// Declare the two HTML div elements we will use for the parent feature id
// and information about the children
<div id="parent"></div>
<div id="my_children"></div>

<script type="text/javascript">
   function setFeature(feature)
   {
     // Show the parent feature's identifier (using its "ID" field)
     document.getElementById('parent').innerHTML = feature.properties.ID;
     //clear the existing relation contents
     document.getElementById('my_children').innerHTML = '';
     feature.properties.my_relation.forEach(function(child_feature) {
     // for each related child feature, create a list element
     // with the feature's name (using its "NAME" field)
       var node = document.createElement("li");
       node.appendChild(document.createTextNode(child_feature.NAME));
       document.getElementById('my_children').appendChild(node);
     });
   }
</script>
```

During atlas creation there will be an iteration over the coverage layer containing the parent features. On each page, you will see a bullet list of the related child features following the parent's identifier.

18.4 Criando um Relatório

Esta seção irá ajudá-lo a configurar um relatório no QGIS.

18.4.1 O que é?

Por definição, um relatório SIG é um documento contendo informações organizadas de forma narrativa, contendo mapas, textos, gráficos, tabelas, etc. Um relatório pode ser preparado ad hoc, periódico, recorrente, regular, ou conforme necessário. Os relatórios podem se referir a períodos, eventos, ocorrências, assuntos ou locais específicos.

No QGIS, um *Relatório* é uma extensão de um *Layouts*.

Os relatórios permitem que os usuários gerem seus projetos GIS de maneira simples, rápida e estruturada.

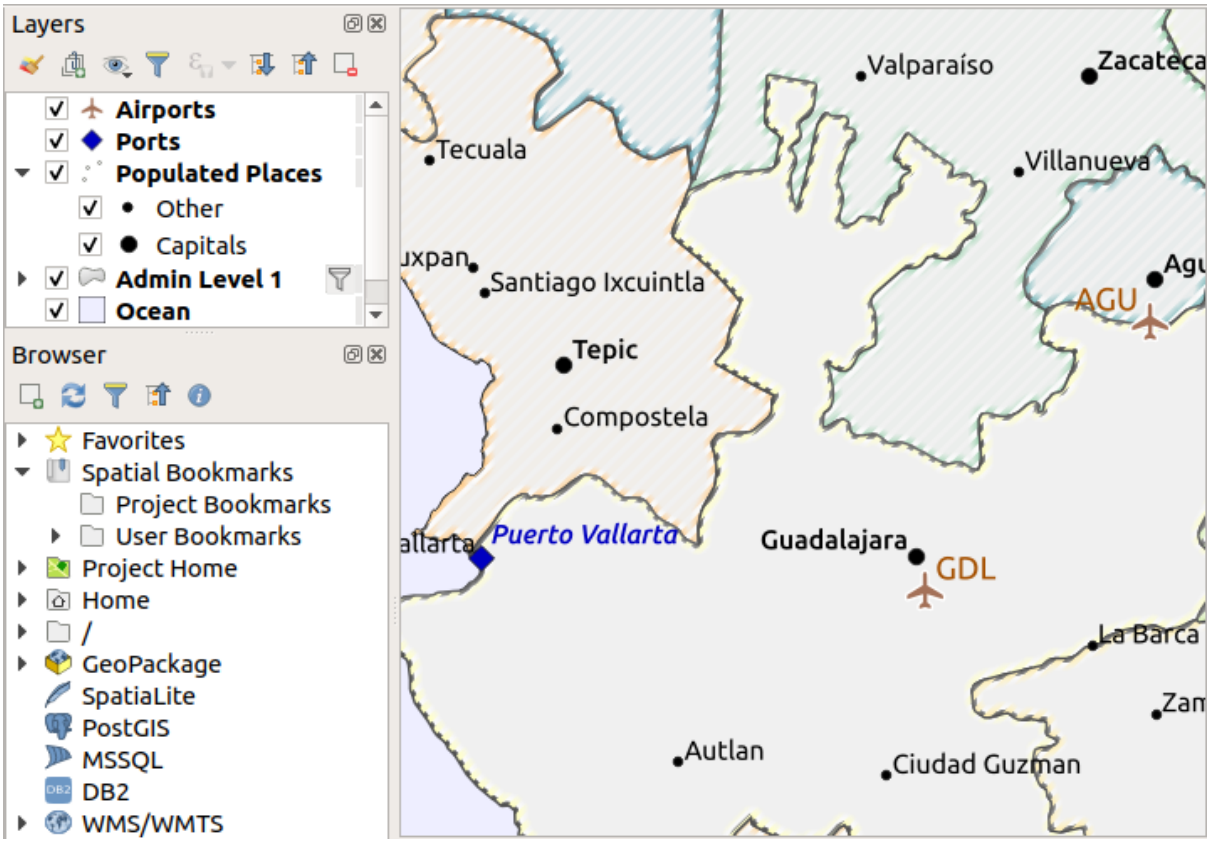
A report can be created with *Project* ► *New Report* or inside the *Project* ► *Layout Manager*.

> **Nota**
>
> Os mapas nos relatórios QGIS comportam-se da mesma forma que os mapas nos layouts e atlas impressos. Vamos nos concentrar nas especificidades dos relatórios QGIS. Para detalhes sobre manuseio de mapas, veja as seções sobre :ref:` layouts de impressão <overview_layout>` e *atlases*.

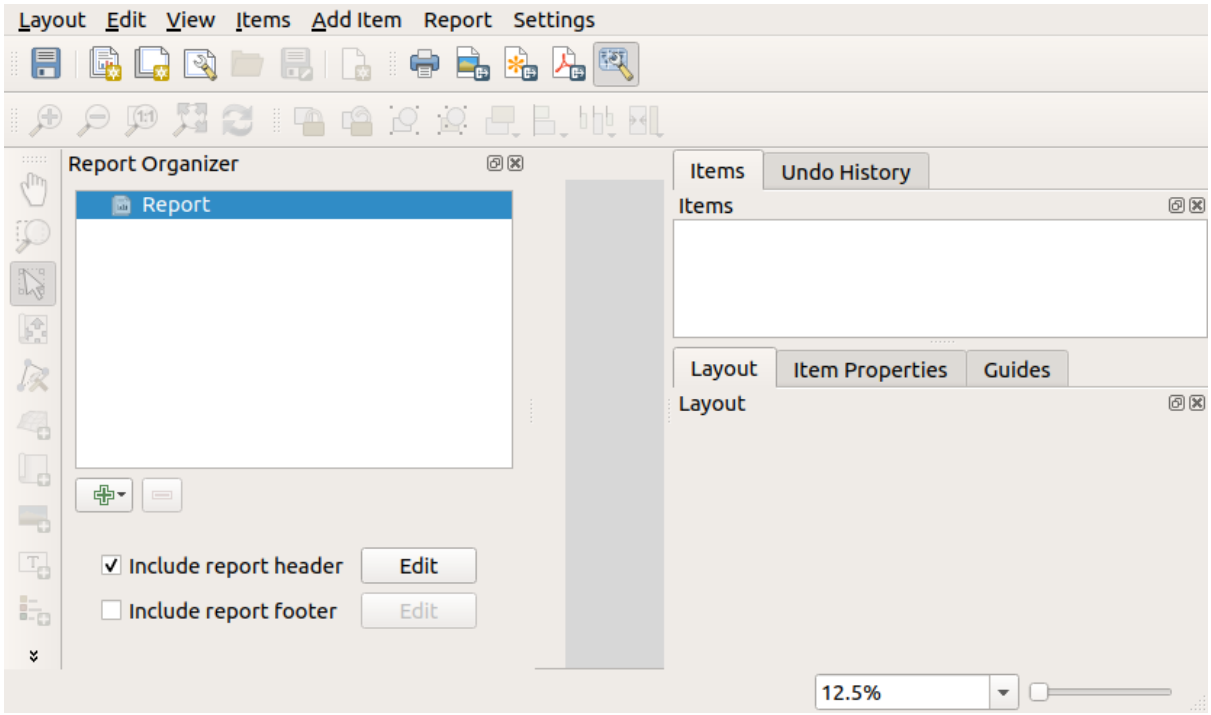
18.4.2 Iniciando

In the *Layout Manager* dialog a report can be created through *New from template* by selecting the dropdown option *Empty Report* and hitting the *Create…* button.

Para este exemplo, usamos alguns limites administrativos, locais com população, portos e aeroportos do Natural Earth dataset (1: 10M).

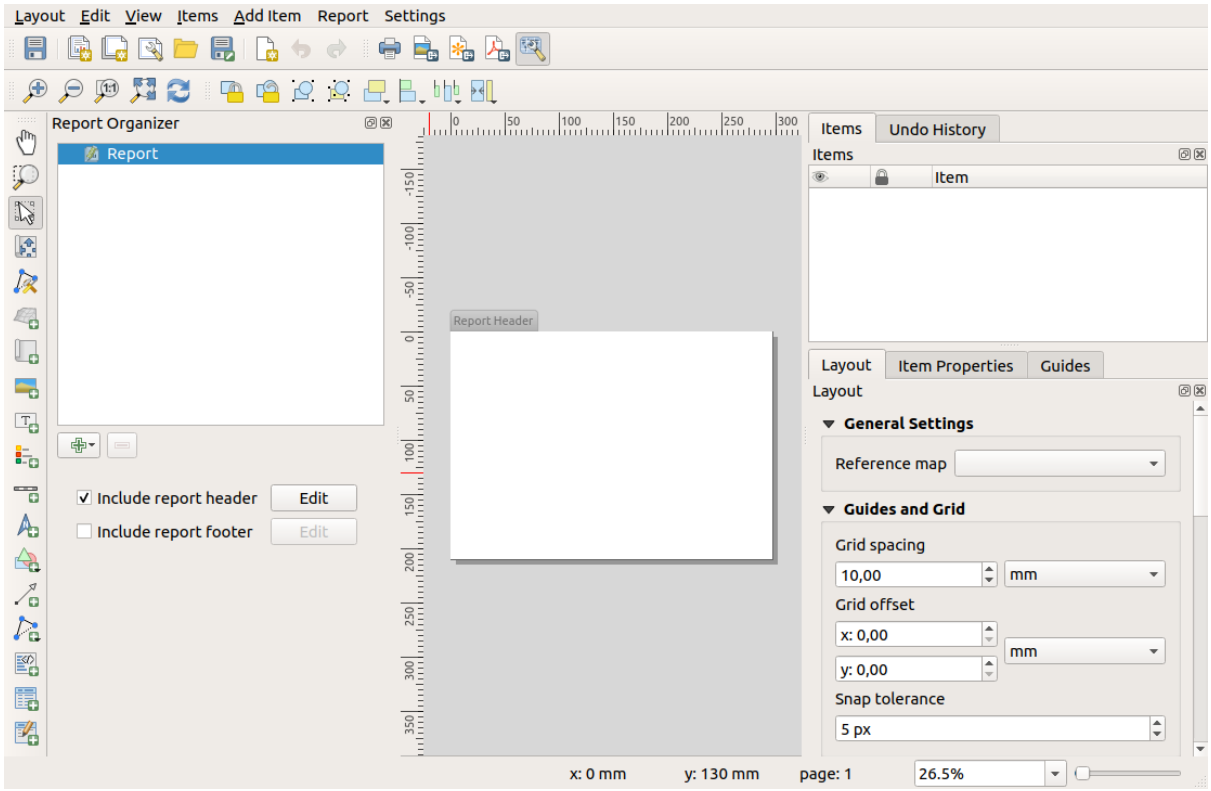


Usando o comando :menuselection:`Projeto –> Novo Relatório', criamos um relatório em branco. Inicialmente, não há muito para ver - o diálogo que é exibido se parece muito com o designer do layout de impressão, exceto pelo painel :guilabel:`Organizador de relatório' à esquerda:



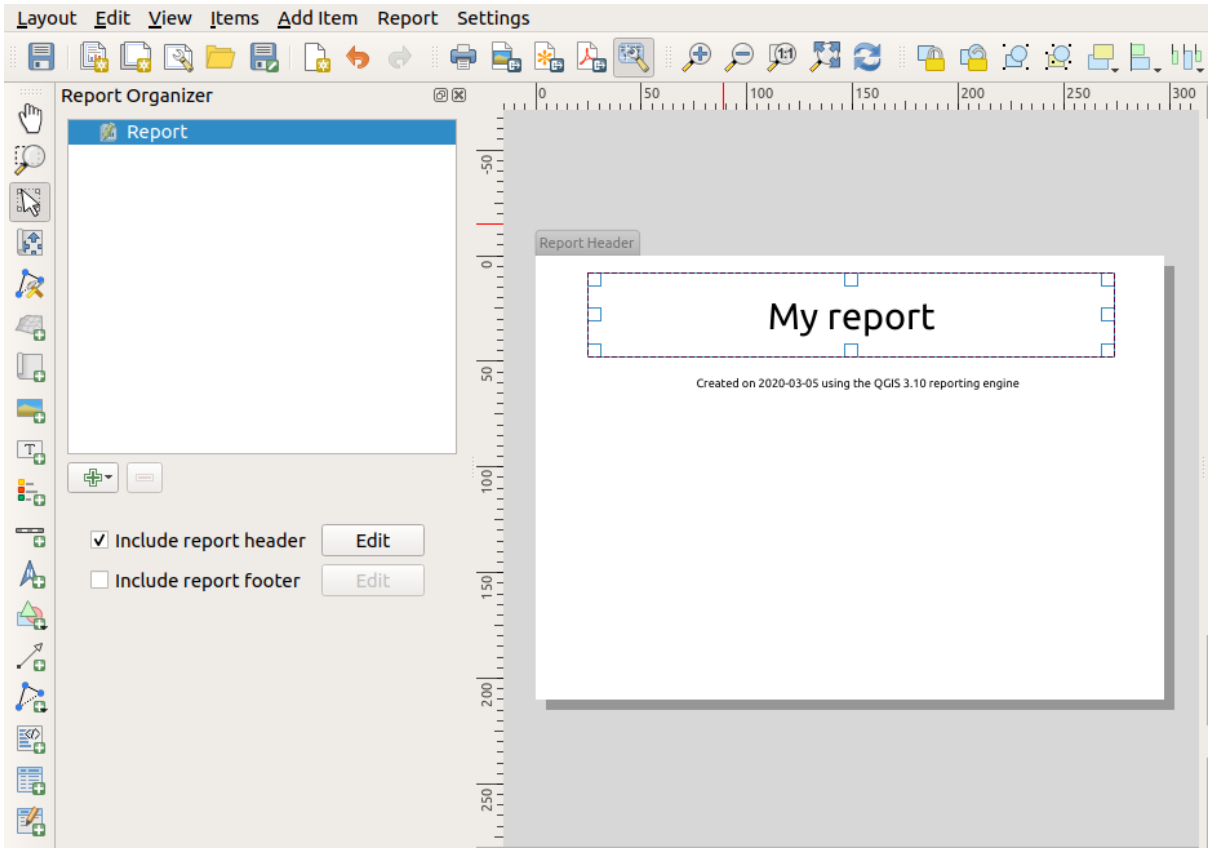
18.4.3 Espaço de Trabalho do Compositor de Relatório

QGIS reports can consist of multiple, nested sections. In our new blank report we initially only have the main report section. The only options for this report section is *Include report header* and *Include report footer*. If we enable these options, a header will be included as the first page(s) (individual parts of reports can be multi-page if desired) in the report, and a footer will constitute the last page(s). Enable the header (*Include report header*), and hit the *Edit* button next to it:

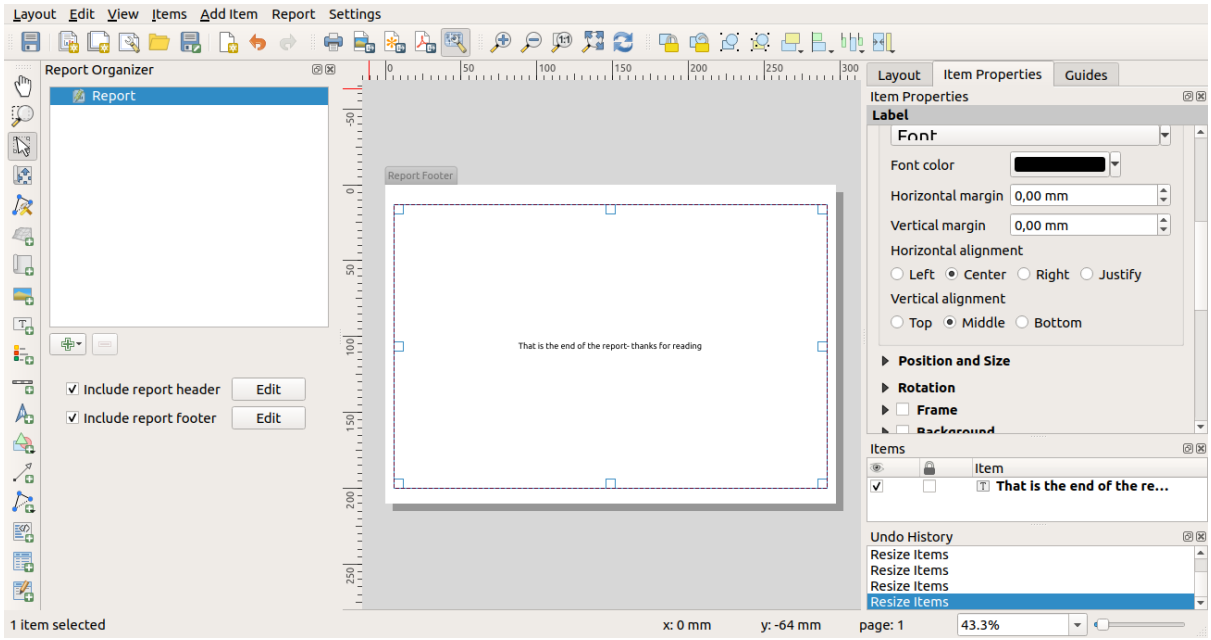


A few things happen as a result. Firstly, an edit pencil is shown next to *Report* in the *Report Organizer*, indicating that the report section is currently being edited in the designer. We also see a new page with a small *Report Header* title. The page has *landscape* orientation by default, but this (and other properties of the page) can be changed by right-clicking on the page and choosing *Page properties*. This will bring up the *Item properties* tab for the page, and page *Size*, *Width*, *Height*, and more can be specified.

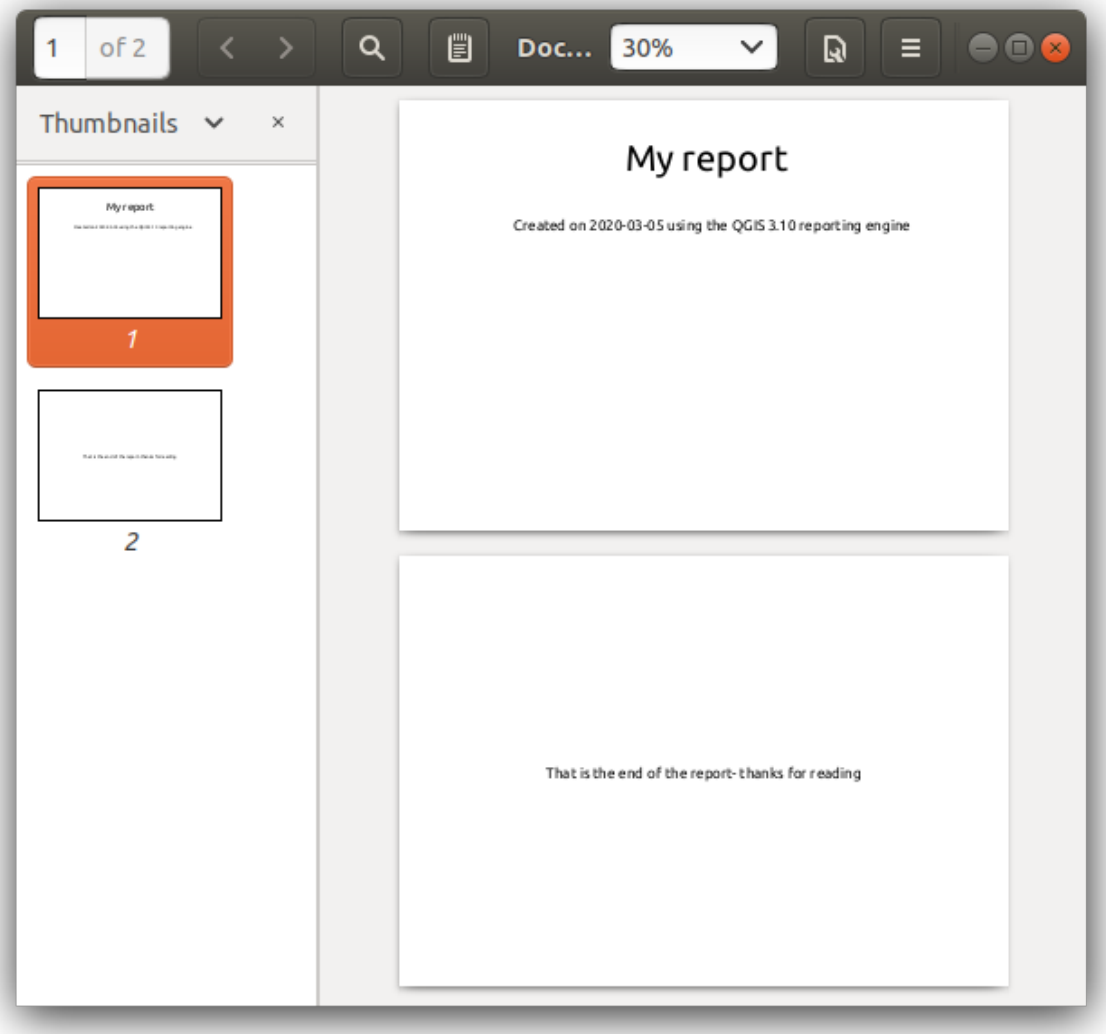
Nos relatórios QGIS, todos os componentes do relatório são compostos por layouts individuais. Eles podem ser criados e modificados usando as mesmas ferramentas dos layouts de impressão padrão - para que você possa usar qualquer combinação desejada de rótulos, figuras, mapas, tabelas etc. Vamos adicionar alguns itens ao cabeçalho do nosso relatório para demonstrar:



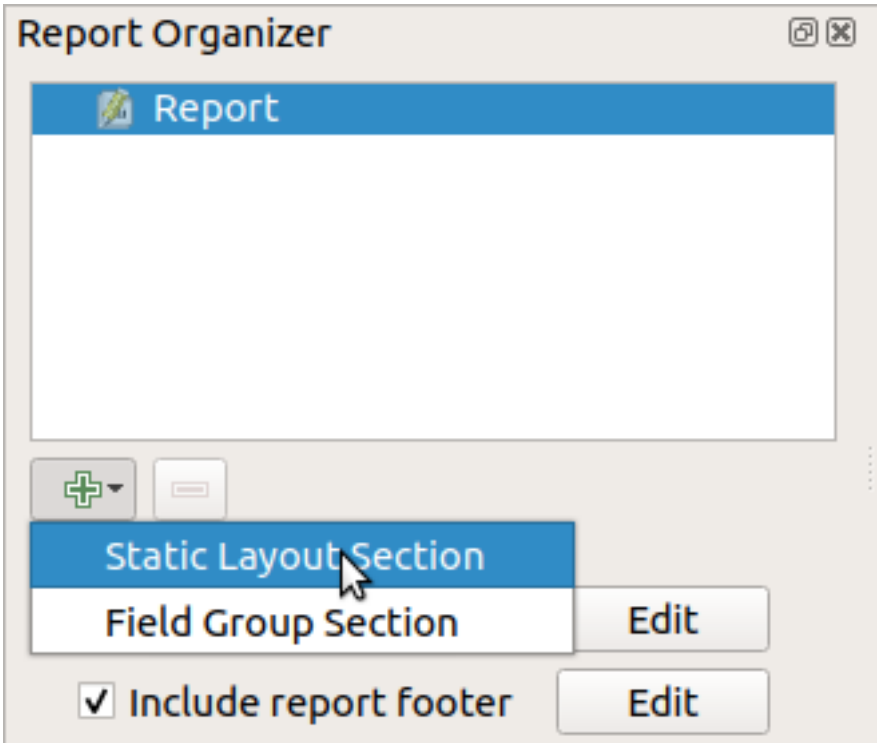
Também criaremos um rodapé simples para o relatório, verificando a opção :guilabel:`incluir rodapé do relatório' e selecionando :guilabel:`Editar'.



Antes de continuarmos, vamos exportar este relatório e ver o que conseguimos. A exportação é feita a partir do menu *Relatório' - neste caso selecionamos :menuselection:`Exportar Relatório como PDF…* para renderizar o relatório inteiro em um arquivo PDF. Aqui está o resultado não muito impressionante - um PDF de duas páginas que consiste de nosso cabeçalho e rodapé:



Let us make things more interesting. By hitting the Add Section button in the *Report Organizer*, we are given a choice of new sections to add to our report.

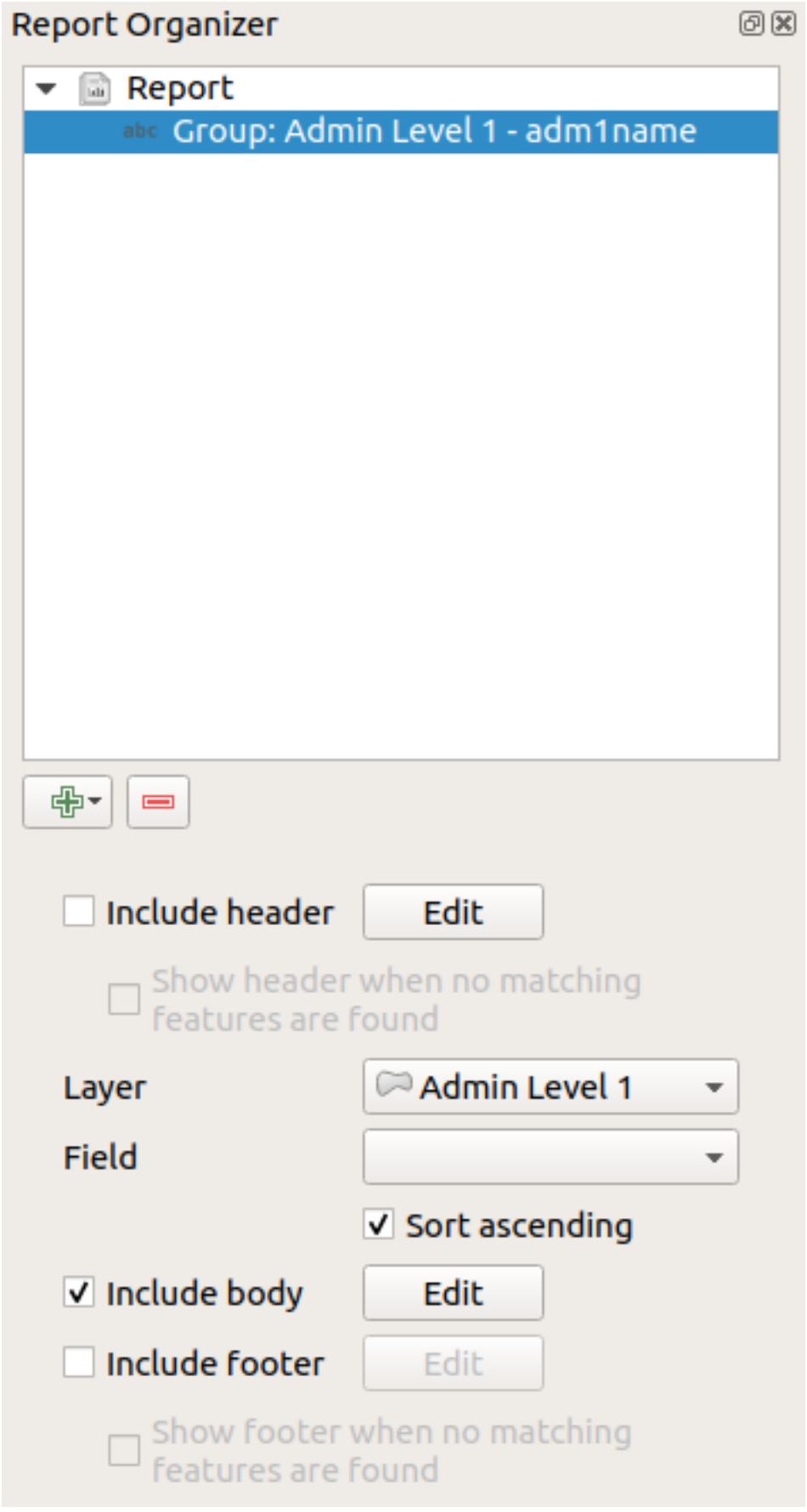


There are two options: *Static Layout Section* and *Field Group Section*.

The *Add Static Layout Section* is a single, static body layout. This can be used to embed static layouts mid-way through a report.

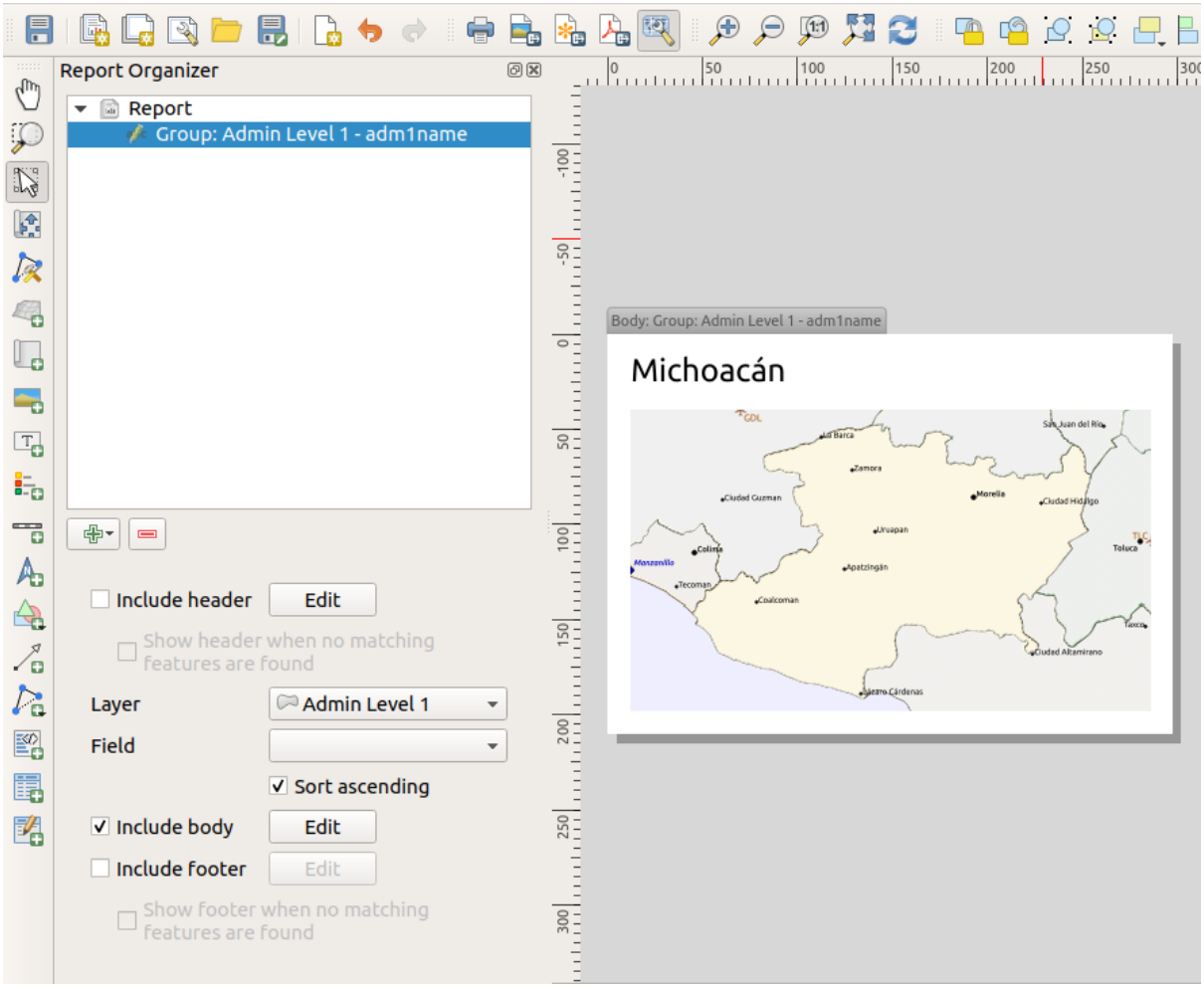
The *Field Group Section* repeats its body layout for every feature of a layer. The features are sorted by the selected grouping feature (with an option for ascending/descending sort). If a field group section has child sections (e.g. another field group section with a different field), then only features with unique values for the group feature are iterated over. This allows nested reports.

For now we will add a *Field Group Section* to our report. At its most basic level, you can think of a *Field Group Section* as the equivalent of a *print atlas*: you select a layer to iterate over, and the report will insert a section for each feature found. Selecting the new *Field Group Section* reveals a number of new related settings:



In this case we've setup our Field Group so that we iterate over all the states from the *Admin Level 1* layer, using the

values from the *adm1name* field. The same options to include header and footer are present, together with a new option to include a *body* for this section. We'll do that, and edit the body:

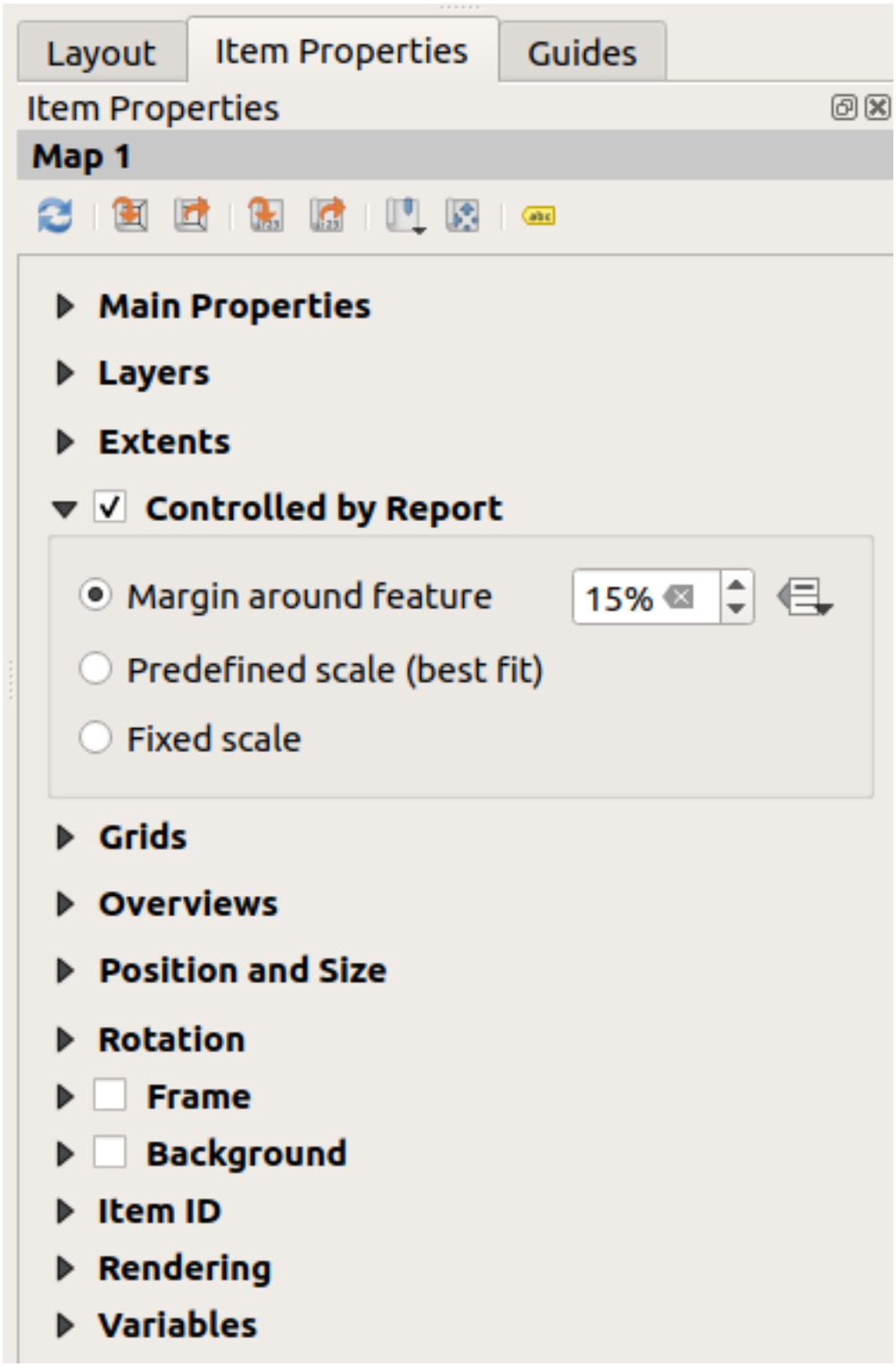


Our body now consists of a map and a label showing the name of the state. To include the name of the state, we selected *Add Item* ► *Add Label* and data defined the text under *Main Properties* with the help of *Insert or Edit an Expression*….

The result was the following expression (*name* is the name of the attribute in the *Admin Level 1* layer that contains the name of the state):

```
[% "name" %]
```

The map is set to follow the current report feature (enabled by checking *Controlled by Report* – just like a map item in an atlas will follow the current atlas feature when *Controlled by Atlas* is checked):



Se fôssemos adiante e exportássemos nosso relatório agora, conseguiríamos algo assim:

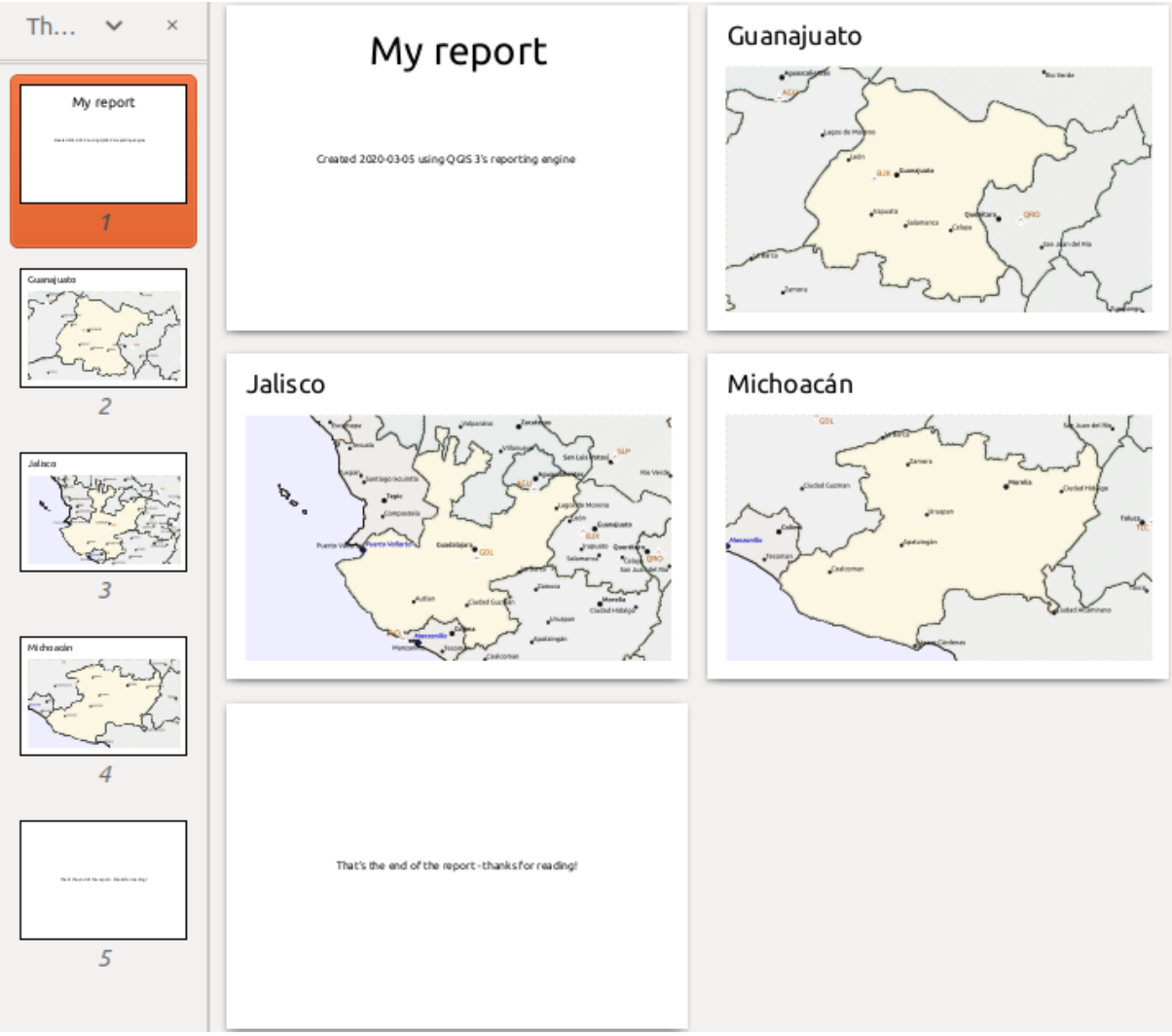
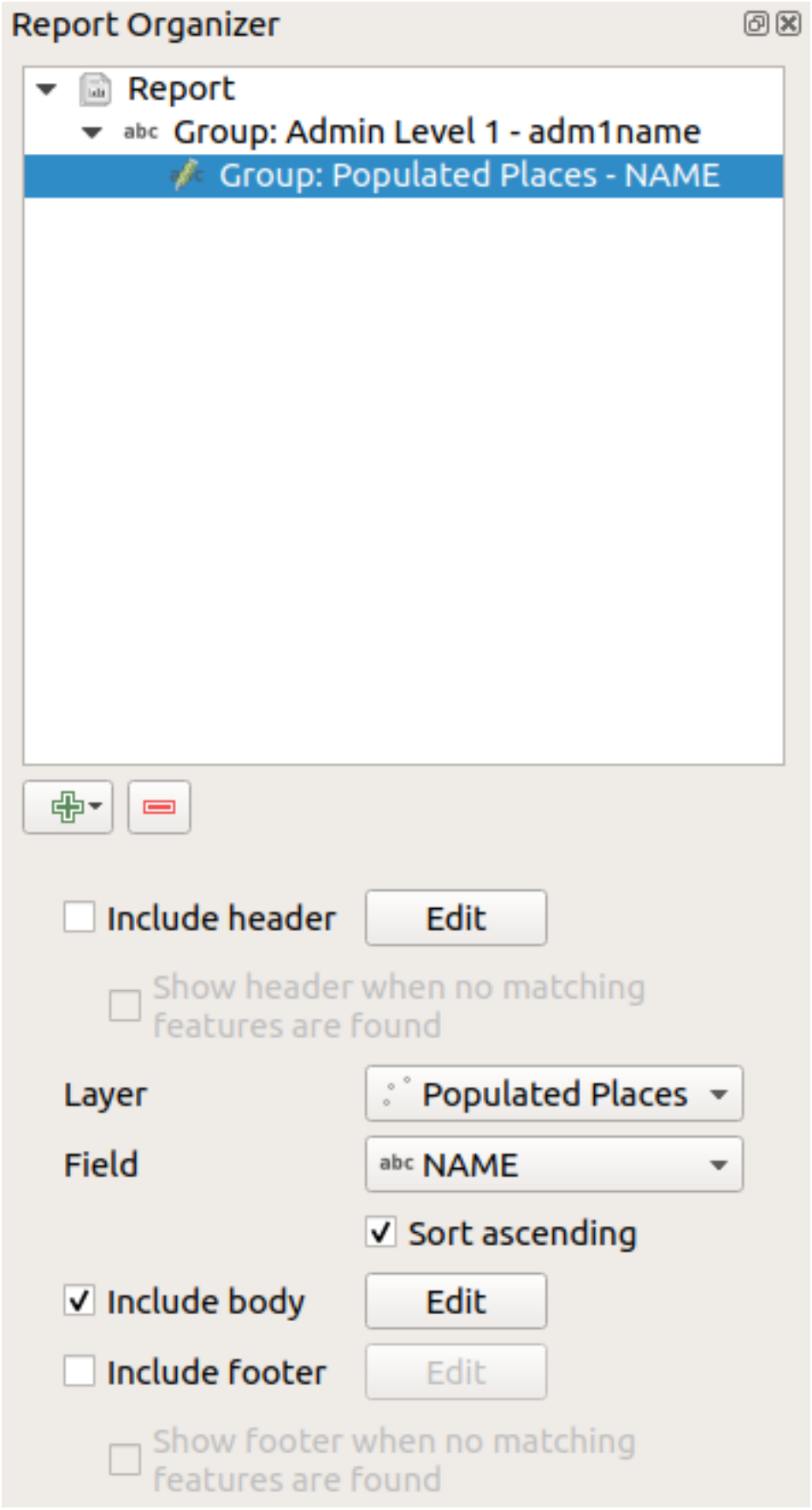


Figura18.70: O cabeçalho do relatório, uma página para cada estado, e o rodapé do relatório.

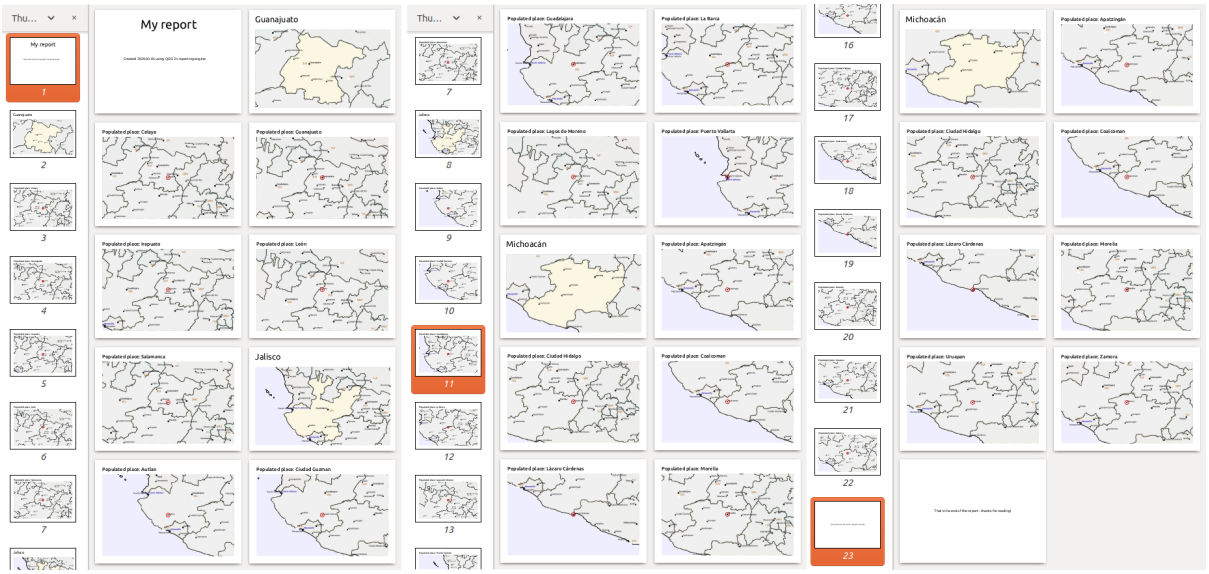
Portanto, mais ou menos um atlas, mas com uma página de cabeçalho e rodapé.

Vamos tornar as coisas mais interessantes, acrescentando uma subseção ao nosso grupo estadual. Fazemos isto primeiro selecionando o grupo de campo *Admin Número 1* no organizador, depois pressionando o botão Adicionar campo e adicionando uma nova *Seção de grupos de campo*:



When iterating over the features of a *Field Group Section*, the features will be filtered to match the defining field

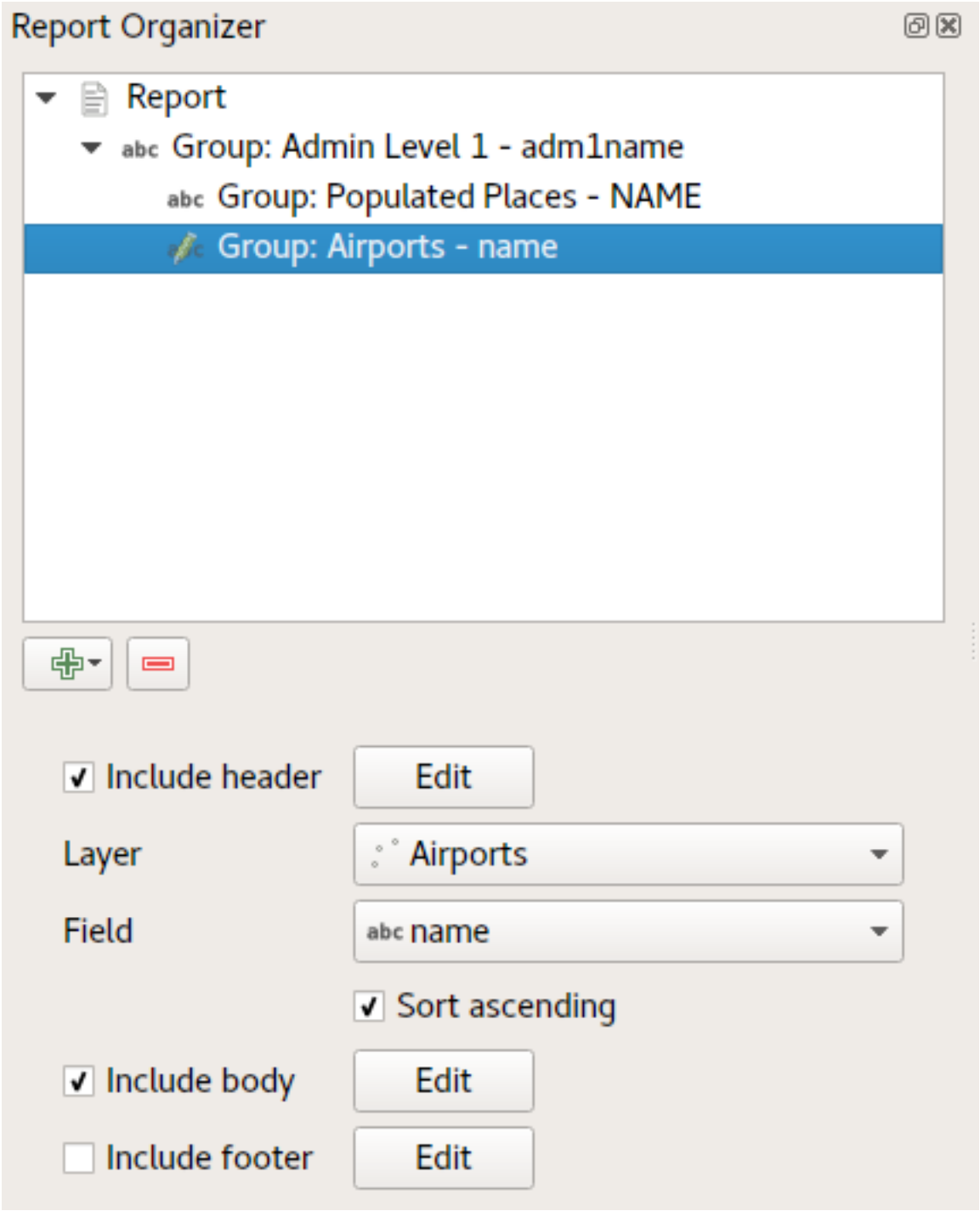
of its parent group (`adm1name` in this case). Here, the subsection we added will iterate over a *Populated Places* layer, including a body section for each place encountered. The magic here is that the *Populated Places* layer has an attribute with the same name as the defining field in the parent layer, *adm1name*, tagging each place with the state it is contained within (if you're lucky your data will already be structured like this – if not, run the *Join Attributes by Location* Processing algorithm and create your own field). When we export this report, QGIS will grab the first state from the *Admin Level 1* layer, and then iterate over all the *Populated Places* with a matching *adm1name* value. Here's what we get:



Here we created a basic body for the Populated Places group, including a map of the place and a table of some place attributes. So our report is now a report header, a page for the first state, followed by a page for every populated place within that state, then the rest of the states with their populated places, and finally the report footer. If we were to add a header for the Populated Places group, it would be included just before listing the populated places for each state, as shown in the illustration below.

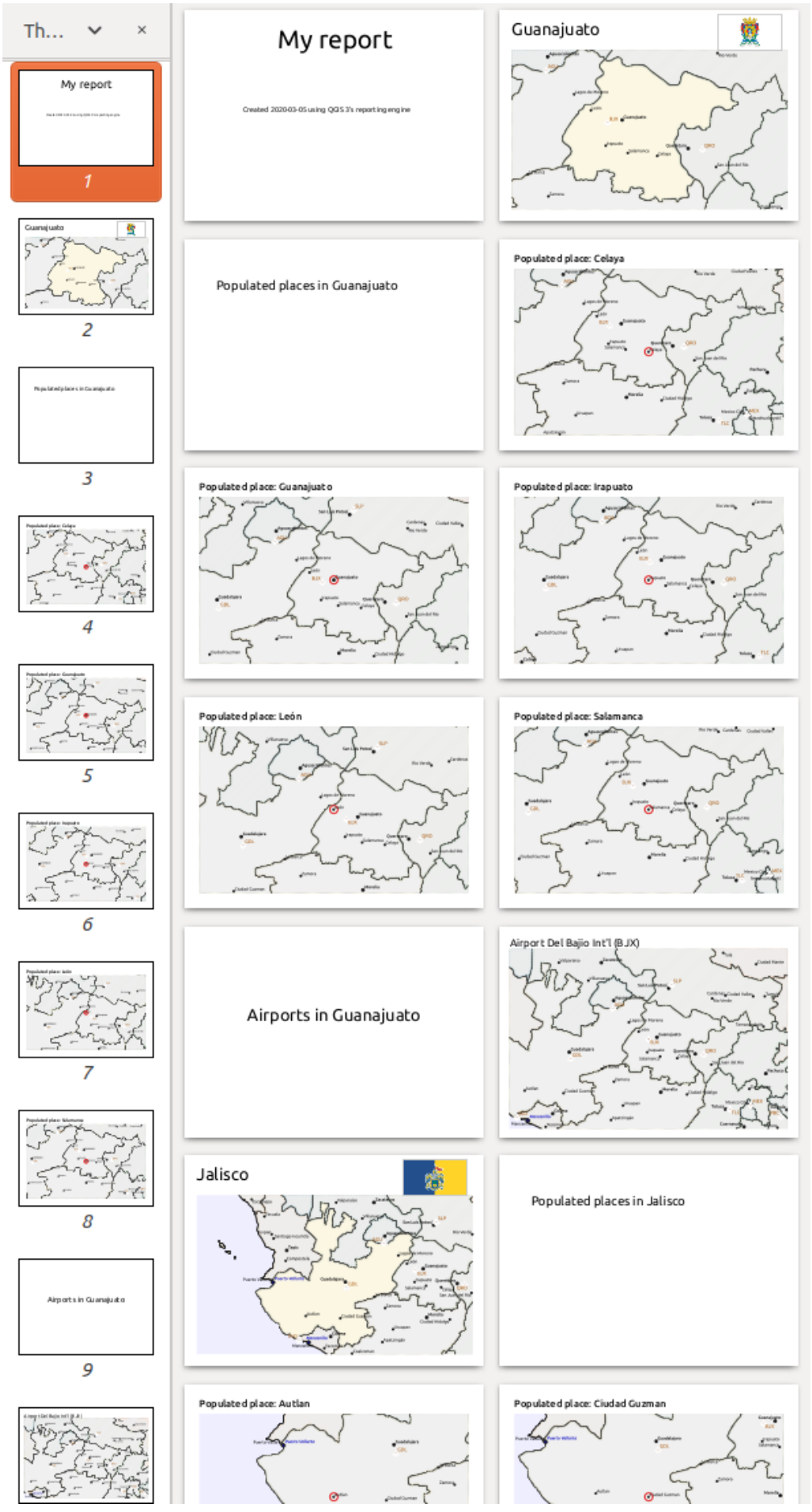
Da mesma forma, um rodapé para o grupo de Lugares Populados seria inserido após o local final para cada estado ser incluído.

In addition to nested subsections, subsections in a report can also be included consecutively. If we add a second subsection to the *Admin Level 1 group* for *Airports*, then (if the *Airports* layer has an attribute *adm1name* that can link it to the parent group) our report will first list ALL the populated places for each state, followed by all the airports within that state, before proceeding to the next state.



The key point here is that our *Airports group* is a subsection of the *Admin Level 1 group* – not the *Populated Places group*.

In this case our report would be structured like this (note that state flags have also been included - the procedure for adding feature specific pictures in this way is described below):


Th...
My report
1
2
3
4
5
6
7
8
9
My report
Created 2020-03-05 using QGIS 3's reporting engine
Guanajuato
Populated places in Guanajuato
Populated place: Celaya
Populated place: Guanajuato
Populated place: Irapuato
Populated place: León
Populated place: Salamanca
Airports in Guanajuato
Airport Del Bajio Int'l (BJX)
Jalisco
Populated places in Jalisco
Populated place: Autlan
Populated place: Ciudad Guzman

Incluindo imagens em um relatório

Pictures can be quite useful in reports, and QGIS allows pictures in both the static and dynamic parts of a report. Pictures are added in the same way as for standard print layouts, and for the static report parts (and static pictures in dynamic parts) there is not more to it.

Mas se você quiser ilustrações que sejam adaptadas às características do relatório, sua camada deve ter um atributo que possa ser usado para definir a imagem a incluir.

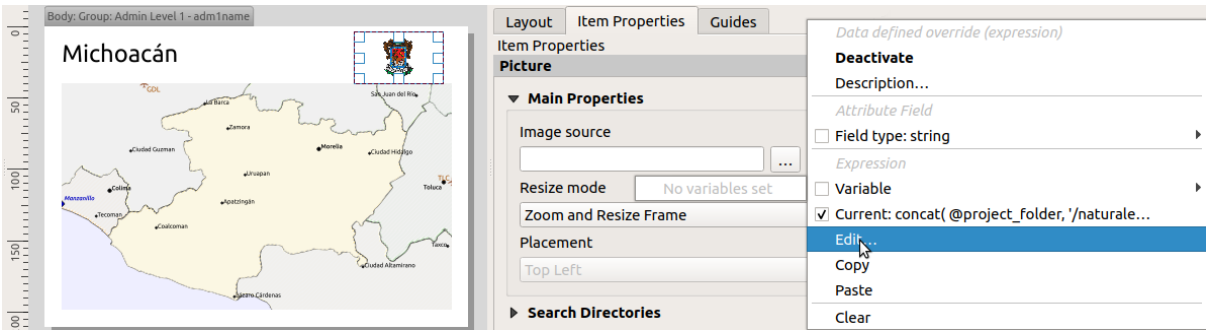
O QGIS depende dos nomes absolutos dos arquivos de imagens nos relatórios.

Para imagens dinâmicas, você primeiro adiciona uma imagem à parte do corpo do grupo, como de costume. No *Propriedades do Item* da imagem, você define *Fonte da Imagem* usando o botão Substituição definida dos dados e selecione um atributo que contenha o caminho absoluto das imagens ou :guilabel:`Editar …`(para inserir uma expressão que gere o caminho absoluto da imagem).

Below is an example expression that uses string concatenation to specify the absolute path to the pictures, using the directory where the project file is located (`@project_path`) and an attribute (`adm1name`) from which the file name is generated (in this case by transforming the string in the `adm1name` attribute to uppercase, and appending '_flag.png'):

```
concat(@project_folder, '/naturalearth/pictures/' ,
       upper("adm1name"), '_flag.png')
```

This means that the pictures are located in the `naturalearth/pictures` subdirectory of the project file directory.



Destacando a característica do relatório atual em um mapa

In the above report, the report features are emphasized in the maps using highlighting (state) and circles (populated places). To emphasize the report features in the maps (apart from placing them at the centre of the maps), you must data define the style using a comparison between its `@id` and the `@atlas_featureid`, as for atlases.

For instance, if you would like to use a thicker line / border for the report feature than the other features you can data define the line width:

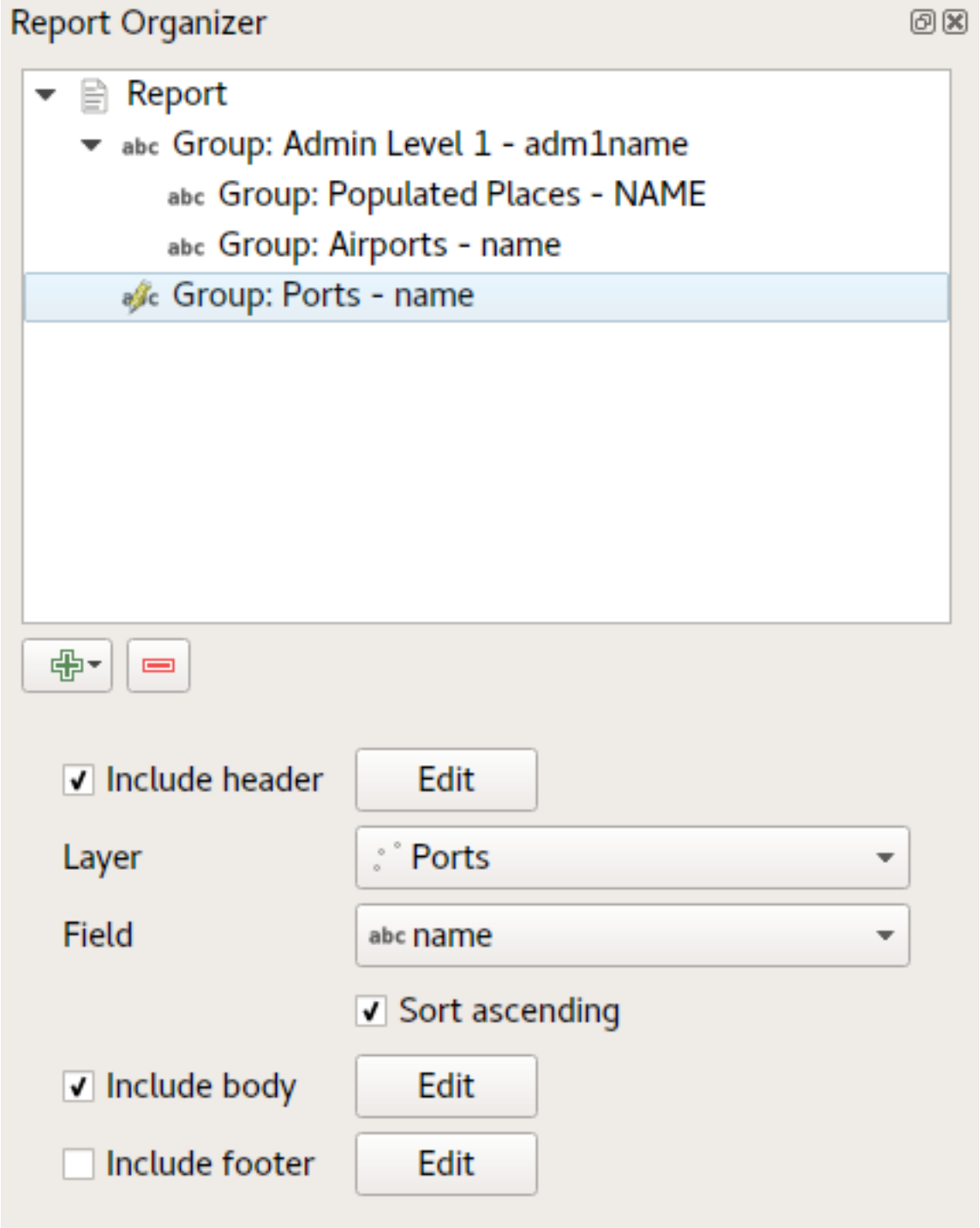
```
if($id=@atlas_featureid, 2.0, 0.1)
```

The report feature will get a 2 units wide polygon outline, while all other features will get a 0.1 units wide line. It is also possible to data define the colour (non-transparent dark magenta for the report feature and semi-transparent light gray for the other features):

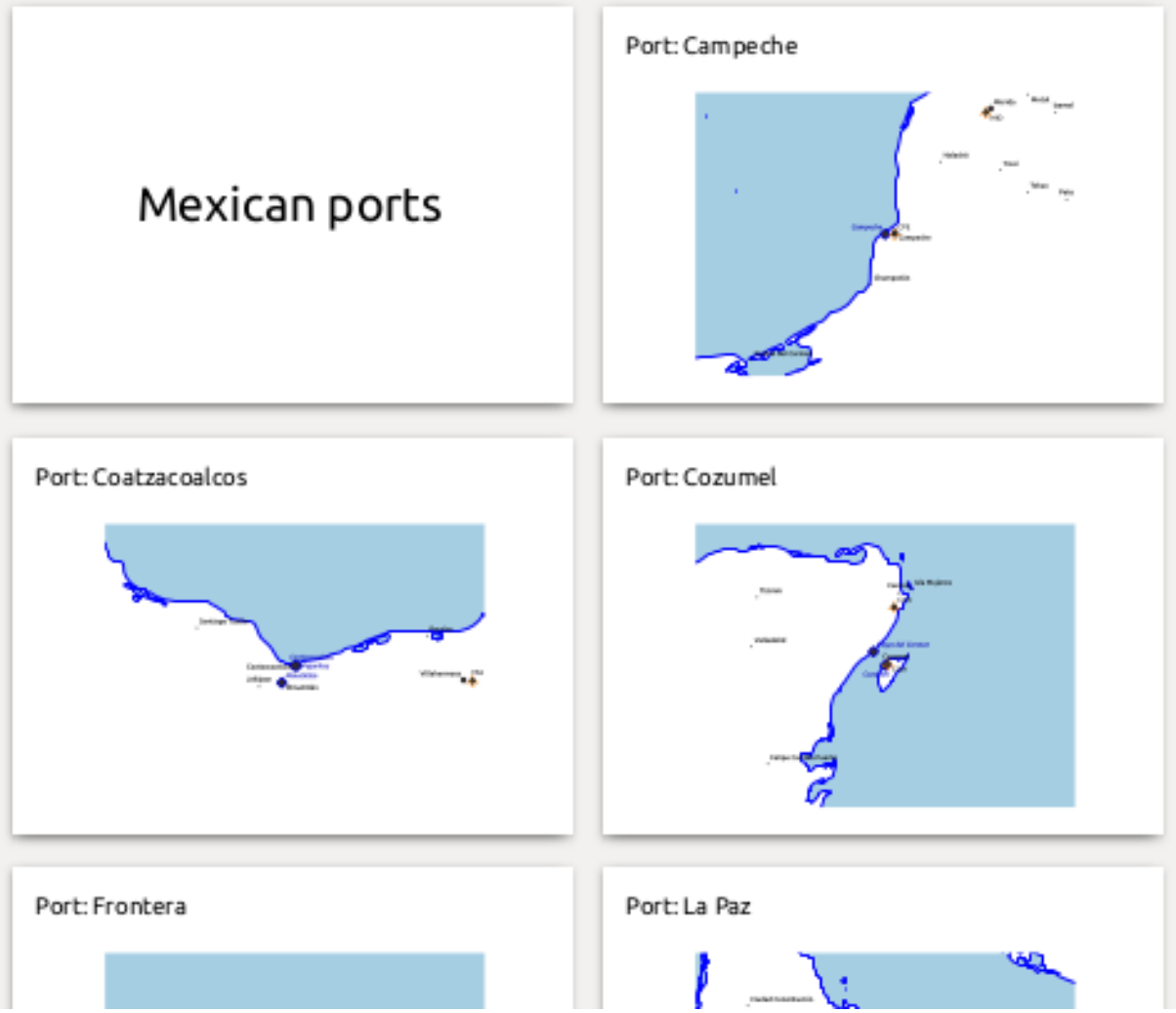
```
if($id=@atlas_featureid, '#FF880088', '#88CCCCCC')
```

Mais grupos de nível 1

Combining nested and consecutive sections, together with section headers and footers allows for tons of flexibility. For instance, in the below report we add another field group as a child of the main report for the *Ports* layer. Now, after listing the states together with their populated places and airports, we'll get a summary list of all the ports in the region:



Isto resulta na última parte de nosso relatório exportando como:



18.4.4 Definições para exportação

Quando você exporta um relatório (:menuselection:` Relatório –> Exportar Relatório como Imagens… / SVG… / PDF…`), será solicitado um nome de arquivo, e então você terá a oportunidade de ajustar as configurações de exportação para obter a saída mais apropriada.

Como você vê, os relatórios no QGIS são extremamente poderosos e flexíveis!

Nota

The current information was adapted from a North Road blog, Exploring Reports in QGIS 3.0 - the Ultimate Guide!

CAPÍTULO 19

Working with OGC / ISO protocols

O Consórcio Geoespacial Aberto (OGC), é uma organização internacional como mais de 300 organizações em todo mundo do tipo comerciais, sem fins lucrativos e de investigação. Os membros desenvolvem e implementam padrões para os conteúdos e serviços geoespaciais, processamento e troca de dados SIG.

Describing a basic data model for geographic features, an increasing number of specifications are developed by OGC to serve specific needs for interoperable location and geospatial technology, including GIS. Further information can be found at https://www.ogc.org/.

Importantes especificações OGC suportadas pelo QGIS são:

- **WMS** — Serviço de Mapas Web (*Cliente WMS/WMTS*)
- **WMTS** — Serviço de Mosaicos de Mapa Web (*Cliente WMS/WMTS*)
- **WFS** — Serviços de Elementos Web (*WFS e WFS-T Cliente*)
- **WFS-T** — Serviços de Elementos Web - Transacionais (*WFS e WFS-T Cliente*)
- **WCS** — Serviços de Cobertura Web (*WCS Cliente*)
- **SPW** — Serviço de Processamento Web
- **SCW** — Serviço de catálogo para a Web
- **SFS** — Elementos Simples para SQL (*PostgreSQL Database*)
- **GML** — Linguagem de Marcadores Geográfico
- **SensorThings API** — SensorThings API (*SensorThings*)

OGC services are increasingly being used to exchange geospatial data between different GIS implementations and data stores. QGIS can deal with the above specifications as a client, being **SFS** (through support of the PostgreSQL data provider, see section *PostgreSQL Database*).

You can also share your maps and data through the WMS, WMTS, WFS, WFS-T and WCS protocols using a web-server with QGIS Server, UMN MapServer or GeoServer installed.

19.1 Cliente WMS/WMTS

19.1.1 Visão Global do Suporte WMS

QGIS currently can act as a WMS client that understands WMS 1.1, 1.1.1 and 1.3 servers. In particular, it has been tested against publicly accessible servers such as DEMIS.

A WMS server acts upon requests by the client (e.g., QGIS) for a raster map with a given extent, set of layers, symbolization style, and transparency. The WMS server then consults its local data sources, rasterizes the map, and sends it back to the client in a raster format. For QGIS, this format would typically be JPEG or PNG.

WMS is generically a REST (Representational State Transfer) service rather than a full-blown Web service. As such, you can actually take the URLs generated by QGIS and use them in a web browser to retrieve the same images that QGIS uses internally. This can be useful for troubleshooting, as there are several brands of WMS server on the market and they all have their own interpretation of the WMS standard.

As camadas WMS podem ser adicionadas facilmente, desde que conheça o acesso URL para o servidor WMS, tenha uma ligação de serviço a esse servidor, e o servidor compreenda HTTP como um mecanismos de transporte de dados.

Additionally, QGIS will cache your WMS responses (i.e. images) for 24h as long as the GetCapabilities request is not triggered. The GetCapabilities request is triggered every time the *Connect* button in the *WMS/WMTS* dialog is used to retrieve the WMS server capabilities. This is an automatic feature meant to optimize project loading time. If a project is saved with a WMS layer, the corresponding WMS tiles will be loaded from the cache the next time the project is opened as long as they are not older than 24h.

19.1.2 Visão Global do Suporte WMTS

QGIS can also act as a WMTS client. WMTS is an OGC standard for distributing tile sets of geospatial data. This is a faster and more efficient way of distributing data than WMS because with WMTS, the tile sets are pre-generated, and the client only requests the transmission of the tiles, not their production. A WMS request typically involves both the generation and transmission of the data. A well-known example of a non-OGC standard for viewing tiled geospatial data is Google Maps.

De maneira a exibir os dados a várias escalas perto do que o utilizador queira ver, os conjuntos de mosaicos WMTS são produzidos em vários diferentes níveis de escala e são disponibilizados para o cliente SIG a seu pedido.

Este diagrama ilustra o conceito dos conjuntos de mosaicos

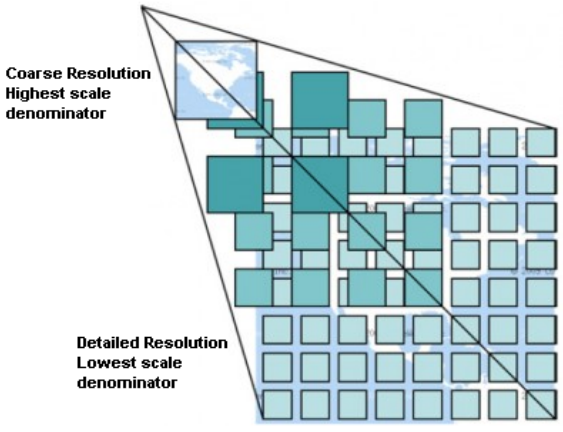


Figura19.1: Conceito dos conjuntos de mosaicos do WMTS

The two types of WMTS interfaces that QGIS supports are via Key-Value-Pairs (KVP) and RESTful. These two interfaces are different, and you need to specify them to QGIS differently.

1. In order to access a **WMTS KVP** service, a QGIS user must open the WMS/WMTS interface and add the following string to the URL of the WMTS tile service:

```
"?SERVICE=WMTS&REQUEST=GetCapabilities"
```

An example of this type of address is:

```
https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
↪request=GetCapabilities
```

A camada topo2 funciona lindamente para testar neste WMTS. Adicionando esta linha e texto indica que é um serviço web WMTS que deverá ser usado em vez de o serviço WMS.

2. O serviço **WMTS RESTful** torna-se uma forma diferente, é um URL simples, a forma recomendada pela OGC é:

```
{WMTSBaseURL}/1.0.0/WMTSCapabilities.xml
```

This format helps you to recognize that it is a RESTful address. A RESTful WMTS is accessed in QGIS by simply adding its address in the WMS setup in the URL field of the form. An example of this type of address for the case of an Austrian basemap is:

```
https://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml
```

Nota

You can still find some old services called WMS-C. These services are quite similar to WMTS (i.e., same purpose but working a little bit differently). You can manage them the same as you do WMTS services. Just add `?tiled=true` at the end of the url. See https://wiki.osgeo.org/wiki/Tile_Map_Service_Specification for more information about this specification.

Quando lê o WMTS pode muitas vezes pensar em WMS-C.

19.1.3 Selecionando os Servidores WMS/WMTS

The first time you use the WMS/WMTS feature in QGIS, there are no servers defined.

You then need to create connections to the server you are targeting:

1. Go to the *WMS/WMTS* tab of the *Data Source Manager* dialog, either by:
   - clicking the Open Data Source Manager button (or pressing `Ctrl+L`) and enabling the tab
   - clicking the Add WMS layer button on the *Manage Layers* toolbar
   - or selecting *Layer* ► *Add Layer* ► *Add WMS/WMTS Layer…* menu
2. Press *New* from the *Layers* tab. The *Create a New WMS/WMTS Connection…* dialog appears.

   Dica

   Right-click the *WMS/WMTS* entry from within the *Browser panel* and select *New Connection…* also opens the *Create a New WMS/WMTS Connection…* dialog.

3. Em seguida, digite os parâmetros para se conectar ao seu servidor WMS desejado, conforme listado abaixo:

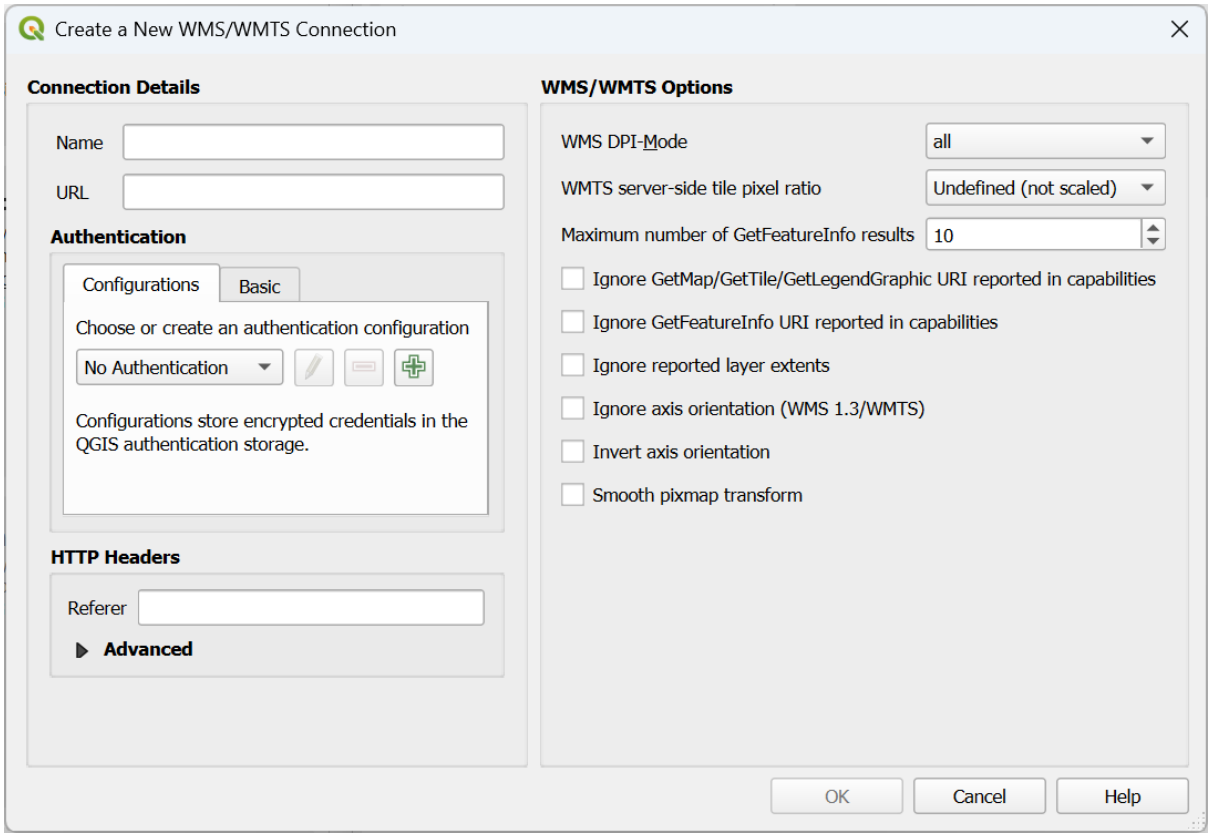


Figura19.2: Criando uma conexão com um servidor WMS

- *Name*: A name for the connection. This name will be used in the Server Connections drop-down box so that you can distinguish it from other WMS servers.
- *URL*: URL of the server providing the data. This must be a resolvable host name – the same format as you would use to open a telnet connection or ping a host, i.e. the base URL only. For example, you shouldn't have fragments such as `request=GetCapabilities` or `version=1.0.0` in your URL.
- *Authentication* (optional): using a *stored configuration* or a basic authentication with *Username* and *Password*.

> **Aviso**
>
> Entering **username** and **password** in the *Authentication* tab will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you shared the project file with someone. Therefore, it's advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab). See *Sistema de Autenticação* for more details.

- HTTP *Referência*
- *WMS DPI-Mode*: Available options are **all**, **off**, **QGIS**, **UMN** and **GeoServer**
- *WMTS server-side tile pixel ratio*: When rendering WMTS layers, allows to scale up or down the tiles based on the device screen DPI. Available options are *Undefined (not scaled)*, *Standard (96 DPI)* and *High (192 DPI)*.
- *Maximum number of GetFeatureInfo results*: specifies a default value for the maximum number of results returned per layer by a GetFeatureInfo request using this connection (see FEATURE_COUNT parameter). Default value is `10`. Set to `0` to use server default value (usually `1`): no FEATURE_COUNT parameter will be added to the request.

- ☐ *Ignore GetMap/GetTile/GetLegendGraphic URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore GetFeatureInfo URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore reported layer extents*: because the extent reported by raster layers may be smaller than the actual area which can be rendered (notably for WMS servers with symbology which takes more space than the data extent), check this option to avoid cropping raster layers to their reported extents, resulting in truncated symbols on the borders of these layers.
- ☐ *Ignorar a orientação dos eixos (WMS 1.3/WMTS)*
- ☐ *Inverter orientação do eixo*
- ☐ *Suavizar transformação do mapa de pixels*

4. Pressione *OK*

Once the new WMS/WMTS server connection has been created, it will be preserved for future QGIS sessions. Note that it is also possible to *Load* the connection parameters from a `.XML` file or *Save* them to a `.XML` file.

If you need to set up a proxy server to be able to receive WMS services from the internet, you can add your proxy server in the options. Choose *Settings* ► *Options* and click on the *Network* tab. There, you can add your proxy settings and enable them by setting ☑ *Use proxy for web access*. Make sure that you select the correct proxy type from the *Proxy type* [...▼] drop-down menu.

19.1.4 Carregando as camadas WMS/WMTS

Once you have successfully filled in your parameters, you can use the *Connect* button to retrieve the capabilities of the selected server. This includes the image encoding, layers, layer styles and projections. Since this is a network operation, the speed of the response depends on the quality of your network connection to the WMS server. While downloading data from the WMS server, the download progress is visualized in the lower left corner of the main QGIS dialog.

Your screen should now look a bit like Fig.19.3, which shows the response provided by a WMS server.

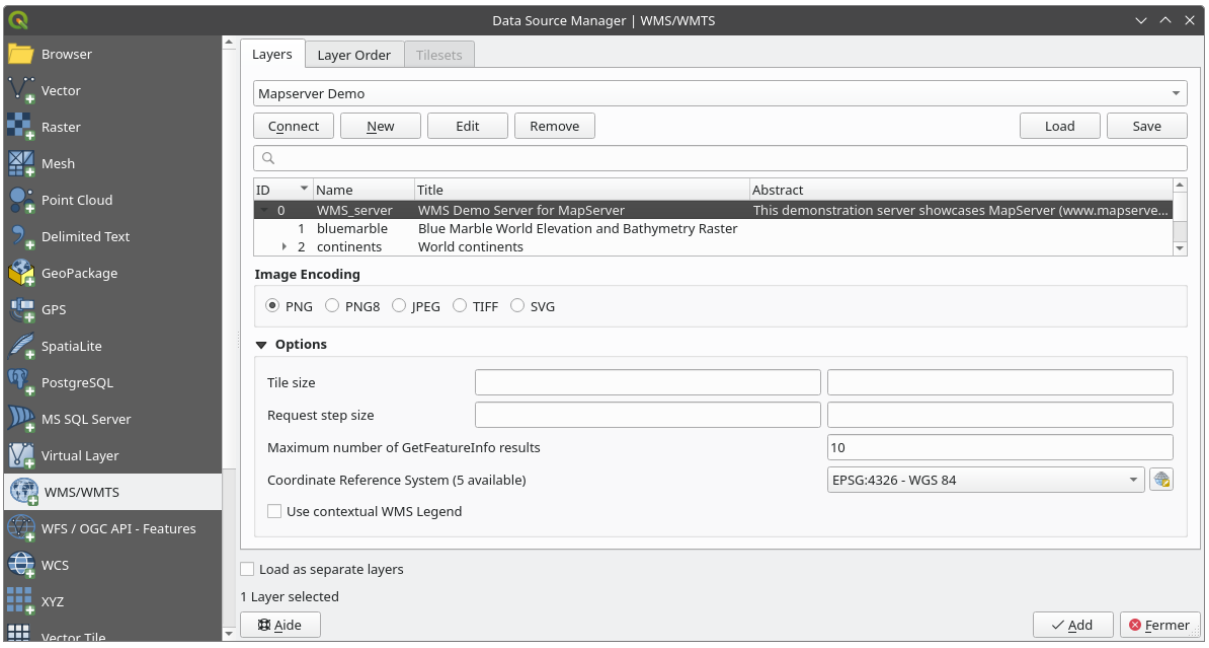


Figura19.3: Dialog for adding a WMS server, with filter on available layers

The upper part of the *Layers* tab of the dialog shows a tree structure that can include layer groups embedding layers with their associated image style(s) served by the server. Each item can be identified by:

- um *ID*
- um *Nome*
- um *Título*
- e um *Resumo*.

A lista pode ser filtrada usando a ferramenta de |pesquisa| no canto superior direito.

Codificação da Imagem

A seção *Codificação de Imagem* agora apresenta uma lista de formatos suportados pelo cliente e servidor. Escolha de acordo com as exigências de precisão da sua imagem.

> **Dica**
>
> **Codificação da Imagem**
>
> Tipicamente irá encontrar servidores WMS que oferecem a escolha da codificação de imagem JPEG ou PNG. O JPEG é um formato de perda de compressão, enquanto que o PNG reprodução fielmente os dados raster brutos.
>
> Use JPEG se espera dados WMS para fotografia na natureza e/ou não se importa da perda de alguma qualidade na imagem. Este típico trade-off reduz em 5 vezes o requerimento de transferência comparado ao PNG.
>
> Use o PNG se necessitar representações precisas dos dados originais, e não se importa dos requerimentos de transferência de dados.

Opções

The Options area of the dialog provides means to configure the WMS requests. You can define:

- *Tile size* if you want to set tile sizes (e.g., 256x256) to split up the WMS request into multiple requests.
- *Request step size*: if you want to reduce the effect of cut labels at tile borders, increasing the step size creates larger requests, fewer tiles and fewer borders. The default value is 2000.
- The *Maximum number of GetFeatureInfo results*: specifies the maximum number of results returned by a GetFeatureInfo request, for the layer(s) being loaded. Allows to override the *connection's default value* for specific layers.
- Each WMS layer can be presented in multiple CRSs, depending on the capability of the WMS server. If you select a WMS from the list, a field with the default projection provided by the web server appears. Press the Select CRS widget to replace the default projection of the WMS with another CRS supported by the WMS server.

  A dialog similar to the one shown in Fig.6.4 will appear. The main difference with the WMS version of the dialog is that only those CRSs supported by the WMS server will be shown.
- Finally you can activate *Use contextual WMS Legend* if the WMS Server supports this feature. Then only the relevant legend for your current map view extent will be shown and thus will not include legend items for items you can't see in the current map.

You can select several layers at once, but only one image style per layer. When several layers are selected, they will be combined at the WMS server and transmitted to QGIS in one go, as a single layer. The default name is a slash (/) separated list of their original title. You can however opt to *Load as separate layers*, in which case each layer is added to the map canvas using its default name from the queried service.

Ordem das Camadas

The *Layer Order* tab lists the selected layers available from the current connected WMS server.

WMS layers rendered by a server are overlaid in the order listed in the *Layers* tab, from top to bottom of the list. If you want to change the overlay order, you can use the *Up* and *Down* buttons of the *Layer Order* tab.

Transparência

The *Global transparency* setting from the *Layer Properties* is hard coded to be always on, where available.

19.1.5 Mosaicos

When using WMTS (Cached WMS) services you are able to browse through the *Tilesets* tab given by the server. Additional information like tile size, formats and supported CRS are listed in this table.

```
# example of WMTS service

https://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?service=WMTS&
↪request=GetCapabilities
```

Selecting a layer to load, it is also possible to apply an *Interpretation method*, converting it into a raster layer of single band float type, ready for styling using QGIS usual *raster renderers*.

In combination with this feature, you can use the tile scale slider by selecting *View* ► *Panels* (or *Settings* ► *Panels*), then choosing *Tile Scale Panel*. This gives you the available scales from the tile server with a nice slider docked in.

19.1.6 Usando a Ferramenta Identificar

Once you have added a WMS server, and if any layer from a WMS server is queryable, you can then use the Identify tool to select a pixel on the map canvas. A query is made to the WMS server for each selection made. The results of the query are returned in plain text. The formatting of this text is dependent on the particular WMS server used.

Seleção do formato

Se múltiplos formatos são suportados pelo servidor, uma caixa de combinação com os formatos suportados é automaticamente adicionado aos resultados da janela de identificação e o formato selecionado irá armazenar no projeto para a camada.

Suporte do formato GML

The Identify tool supports WMS server response (GetFeatureInfo) in GML format (it is called Feature in the QGIS GUI in this context). If "Feature" format is supported by the server and selected, results of the Identify tool are vector features, as from a regular vector layer. When a single feature is selected in the tree, it is highlighted in the map and it can be copied to the clipboard and pasted to another vector layer. See the example setup of the UMN Mapserver below to support GetFeatureInfo in GML format.

```
# in layer METADATA add which fields should be included and define geometry␣
↪(example):

"gml_include_items"   "all"
"ows_geometries"      "mygeom"
"ows_mygeom_type"     "polygon"

# Then there are two possibilities/formats available, see a) and b):

# a) basic (output is generated by Mapserver and does not contain XSD)
# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "application/vnd.ogc.gml,text/html"

# b) using OGR (output is generated by OGR, it is sent as multipart and contains␣
```

(continua na próxima página)

(continuação da página anterior)

```
↪XSD)
# in MAP define OUTPUTFORMAT (example):
OUTPUTFORMAT
    NAME "OGRGML"
    MIMETYPE "ogr/gml"
    DRIVER "OGR/GML"
    FORMATOPTION "FORM=multipart"
END

# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "OGRGML,text/html"
```

19.1.7 Viewing Properties

Once you have added a WMS server, you can view its properties by right-clicking on it in the legend and selecting *Properties*. The WMS/WMTS layer properties is much like the raster layer properties so you will find detailed description at *Caixa de Diálogo de Propriedades do Raster*. However, there are some differences, which will be explained below.

Information properties

Separador de Metadados

O separador *Metadados* exibem a riqueza de informação sobre o servidor WMS, geralmente recolhidos a partir da declaração de de Capacidades devolvidos a partir desse servidor. Muitas das definições podem ser removidas através da leitura dos padrões WMS (veja OPEN-GEOSPATIAL-CONSORTIUM *Referências Bibliográficas e Web*), mas aqui estão algumas definições úteis:

- **Propriedades do Servidor**
  - **Versão do WMS** — Versão do WMS suportada pelo servidor.
  - **Image Formats** — The list of MIME-types the server can respond with when drawing the map. QGIS supports whatever formats the underlying Qt libraries were built with, which is typically at least `image/png` and `image/jpeg`.
  - **Identity Formats** — The list of MIME-types the server can respond with when you use the Identify tool. Currently, QGIS supports the `text-plain` type.
- **Propriedades da Camada**
  - **Selecionado** — Querendo ou não esta camada selecionado quando o seu servidor foi adicionado a este projeto.
  - **Visível** — Se esta camada é ou não selecionada como visível na legenda (ainda não usada nesta versão do QGIS).
  - **Pode Identificar** — Camada que retorna ou não resultados através do uso da ferramenta Identificar.
  - **Pode ser transparente*** — Se esta camada pode ou não ser renderizada com transparência. Esta versão do QGIS utilizará sempre a transparência se esta for "Sim" e a codificação da imagem suportar a transparência.
  - **Can Zoom In** — Whether or not this layer can be zoomed in by the server. This version of QGIS assumes all WMS layers have this set to `Yes`. Deficient layers may be rendered strangely.
  - **Contagem em cascata** — Os servidores WMS pode funcionar como proxy para outros servidores WMS para obter dados raster para uma camada. Esta entrada mostra quantas vezes o pedido para esta camada é avançada por peer nos servidores WMS para o resultado.
  - **Fixed Width, Fixed Height** — Whether or not this layer has fixed source pixel dimensions. This version of QGIS assumes all WMS layers have this set to nothing. Deficient layers may be rendered strangely.

- **WGS 84 Bounding Box** — The bounding box of the layer, in WGS 84 coordinates. Some WMS servers do not set this correctly (e.g., UTM coordinates are used instead). If this is the case, then the initial view of this layer may be rendered with a very 'zoomed-out' appearance by QGIS. The WMS webmaster should be informed of this error, which they may know as the WMS XML elements `LatLonBoundingBox`, `EX_GeographicBoundingBox` or the CRS:84 `BoundingBox`.
- **SRC disponíveis** — As projeções desta camada podem ser renderizadas pelo servidor WMS. Os mesmos estão listados num formato nativo WMS.
- **Disponível em estilo** — Os estilos da imagem desta camada podem ser renderizadas pelo o servidor WMS.

Temporal properties

Raster *temporal properties* (namely *Dynamic Temporal Control*) can be set for WMS and WMTS layers. By default, when a time-dimension enabled WMS or WMTS layer is added to the project, it is indicated in the *Layers* panel with the Temporal Layer icon next to it. Its *Temporal* properties default to the *Automatic* temporal mode, meaning that the layer will follow the temporal controller's current time range by default.

You can then opt to show a specific static time value for the layer by unchecking *Dynamic Temporal Control* and picking an option under *Static WMS-T Temporal Range*:

- *Server default*
- *Predefined date* with a server exposing data for non-contiguous temporal ranges or *Predefined range* with a server exposing a range of available dates. A *Start date* and *End date* are necessary in the latter case. Their expected formatting can be deduced from the reference time option (see below). depending on whether the provider has data for contiguous period or not
- *Follow project's temporal range* as defined in the project's properties dialog

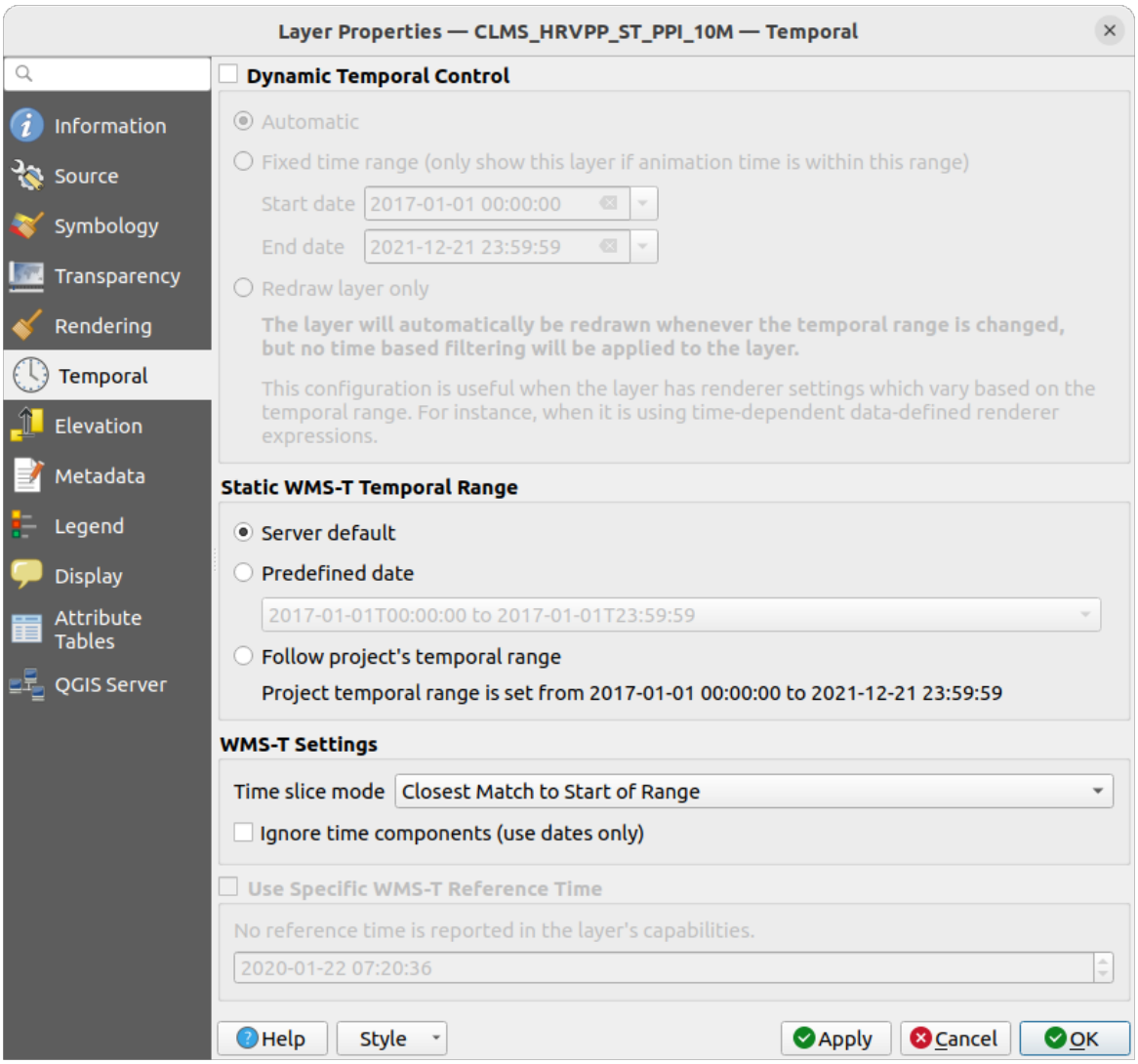


Figura19.4: Temporal properties of a WMTS layer

Whatever temporal data control is in use, there are some *WMS-T Settings* to help display the correct temporal data:

- *Time slice mode* which can be:
  - *Use whole temporal range*
  - *Match to start of range*
  - *Match to end of range*
  - *Closest match to start of range*
  - *Closest match to end of range*
- *Ignore time components (use dates only)*: If checked, the time component of temporal queries will be discarded and only the date component will be used in server requests.

You can also *Use Specific WMS-T Reference Time* picked from times reported in the layer's capabilities. Convenient for servers which expose a non-contiguous set of date time instances (instead of a range of dates).

QGIS Server properties

In addition to the raster layer *properties*, publishing a WMS/WMTS layer with QGIS Server will display following options:

- *WMS Print layer*: Allows to set alternative WMS layer that should be used for printing (in the GetProjectSettings reply). Convenient for WMTS layers that are generally not suitable for printing.
- ☑ *Publish WMS/WMTS data source uri*: Allows a web client to fetch the WMS/WMTS data directly
- ☑ *Advertise as background layer*

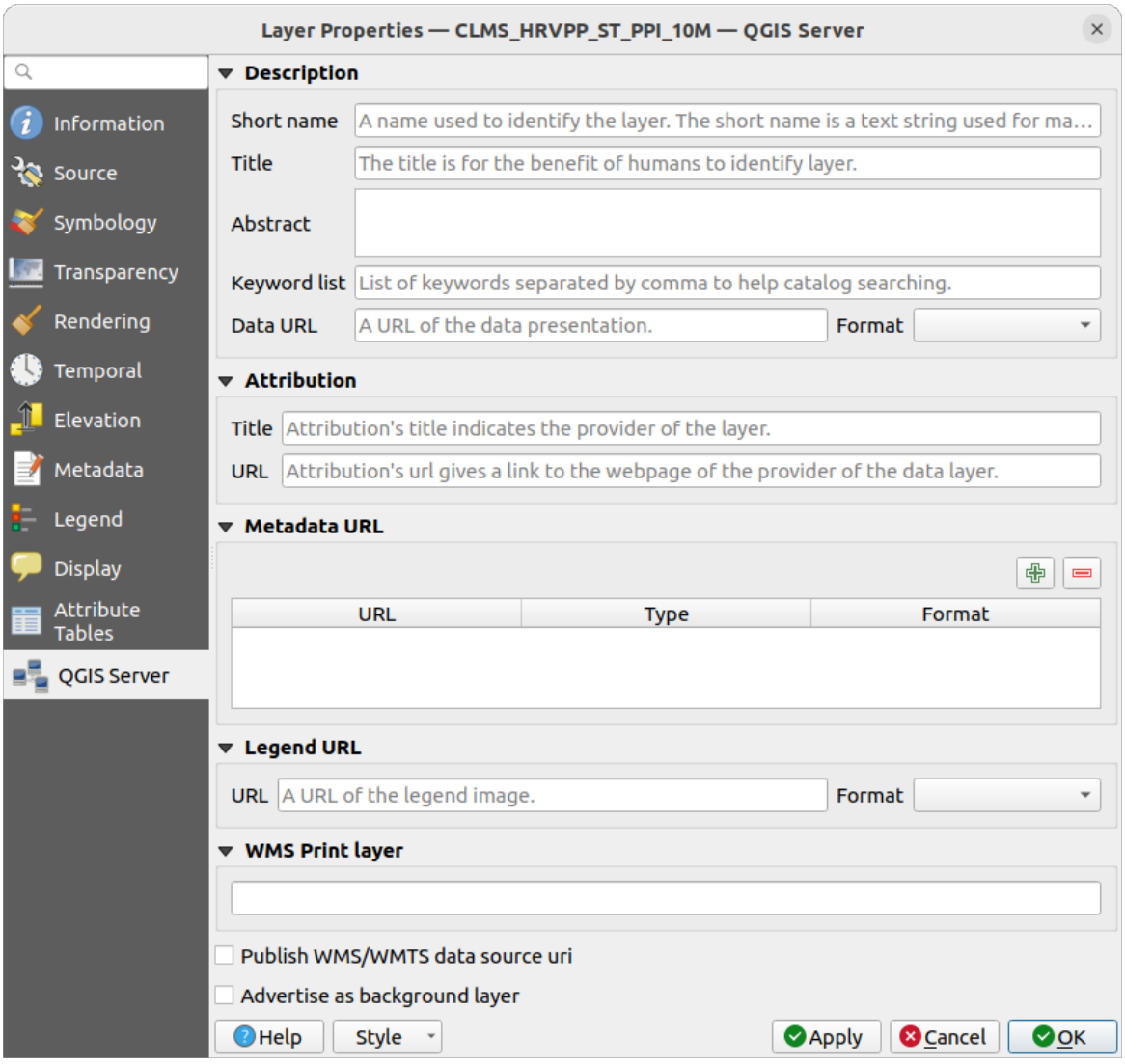


Figura19.5: QGIS Server properties of a WMS/WMTS layer

19.1.8 Mostrar gráfico de legendas WMS no índice e layout

The QGIS WMS data provider is able to display a legend graphic in the table of contents' layer list and in the print layout. The WMS legend will be shown only if the WMS server has GetLegendGraphic capability and the layer has getCapability url specified, so you additionally have to select a styling for the layer.

Se uma legendGraphic está disponível, será mostrada abaixo da camada. É pequena, e você tem que clicar sobre ela para abrir na dimensão real (devido à limitação arquitetônica QgsLegendInterface). Clicando sobre a legenda da camada abrirá um quadro com a legenda em resolução máxima.

In the print layout, the legend will be integrated at it's original (downloaded) dimension. Resolution of the legend graphic can be set in the item properties under *Legend* ► *WMS LegendGraphic* to match your printing requirements.

A legenda irá exibir informações contextuais com base em sua escala atual. A legenda WMS será exibida somente se o servidor WMS tiver capacidade GetLegendGraphic e a camada getCapability url for especificada, então você tem que selecionar um estilo.

19.2 WCS Cliente

O Serviço Web de Cobertura (WCS) fornecem acesso aos dados matriciais de forma a serem úteis à renderização cliente, como entrada a modelos científicos, e para outros clientes. O WCS pode ser comprado ao WFS e ao WMS. Como as instâncias WMS e WFS, o WCS permite aos clientes escolher porções de informação salvadas de servidores baseados condicionantes espaciais e outros critérios de consulta.

QGIS has a native WCS provider and supports both version 1.0 and 1.1 (which are significantly different), but currently it prefers 1.0, because 1.1 has many issues (i.e., each server implements it in a different way with various particularities).

The native WCS provider handles all network requests and uses all standard QGIS network settings (especially proxy). It is also possible to select cache mode ('always cache', 'prefer cache', 'prefer network', 'always network'), and the provider also supports selection of time position, if temporal domain is offered by the server.

Loading a WCS Layer

To be able to load a WCS Layer, first create a connection to the WCS server:

1. Open the *Data Source Manager* dialog by pressing the Open Data Source Manager button
2. Enable the *WCS* tab
3. Click on *New…* to open the *Create a New WCS Connection* dialog

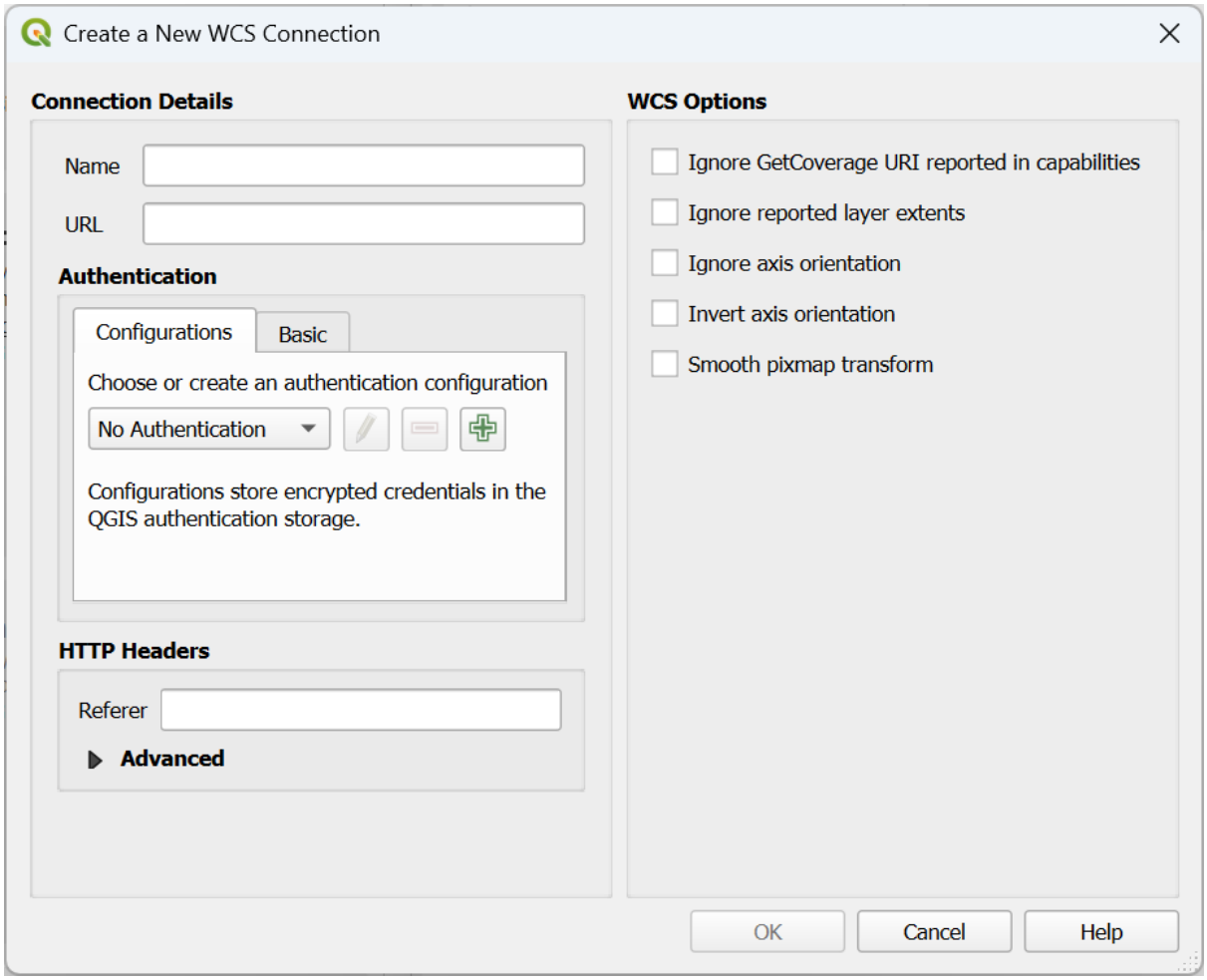


Figura19.6: Creating a connection to a WCS server

- *Name*: A name for the connection. This name will be used in the Server Connections drop-down box so that you can distinguish it from other WCS servers.
- *URL*: URL of the server providing the data. This must be a resolvable host name – the same format as you would use to open a telnet connection or ping a host, i.e. the base URL only. For example, you shouldn't have fragments such as `request=GetCapabilities` or `version=1.0.0` in your URL.
- *Authentication* (optional): using a *stored configuration* or a basic authentication with *Username* and *Password*.

> **Aviso**
>
> Entering **username** and **password** in the *Authentication* tab will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you shared the project file with someone. Therefore, it's advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab). See *Sistema de Autenticação* for more details.

- HTTP *Referência*
- ☐ *Ignore GetCoverage URI reported in capabilities*: if checked, use given URI from the *URL* field above.
- ☐ *Ignore reported layer extents*: because the extent reported by raster layers may be smaller than the actual area which can be rendered (notably for WCS servers with symbology which takes more space than the data extent), check this option to avoid cropping raster layers to their reported extents, resulting in truncated symbols on the borders of these layers.

- ☐ *Ignore axis orientation*
- ☐ *Inverter orientação do eixo*
- ☐ *Suavizar transformação do mapa de pixels*

4. Pressione *OK* para criar a conexão.

Note that any proxy settings you may have set in your preferences are also recognized. Also note that it is possible to *Load* the connection parameters from a `.XML` file or *Save* them to a `.XML` file.

Now we are ready to load WCS layers from the above connection.

19.3 WFS e WFS-T Cliente

In QGIS, a WFS layer behaves pretty much like any other vector layer. You can identify and select features, and view the attribute table. QGIS supports WFS 1.0.0, 1.1.0, 2.0 and OGC API - Features (OAPIF), including editing (through WFS-T). QGIS also supports background download and progressive rendering, on-disk caching of downloaded features and version autodetection.

Layers of servers implementing OGC API - Features - Part 4: Create, Replace, Update and Delete can be turned into editing mode to allow creating, modifying and deleting features. Note that each created/modified/deleted feature requires a dedicated network request, so performance might suffer in case of simultaneous modification of hundreds or more features at a time.

In general, adding a WFS layer is very similar to the procedure used with WMS. There are no default servers defined, so you have to add your own. You can find WFS servers by using the *MetaSearch plugin* or your favourite web search engine. There are a number of lists with public URLs, some of them maintained and some not.

Carregando uma camada WFS

As an example, we use the Gateway Geomatics WFS server and display a layer.

```
https://demo.gatewaygeomatics.com/cgi-bin/wfs_gateway?REQUEST=GetCapabilities&
↪VERSION=1.0.0&SERVICE=WFS
```

Para poder carregar uma camada WFS, primeiro crie uma conexão com o servidor WFS:

1. Open the *Data Source Manager* dialog by pressing the Open Data Source Manager button
2. Enable the *WFS / OGC API - Features* tab
3. Clique em *Novo...* para abrir o diálogo *Criar uma nova conexão WFS*
4. In the *Connection Details* group, enter `Gateway Geomatics` as name
5. Insira a URL (veja acima)
6. Optionally, enter the *Authentication* settings

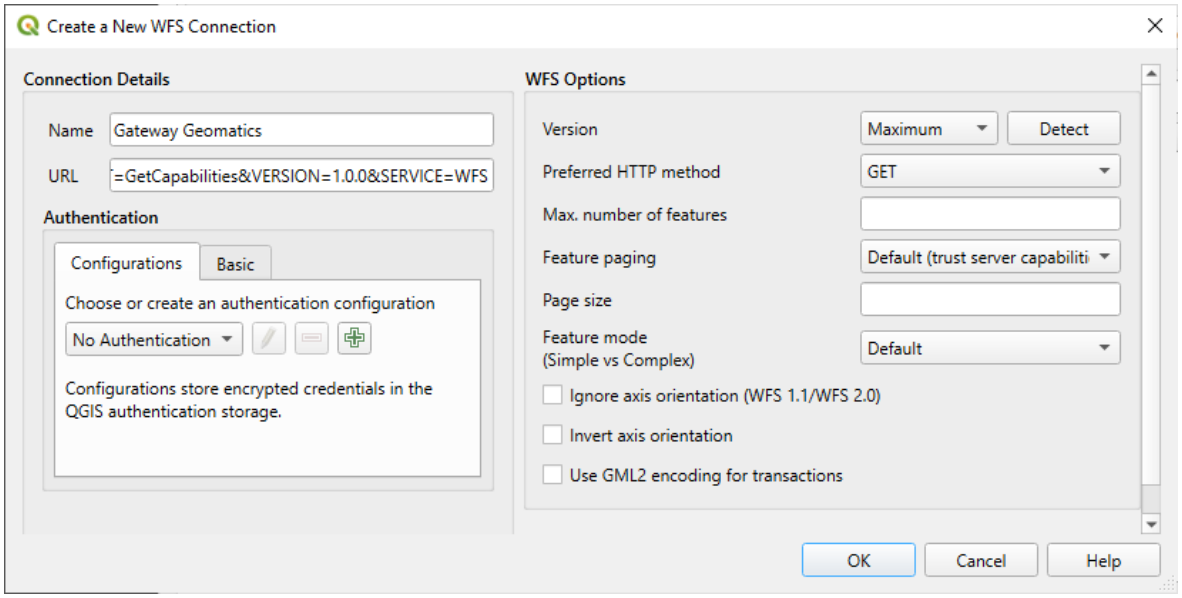


Figura19.7: Criando uma conexão com um servidor WFS

Nota

In case of an OGC API - Features (OAPIF), the URL to provide should be the landing page, i.e., the main page from which it is possible to navigate to all the available service endpoints.

7. In the *WFS Options* group, you can:
   - Indique a versão WFS do servidor. Se desconhecido, pressione o botão *Detectar* para recuperá-la automaticamente.
   - Select the *Preferred HTTP method* to use for requests. The default is *GET*, but you can also select *POST*.
   - Define the *Maximum number of features* retrieved in a single GetFetFeature request. If empty, no limit is set.
   - *Feature mode (simple vs complex)*: helps determine how to handle servers serving data with simple or complex GML schemas. Available options are:
     - *Default*: if the server supports editing capabilities, this is the same as using *Simple Features*. If not, it fallbacks to using *Complex Features* mode.
     - *Simple Features*: returns data as simple features only
     - *Complex Features*: uses the *OGR GMLAS <https://gdal.org/en/stable/drivers/vector/gmlas.html>*, potentially returning complex features, depending on the data schema, where nested data structures are exposed by QGIS as JSON.
   - Depending on the WFS version, indicate whether to:
     - Enable *Feature paging* and specify the maximum number of features to retrieve with *Page size*. If no limit is defined, then the server default is applied.
     - Forçar a *Ignorar orientação dos eixos (WFS 1.1/WFS 2.0)*
     - *Inverter orientação do eixo*
     - *Use GML2 encoding for transactions.*

> **Aviso**
>
> Entering **username** and **password** in the *Authentication* tab will keep unprotected credentials in the connection configuration. Those **credentials will be visible** if, for instance, you shared the project file with someone. Therefore, it's advisable to save your credentials in an *Authentication configuration* instead (*Configurations* tab). See *Sistema de Autenticação* for more details.

8. Pressione *OK* para criar a conexão.

Note that any proxy settings you may have set in your preferences are also recognized. Also note that it is possible to *Load* the connection parameters from a `.XML` file or *Save* them to a `.XML` file.

Agora estamos prontos para carregar as camadas WFS a partir da conexão acima.

1. Choose 'Gateway Geomatics' from the *Server Connections* drop-down list.
2. Clique :guilabel:`Conectar'.
3. Selecione a camada *Parques* na lista
4. Você também pode escolher se deseja:
   - *Only request features overlapping the view extent*
   - *Change…* the layer's CRS to any other supported by the service
   - or build a query to specify particular features to retrieve from the service: double-click on the row of the layer to open the *SQL Query Composer* dialog. That dialog provides widgets to write an advanced SQL query relying on available tables and columns of the service, with sorting and filtering and a bunch of SQL functions, spatial predicates and operators.

     The query you build will appear after validation in the *SQL* column within the *WFS / OGC API - Features* table, and the filtered layer will display the icon next to it in the *Layers* panel. It is thus possibe to adjust the query at any moment.
5. Click *Add* to add the layer to the map, using its default name from the queried service.

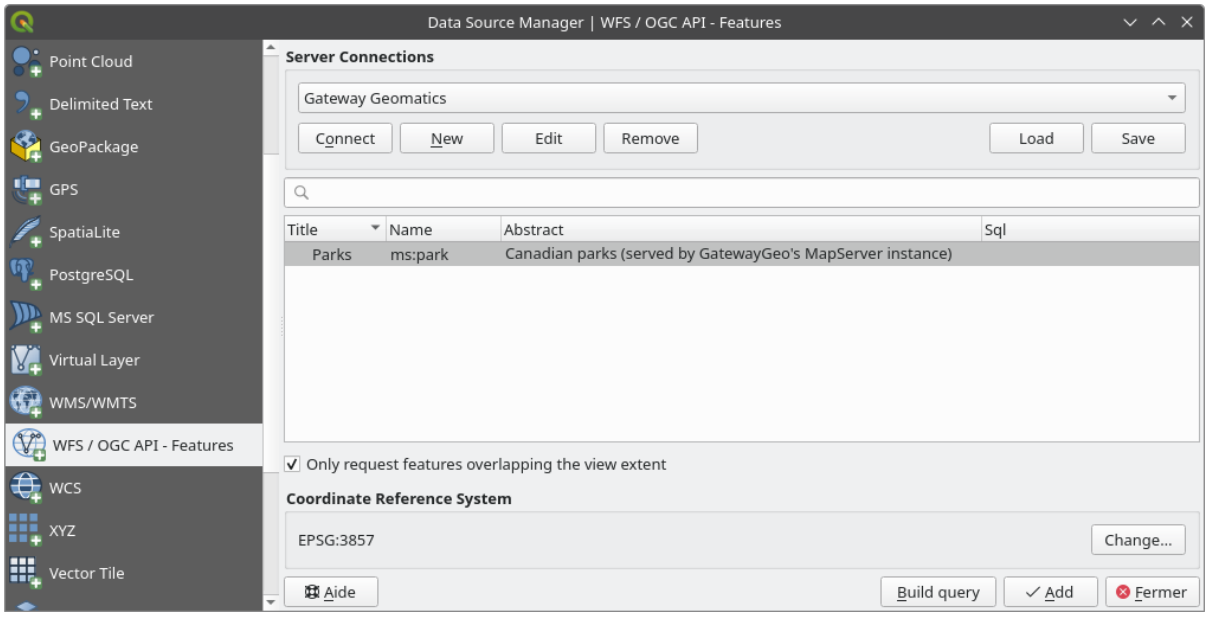


Figura19.8: Adicionando uma camada WFS

O progresso do download é visualizado na parte inferior esquerda da janela principal do QGIS. Uma vez carregada a camada, você pode identificar e selecionar algumas feições e visualizar a tabela de atributos.

19.4 SensorThings

QGIS supports connections to OGC SensorThings API, a standard providing an open and unified framework to interconnect IoT sensing devices, data, and applications over the Web. It is an open standard addressing the syntactic and semantic interoperability of the Internet of Things. It is based on the Observations and Measurements data model, a standardized model for observations, and for features involved in sampling when making observations.

19.4.1 Setting connection

To add SensorThings data to QGIS use the *SensorThings* tab in the *Data Source Manager* dialog.

To establish a new connection, press *New* (or *New SensorThings Connection* from the Browser panel) and provide *Name* and *URL*. Advanced options, such as *authentication* and a *Referer*, can also be configured.

Press *OK* to establish the connection. Then you will be able to:

- *Edit* the SensorThings connection settings
- *Remove* the SensorThings connection

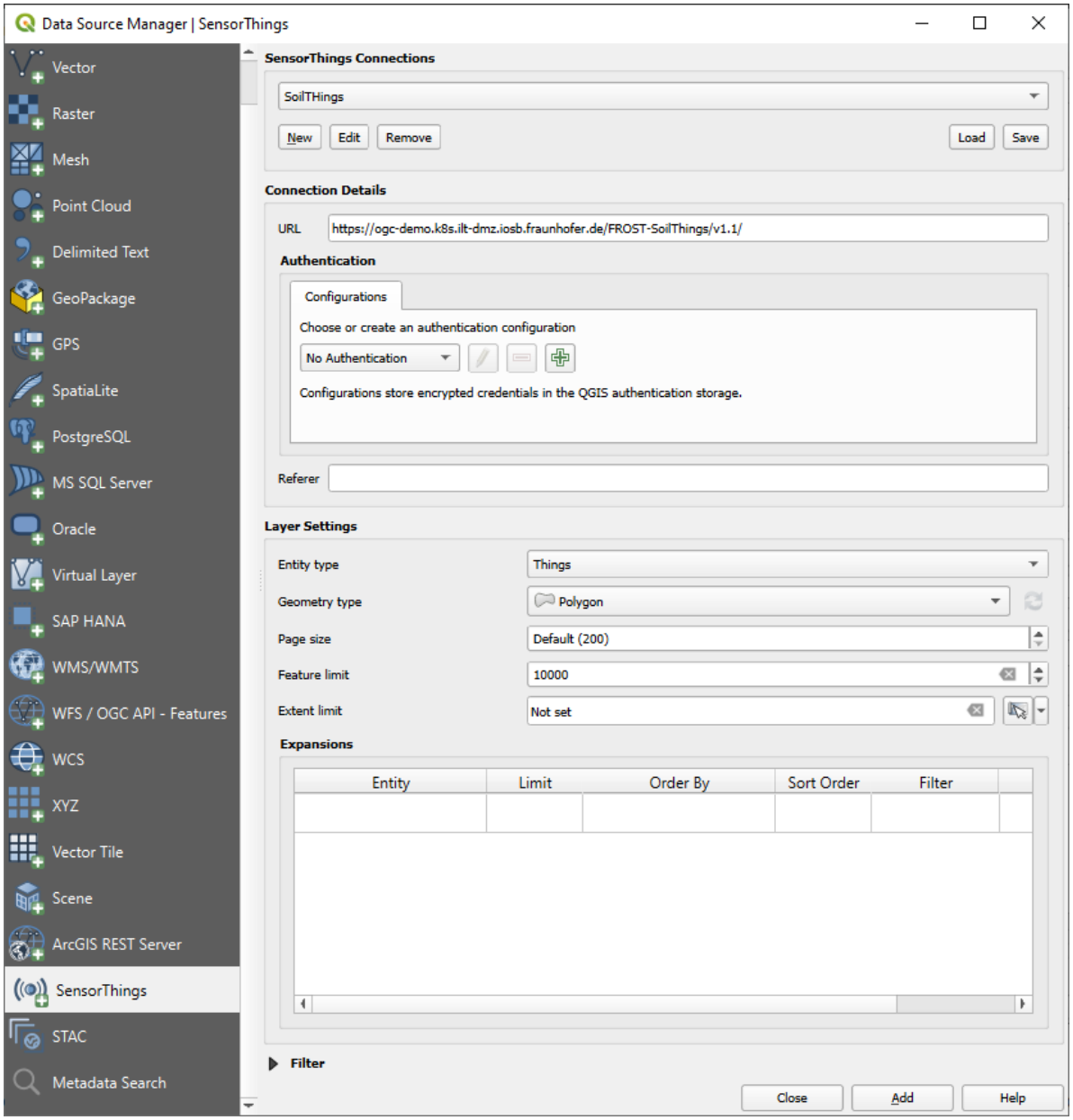


Figura19.9: SensorThings Connection dialog

Configurations can be saved to an `.XML` file (*Save*) through the *SensorThings* entry in *Data Source Manager* dialog or its contextual menu in the *Browser* panel (*Save Connections*). Likewise, configurations can be added from a file (*Load*).

19.4.2 Loading SensorThings data

Relations between layers (so-called entities) stored in a SensorThings dataset are expressed in the diagram below.

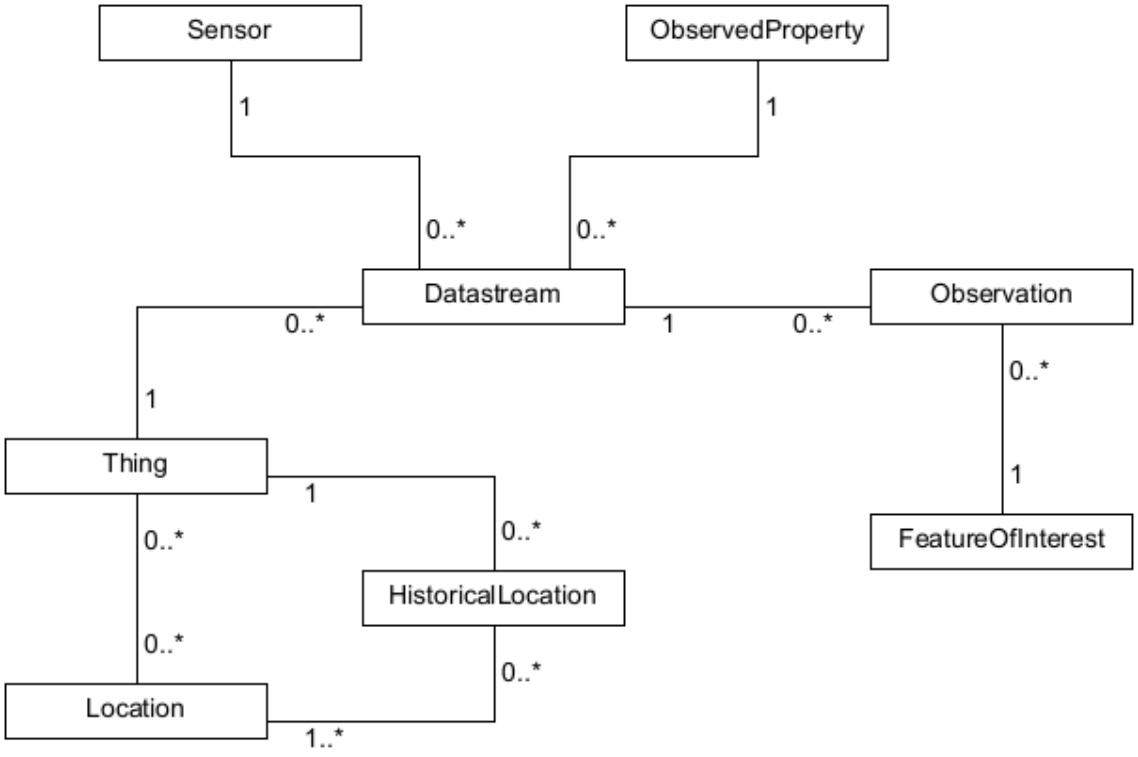


Figura19.10: Data model Observations and Measurements (Source: Sensor Web Tutorial by SIST network)

Any type of entity can be loaded in QGIS, but not all are spatial data. To load an entity, there are *Layer Settings* that can be configured:

- *Entity Type*: the entity to load from the data model as layer in QGIS
- *Geometry Type*: the geometry type of the selected entity to load. Press Check available types to limit the list to the actually supported geometry types.
- *Page Size*
- *Feature Limit* sets a maximum number of features to request from the service
- *Extent Limit* sets a maximum extent limit for the layer, so that only features within the extent are requested
- *Expansions*: The data model of SensorThings provides a mechanism of expansion of the results to related entities, similar to how tables are joined together in a relational database. Using this approach, you can expand the selected layer to include data from other items. This will flatten the relationship, creating as many parent features as children, and additional properties are added as columns in the attribute table.

Use *Filter* to build a query to filter the data, using SensorThings filter syntax.

Nota

The above settings and filtering options are also available for update in the layer properties dialog, *Source* tab, once loaded in QGIS.

Press *Add* to load the selected entity type as layer in QGIS.

19.4.3 Working with a vector layer from SensorThings

A SensorThings layer is loaded in QGIS as a vector layer. As such, it displays the same tabs in the *layer properties* and allows same feature interactions using the selection or identify tools. There are however some specificities you should consider while working with SensorThings data.

Because of the data model, the result property of a SensorThings Observation is a string field. In case you want to use its numerical representation in for example a graduated style, use an expression to convert the value to real and try() in case this fails (e.g., `try( to_real("Observation_result"), Null)`).

In case you want to create a chart of the observations at one or more locations, you can install the QGIS plugin *Data Plotly*.

1. Now select the observations at a point location in the map view.
2. Open the plotly panel and activate the *Use only selected features* checkbox.
3. Select on the x-column a date-time property and on the y-column the *Observation_result*. This will plot the observations at that location over time.
4. Verify to filter by a single Observed Property.
5. Notice that the chart changes as soon as you select other locations on the map.

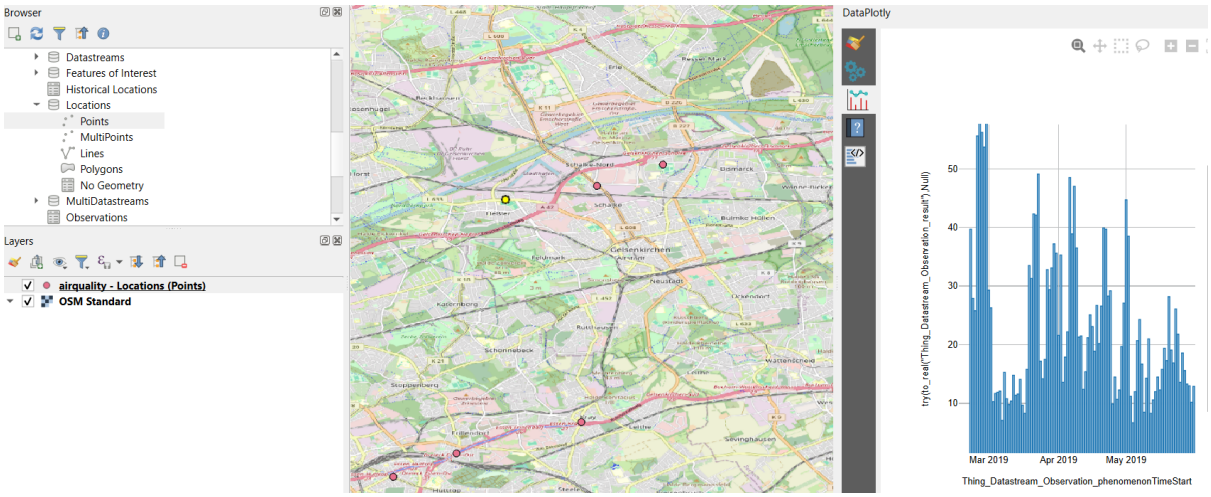


Figura19.11: Use Data plotly to plot the air quality observations at a location

19.5 STAC (SpatioTemporal Asset Catalogs)

STAC (SpatioTemporal Asset Catalog) is a specification for describing geospatial data in a consistent and accessible way. It defines a standard structure for organizing and indexing spatial-temporal assets such as satellite imagery, drone photos, or sensor data so that you can search, preview, and use these datasets across different platforms and tools.

STAC enables QGIS users to connect to and browse geospatial datasets, either from static catalogs or dynamic APIs, and download or stream assets such as imagery or other spatial data.

More information at https://stacspec.org/en.

Key components of STAC in QGIS include:

- **STAC Items**: The basic unit of a STAC catalog. Each item represents one geospatial asset or a group of related assets (e.g. image bands) at a specific time and location. It includes metadata such as geometry, datetime, links to assets (e.g., TIFFs, JSON, COG), and properties.
- **STAC Collections**: A grouping of items that share common characteristics and metadata (e.g. satellite mission). Collections may define spatial and temporal extents, licensing, and keywords.
- **STAC Catalogs**: A hierarchical container that organizes items and/or collections. Catalogs allow navigation of STAC datasets but do not necessarily include search capabilities.
- **STAC API**: An implementation of the STAC specification that allows querying and filtering of STAC items using spatial and temporal based filters, as well as searching within specific collections. STAC APIs follow the OGC API - Features pattern and support dynamic access to datasets.

There is an important distinction between static STAC catalogs and STAC API endpoints:

- **Static STAC Catalogs**: are collections of JSON files without search capabilities. They can be browsed via the Browser panel.
- **STAC API**: provide search capabilities and can be accessed in QGIS through both the Browser panel and the Data Source Manager.

19.5.1 Setting connection

STAC connections can be added in QGIS using either the *Browser panel* or the *Data Source Manager*:

- **Browser Panel:** In the *Browser panel*, right-click on the *STAC* entry and select *New STAC Connection…*. In the dialog that appears, enter a *Name* for the connection, the *URL* of the STAC catalog and optionally fill in *Authentication* credentials and a *Referer*. Then click *OK*.

  Use this method for browsing static STAC catalogs that do not support search or filtering.

- **Data Source Manager:** For STAC APIs you can use *Data Source Manager* dialog.

  Open the *Data Source Manager*, choose the *STAC* tab and click the *New* button. Fill in the *Name* and *URL* fields, and (optional) the *Authentication* credentials and a *Referer*. Press *OK* and then *Connect* to establish the connection, after that you will be able to:

  - *Edit* the STAC connection settings
  - *Remove* the STAC connection

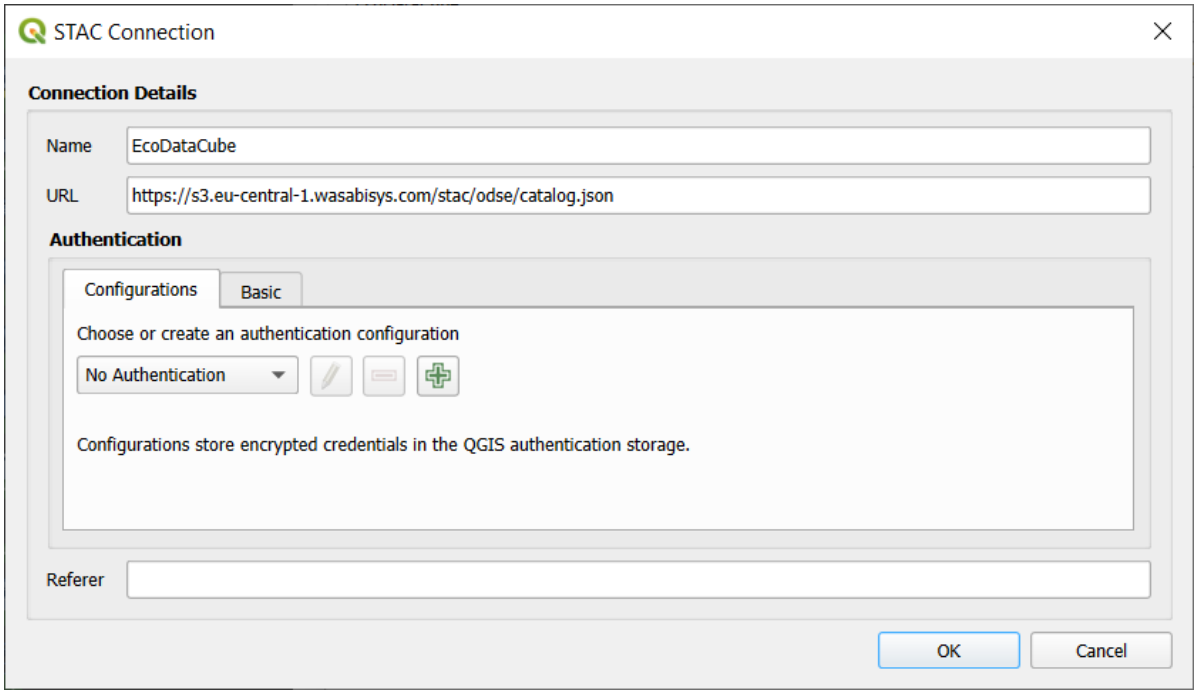


Figura19.12: Creating a connection to a STAC server

19.5.2 Browsing STAC Catalogs

Static STAC catalogs are displayed as hierarchical structures within the *Browser panel*. Once connected, the STAC catalog appears under *STAC* in the *Browser* panel. You can expand the catalog node to see its *Collections*. Expanding a collection reveals the individual *Items* it contains.

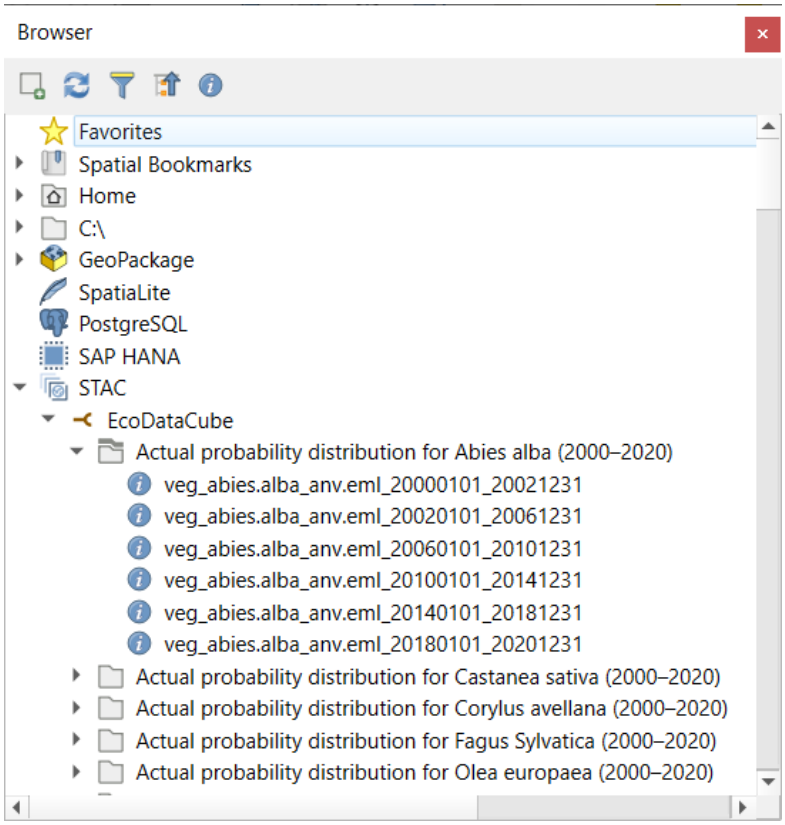


Figura19.13: STAC connection expanded in the Browser, showing Collections and Items

Right-click any STAC *Item* and choose *Details…* to view its metadata. The details panel shows the item's JSON content and a map of its coverage. If an item's asset is a cloud-optimized format (e.g. a `.COG`), you can add it directly to the map canvas. Otherwise, *Download Assets…* to save it locally before use.

19.5.3 Filtering and Searching STAC Items

For STAC API endpoints, QGIS supports spatial and temporal filtering, as well as limiting searches to specific collections.

To search for items:

1. Open *Data Source Manager* and go to the *STAC*
2. Select the STAC connection you created earlier and click *Filters…*

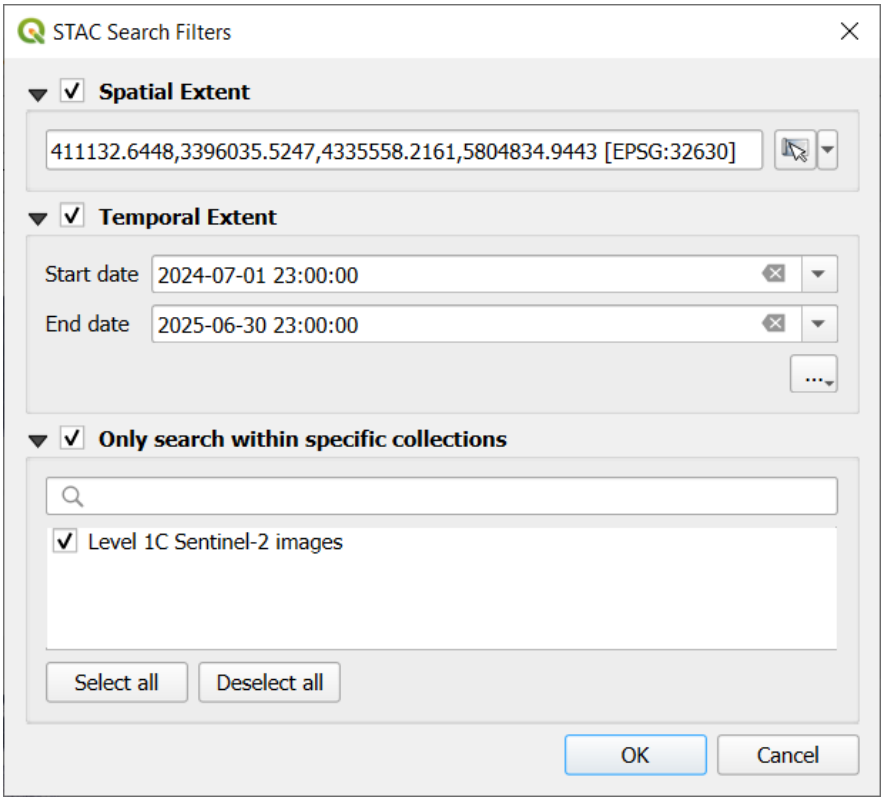


Figura19.14: Define spatial and temporal filters for a STAC catalog search

3. In the filter dialog, you can:
   - Enable *Spatial Extent* to restrict results to a bounding box or drawn area.
   - Enable *Temporal Extent* and define a date or range.
   - Limit to specific collections using *Only search within specific collections*
4. After applying filters, results will be listed in the main *Data Source Manager* panel.
5. Optionally check *Show Footprints* to display the footprints of the results on the map.

Right-click a result item to access actions:

- *Zoom to Item*
- *Pan to Item*
- *Download Assets*
- *Details…*

If the item's asset requires download, use the *Download Assets* option.

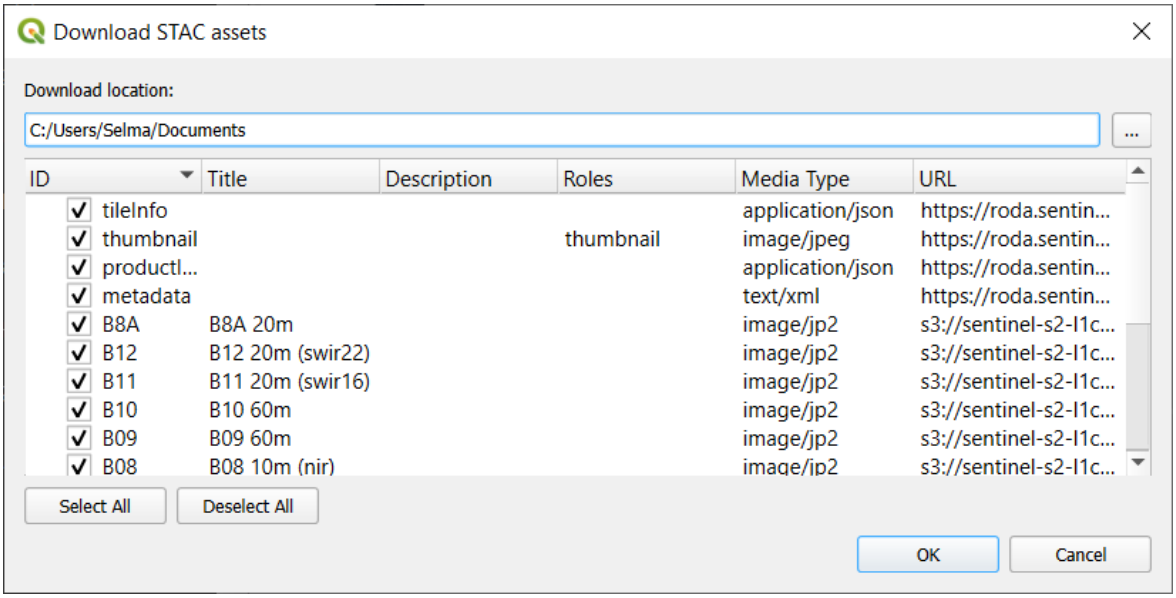


Figura19.15: Download STAC item assets

Downloaded assets include the main dataset and any auxiliary files such as thumbnails or style files. After download, use standard QGIS tools (e.g., *Add Raster Layer…*) to load and display the data.

19.6 ArcGIS REST Server

An ArcGIS REST Server can host many different types of web services (feature service, map service, image service, …). Read *Using ArcGIS REST Servers* for instructions to connect to an ArcGIS REST server.

When loading layers form an ArcGIS Feature Service, symbols are automatically converted to QGIS-native symbology, allowing QGIS to visually reproduce the same appearance as ArcGIS web maps.

Layers from an ArcGIS Feature Service can be editable in QGIS if:

1. The ArcGIS service has editing enabled
2. Your credentials grant write access
3. And the remote layer supports editing operations (Create, Update, or Delete capabilities). If these options are disabled on the server, the layer will be read-only in QGIS.

Editing these layers follow the *same procedure* as for other vector layers. To update your map with the latest changes, use *View* ► *Refresh* or press `F5`. If multiple users edit the same layer concurrently, it is recommended to refresh your data before performing edits to avoid conflicts.

CAPÍTULO 20

Trabalhando com dados GPS

20.1 Introducing GNSS/GPS Data

20.1.1 O que é GPS?

GPS, the **Global Positioning System**, is a satellite-based system that allows anyone with a GPS receiver to find their exact position anywhere in the world. GPS is used as an aid in navigation, for example in airplanes, in boats and by hikers. The GPS receiver uses the signals from the satellites to calculate its latitude, longitude and (sometimes) elevation. Most receivers also have the capability to store:

- locations (known as **waypoints**)
- sequences of locations that make up a planned **route**
- and a **track** log of the receiver's movement over time.

Waypoints, routes and tracks are the three basic feature types in GPS data. QGIS displays waypoints in point layers, while routes and tracks are displayed in linestring layers.

> **Nota**
>
> O QGIS também suporta receptores GNSS. Mas continuamos usando o termo GPS nesta documentação.

20.1.2 Transferring or loading GPS data

Loading a GPX file

Existem dezenas de formatos de arquivo diferentes para armazenar dados GPS. O formato que o QGIS usa é chamado GPX (formato GPS eXchange), que é um formato padrão de intercâmbio que pode conter qualquer número de pontos de passagem, rotas e trilhas no mesmo arquivo.

To load a `GPX` file:

1. Open the *GPS* tab in the *Data Source Manager* dialog, i.e.:
   - Click the Open Data Source Manager button on the toolbar (or press `Ctrl+L`) and enable the target tab
   - Or select *Layer* ► *Add Layer* ► *Add GPX Layer…*
2. Use the … Browse button next to the *GPX dataset* option to select the GPX file
3. Use the check boxes to select the *Feature types* you want to load from the file. Each feature type (*Waypoints*, *Tracks* or *Routes*) will be loaded in a separate layer.

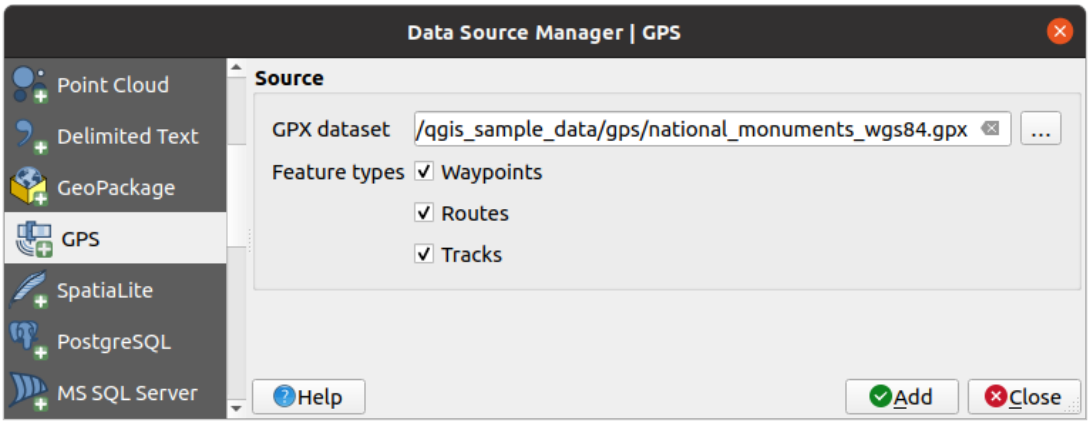


Figura20.1: Carregando a janela de dados do GPS

Loading to or from a device

There are lots of different types of GPS devices and formats. Since QGIS uses GPX files, you need a way to convert other GPS file formats to GPX. QGIS can do that using the free program GPSBabel. GPSBabel can help you convert waypoints, tracks, and routes between popular GPS receivers such as Garmin or Magellan and mapping programs like Google Earth or Basecamp. Literally hundreds of GPS receivers and programs are supported. It can also transfer GPS data between your computer and a GPS device.

Under *Settings* ► *Options* ► *GPS* ► *GPSBabel*, QGIS allows you to define your own device type and set parameters of conversion that could later be used by the *Processing GPS algorithms*.

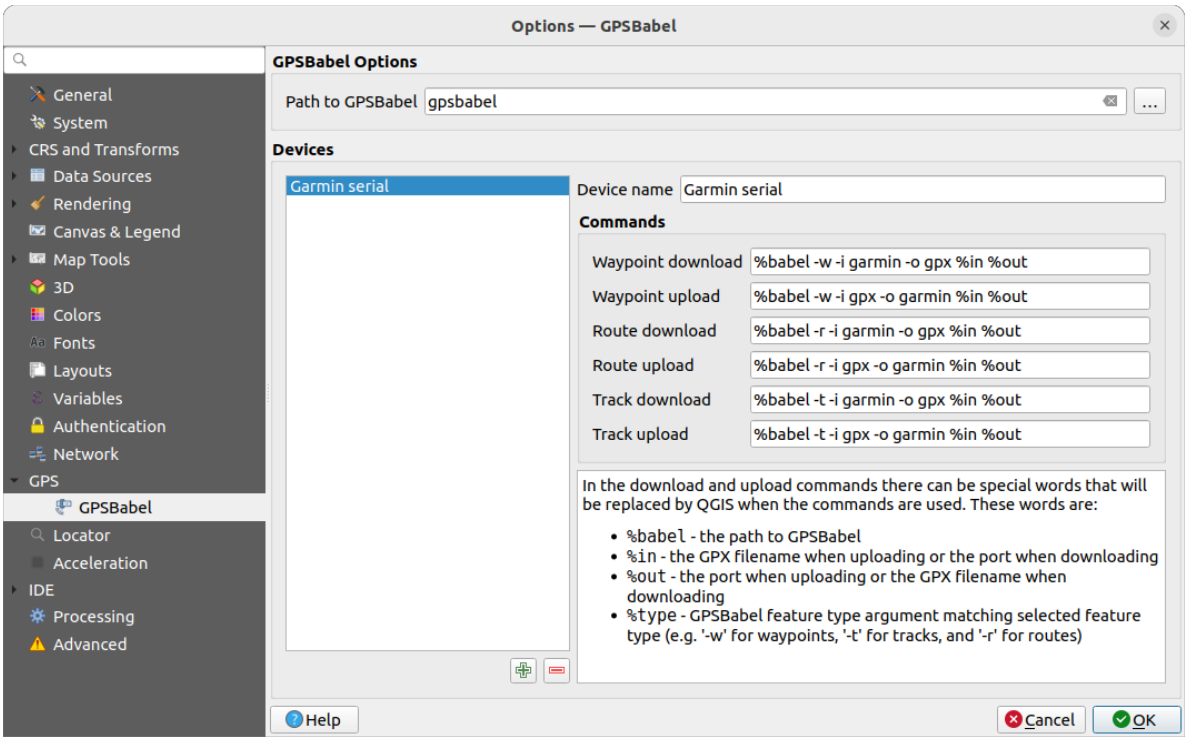


Figura20.2: Configurações do GPS Babel

1. Primeiro você tem que definir os binários *Path to GPSBabel*.
2. Então você pode querer adicionar seu dispositivo. Você pode atualizar a lista de dispositivos usando o botão Adicionar novo dispositivo ou Remover dispositivo.
3. Para cada aparelho:
   - Você fornece um *Nome do dispositivo*
   - Você configura diferentes *Comandos* que o QGIS usará ao interagir com ele, como:
     - *Baixar Waypoint* do dispositivo.
     - *Upload de waypoint* para o dispositivo
     - *Download de rota* do dispositivo
     - *Encaminhar upload* para o dispositivo
     - *Faixa de download* do dispositivo
     - *Track upload* para o dispositivo

     Embora os comandos sejam geralmente comandos GPSBabel, você também pode usar qualquer outro programa de linha de comando que possa criar um arquivo GPX.O QGIS substituirá as palavras-chave``%type``, `%in`, e `%out` quando rodar o comando.

     As an example, if you create a device type with the download command `gpsbabel %type -i garmin -o gpx %in %out` and then use it to download waypoints from port `/dev/ttyS0` to the file `output.gpx`, QGIS will replace the keywords and run the command `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx`.

     Leia o manual do GPSBabel para as opções de linha de comando que podem ser específicas para o seu caso de uso.

Uma vez que você criou um novo tipo de dispositivo, ele aparecerá nas listas de dispositivos para os algoritmos de download e upload do GPS.

> **Nota**
>
> GPS units allow you to store data in different coordinate systems. When downloading a GPX file (from your GPS unit or a web site) and then loading it in QGIS, be sure that the data stored in the GPX file uses WGS 84 (latitude/longitude). QGIS expects this, and it is the official GPX specification. See GPX 1.1 Schema Documentation.

20.2 Rastreio GPS em tempo real

QGIS can help you do field mapping with a GPS receiver. Such a live tracking operation is done using the *GPS Toolbar*. Some *device configuration* may be needed before connecting QGIS and the GPS receiver.

20.2.1 GPS toolbar

The *GPS Toolbar* provides the main tools to control a live tracking session. It can be activated from *View* ► *Toolbars* ► *GPS*. It follows the state of the project, GPS, and current GPS track, and enables actions only when they make sense. Once a device is detected by QGIS, you will be able to interact with it:

- Connect to GPS
- Disconnect from GPS
- Recenter map on GPS location: the map will immediately recenter on the current GPS position regardless of the *automatic recentering parameter*.
- Set destination layer for GPS digitized features: By default, QGIS will *Follow active layer* for feature digitizing, meaning that GPS digitizing tools will adapt to the layer selected in *Layers* panel and created features will be stored in it accordingly. While this may be useful in some contexts, it also implies careful interaction with other layers to avoid storing inadvertently features in undesired layers. This option allows you to explicitly indicate a layer for the data storage during the live tracking session, and you could switch depending on your needs. Exposed tools will adapt as well to the selected layer type.

  The GPS destination layer is made automatically editable when a feature is being created, and the user is notified accordingly.
- actions for feature creation:
  - Add vertex to GPS track using GPS current location
  - Create a new point feature at current GPS location
  - Create a new line feature using the current GPS track
  - Create a new polygon feature using the current GPS track
  - Reset track
- Show GPS information: opens the *GPS Information* panel
- a display box for quick access to some GPS information components you may need:
  - Show location
  - Show altitude (Geoid)
  - Show altitude (WGS 84 ellipsoid)

  - Show ground speed
  - Show bearing
  - Show total track length
  - Show distance from start of track

- The Settings button opens a drop-down menu with common settings which are expected to be modified mid-session:
  - *Show location marker*
  - *Show bearing line*
  - *Rotate map to match GPS direction*
  - Options to control map recentering:
    - *Always recenter map* when the GPS position is offset from the map canvas center by a certain distance (as a ratio of the map canvas extent)
    - *Recenter map when leaving extent*
    - *Never recenter*
  - *Automatically add track vertices* whenever new locations are received from the GPS device.
  - *Automatically save added features*: features created from GPS locations are immediately committed to their target layers (skipping the usual layer edit buffer)
  - *Time stamp destination* to adjust the field for time fix storage
  - *Log to GeoPackage/SpatiaLite…*: When activated, the user will be prompted to select an existing GeoPackage/SpatiaLite file or enter a new file name. A `gps_points` and `gps_tracks` table will be created in the file with a predefined structure.

    All incoming GPS messages will be logged to the `gps_points` layer, along with speed, bearing, altitude and accuracy information from the GPS.

    When the GPS is disconnected (or QGIS closed), the entire recorded GPS track will be added to the `gps_tracks` table (along with some calculated information like track length, start and end times).
  - *Log NMEA sentences…*: enables logging all raw NMEA strings from the device to a text file
  - *GPS settings…* to access the GPS *global options* dialog

> **Dica**
>
> **Live status bar information on position**
>
> When a GPS device is connected and the user moves the cursor over the map canvas, a live status bar message displays the distance and bearing from the cursor to the GPS position. Project distance and bearing settings are respected in this display. On a touch screen device, use a tap-and-hold event to trigger the live status bar message.

20.2.2 GPS Information Panel

For full monitoring of a live GPS tracking in QGIS, you may need to enable the *GPS Information Panel* (*View* ► *Panels* ► or press `Ctrl+0`).

In the top right corner of the *GPS Information Panel*, you press *Connect* to initiate connection between QGIS and a plugged-in GPS receiver, or *Disconnect* them.

In the top left of the panel, following buttons are accessible:

- Position: live details on the GPS position and sensors
- Sinal: força do sinal das conexões via satélite
- Settings: drop-down menu to *live tracking options* you may need to modify during a session

Posição e atributos adicionais

In the Position tab, if the GPS is receiving signals from satellites, you will see your position in latitude, longitude and altitude together with additional attributes.

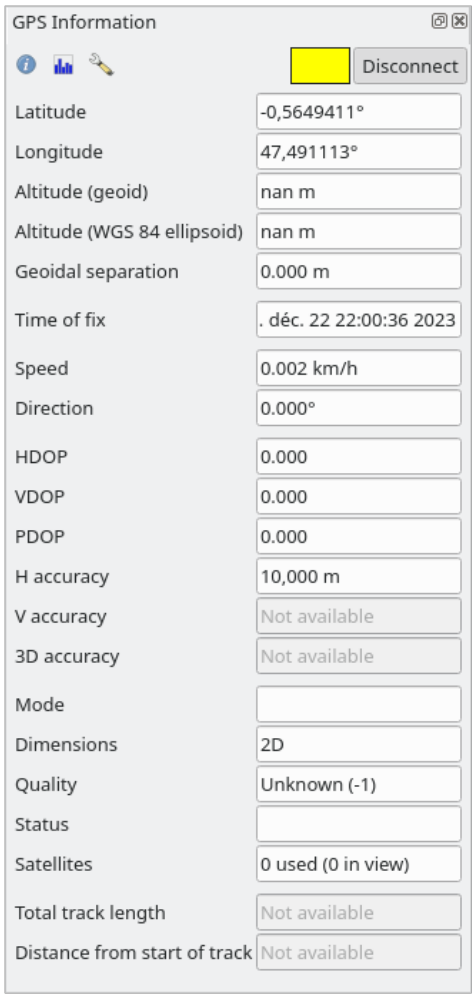


Figura20.3: Posição de rastreamento GPS e atributos adicionais

- *Latitude*
- *Longitude*

- *Altitude (Geoid)*: Altitude/elevation above or below the mean sea level
- *Altitude (WGS 84 ellipsoid)*: Altitude/elevation above or below the WGS-84 Earth ellipsoid
- *Geoidal separation*: Difference between the WGS-84 Earth ellipsoid and mean sea level (geoid), − means mean sea level is below ellipsoid
- *Time of fix*
- *Speed*: Ground speed
- *Direction*: Bearing measured in degrees clockwise from true north to the direction of travel
- *HDOP*: Horizontal dilution of precision
- *VDOP*: Vertical dilution of precision
- *PDOP*: Dilution of precision
- *H accuracy*: Horizontal accuracy in meters
- *V accuracy*: Vertical accuracy in meters
- *3D accuracy*: 3D Root Mean Square (RMS) in meters
- *Mode*: GPS receiver configuration 2D/3D mode, can be `automatic` or `manual`
- *Dimensions*: Position fix dimension, can be `2D`, `3D` or `No fix`
- *Quality*: Positioning quality indicator
- *Status*: Position fix status, can be `Valid` or `Invalid`
- *Satellites*: Count of satellites used in obtaining the fix
- *Total track length*: Total distance of current GPS track
- *Distance from start of track*: Direct distance from first vertex in current GPS track to latest vertex

Signal

With Signal tab, you can see the signal strength of the satellites you are receiving signals from.

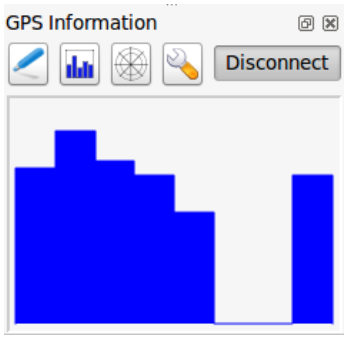


Figura20.4: Intensidade do sinal de rastreamento GPS

20.2.3 Conectar um GPS com Bluetooth para rasteamento ao vivo

Com QGIS você pode conectar um GPS via Bluetooth para a coleta de dados em campo. Para executar esta tarefa, você precisa de um dispositivo GPS com Bluetooth e um receptor Bluetooth em seu computador.

Em primeiro lugar você deve deixar seu aparelho de GPS ser reconhecido e emparelhado com o computador. Ligue o GPS, vá até o ícone Bluetooth na área de notificação e busca de um novo dispositivo.

No lado direito da máscara de seleção de dispositivos, verifique se todos os dispositivos estão selecionados, para que sua unidade GPS provavelmente apareça entre os disponíveis. Na próxima etapa, um serviço de conexão serial deve estar disponível, selecione-o e clique em :guilabel:botão *Configure*.

Lembre-se o número da porta COM atribuído à conexão GPS como resultante pelas propriedades Bluetooth.

Após o reconhecimento do GPS, faça o emparelhamento para a conexão. Normalmente, o código de autorização é `0000`.

Agora abra :guilabel:painel *informações de GPS* e alterne para **|opções|** Tela de opções de GPS. Selecione a porta COM atribuída à conexão GPS e clique em *Conectar*. Depois de um tempo, um cursor indicando sua posição deve aparecer.

Se QGIS não pode receber dados do GPS, então você deve reiniciar seu dispositivo GPS, espere 5-10 segundos, em seguida, tente se conectar novamente. Normalmente este trabalho soluciona. Se você receber um erro de conexão novamente certifique-se que você não tem outro receptor Bluetooth perto de você, emparelhado com a mesma unidade GPS.

20.2.4 Examples of GPS devices connection in QGIS

Utilizando GPSMAP 60cs

MS Windows

Easiest way to make it work is to use a middleware (freeware, not open) called GPSGate.

Launch the program, make it scan for GPS devices (works for both USB and BT ones) and then in QGIS:

1. In *Settings* ► *Options* ► *GPS*, detect the plugged-in device. You can use the *Autodetect* mode.
2. Press *Connect* in the *GPS Information* panel

Ubuntu/Mint GNU/Linux

Como para Windows a maneira mais fácil é usar um servidor no meio, neste caso GPSD, assim

1. Install the program

   ```
   sudo apt install gpsd
   ```

2. Em seguida, carregar o módulo do kernel `garmin_gps`

   ```
   sudo modprobe garmin_gps
   ```

3. And then connect the unit and check with `dmesg` the actual device being used by the unit, for example `/dev/ttyUSB0`.
4. Now you can launch gpsd

   ```
   gpsd /dev/ttyUSB0
   ```

5. E, finalmente, conectar-se com a ferramenta de acompanhamento ao vivo do QGIS.

Usando BTGP-38KM datalogger (somente Bluetooth)

Usando GPSD (no Linux) ou GPSGate (no Windows) é fácil.

Usando datalogger BlueMax GPS-4044 (ambos BT e USB)

MS Windows

The live tracking works for both USB and BT modes, by using GPSGate or even without it, just use the *Autodetect* mode, or point the tool to the right port.

Ubuntu/Mint GNU/Linux

Para USB

O acompanhamento ao vivo funciona tanto com GPSD

```
gpsd /dev/ttyACM3
```

ou sem ela, ligando a ferramenta de acompanhamento ao vivo QGIS diretamente para o dispositivo (por exemplo, `/dev/ttyACM3`).

Para Bluetooth

O acompanhamento ao vivo funciona tanto com GPSD

```
gpsd /dev/rfcomm0
```

ou sem ela, ligando a ferramenta de acompanhamento ao vivo QGIS diretamente para o dispositivo (por exemplo, `/dev/rfcomm0`).

CAPÍTULO 21

Sistema de Autenticação

21.1 Visão Geral do Sistema de Autenticação

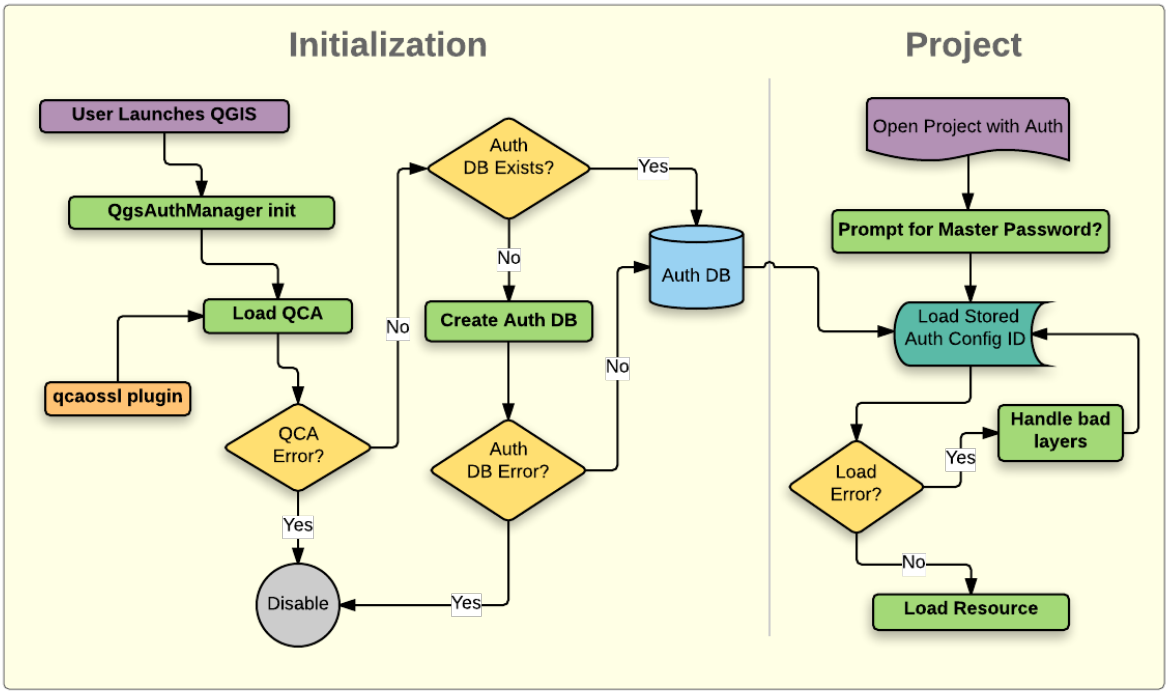


Figura21.1: Anatomia do sistema de autenticação

21.1.1 Banco de Dados de Autenticação

The authentication system stores authentication configurations in an SQLite database file located, by default, at `<profile directory>/qgis-auth.db`.

Este banco de dados de autenticação pode ser movido entre as instalações do QGIS sem afetar outras preferências do usuário QGIS atuais, já que é completamente separado das configurações normais do QGIS. Um ID de configuração (uma strig alfanumérica aleatória de 7 caracteres) é gerado ao armazenar inicialmente uma configuração no banco de dados. Isso representa a configuração, permitindo que o ID seja armazenado em componentes de aplicativo de texto simples (como arquivos de projeto, complemento ou configurações) sem a divulgação de suas credenciais associadas.

Nota

O diretório pai do *qgis-auth.db* pode ser definido usando a seguinte variável de ambiente,``QGIS_AUTH_DB_DIR_PATH``, ou definido na linha de comando durante o lançamento com a opção `--authdbdirectory`.

21.1.2 Custom authentication databases

QGIS can be configured to use a custom authentication database instead of the above mentioned default SQLite one: any database suppported by the Qt SQL module can be used (e.g. PostgreSQL, MySQL, etc), provided that the corresponding Qt SQL driver is available in the system.

This can be useful in scenarios where a user wants to share the same authentication database between multiple QGIS installations, or when a user wants to use a different authentication database than the default SQLite one or when a centralized authentication database is used by QGIS server.

The only way to configure a custom authentication database is by setting the `QGIS_AUTH_DB_URI` environment variable to the URI of the connection, the URI is in the form of `driver://username:password@hostname:port/database?options`.

Where:

- `driver` is the name of the Qt SQL driver to use, e.g. `QPSQL` for PostgreSQL, `QMYSQL` for MySQL, etc.
- `username` is the username to use to connect to the database
- `password` is the password to use to connect to the database
- `hostname` is the hostname of the database server
- `port` is the port of the database server
- `database` is the name of the database to use
- `options` are the options to pass to the driver, e.g. `sslmode=require` for PostgreSQL

Nota

The `schema` can be specified in the URI options, e.g. `QPSQL://username:password@hostname:port/database?schema=schema_name`

The database must exist before starting QGIS, and the user must have the necessary permissions to connect to the database and to create the required tables if they do not exist.

Aviso

The password in the URI is stored in plain text in the environment variable, so it is recommended to use a passwordless user or a user with limited permissions to connect to the database.

Aviso

Any database not based on SQLite is considered to be read-only (this can be changed by Python plugins if necessary).

This is an advanced feature, designed to allow one or more custom authentication databases or even custom Python implementations of credentials storages to be used by QGIS.

The system is also designed to allow for multiple authentication databases to be used but there is currently no user facing interface to manage multiple credential storages so its usage requires manual configuration and management, typically from a Python plugin.

21.1.3 Senha mestra

Para armazenar ou acessar informações confidenciais no banco de dados, o usuário deve definir uma *senha mestra*. Uma nova senha mestra é solicitada e verificada ao armazenar inicialmente quaisquer dados criptografados no banco de dados. Quando informações confidenciais são acessadas, o usuário é solicitado a fornecer a senha mestra. A senha é então armazenada em cache pelo restante da sessão (até que o aplicativo seja encerrado), a menos que o usuário escolha manualmente uma ação para limpar seu valor em cache. Algumas instâncias de uso do sistema de autenticação não exigem a entrada da senha mestra, como ao selecionar uma configuração de autenticação existente ou ao aplicar uma configuração a uma configuração de servidor (como ao adicionar uma camada WMS).

Você pode optar por salvar a senha na `Carteira/Chaveiro` do seu computador.

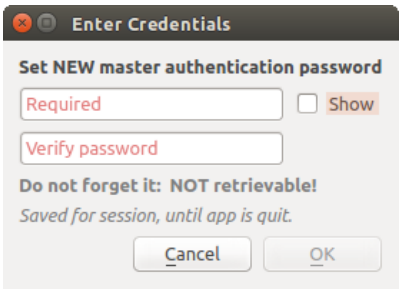


Figura21.2: Insira uma nova senha mestra

Nota

Um caminho para um arquivo contendo a senha mestra pode ser definido usando a seguinte variável de ambiente, `QGIS_AUTH_PASSWORD_FILE`.

Gerenciando a senha mestre

Uma vez definida, a senha mestra pode ser redefinida; a senha mestra atual será necessária antes da redefinição. Durante este processo, existe a opção de gerar um backup completo do banco de dados atual.

Figura21.3: Redefinindo a senha mestre

Se o usuário esquecer a senha mestra, não haverá como recuperá-la ou substituí-la. Também não há meio de recuperar informações criptografadas sem saber a senha mestra.

Se um usuário inserir sua senha existente incorretamente três vezes, a caixa de diálogo se oferecerá para apagar o banco de dados.

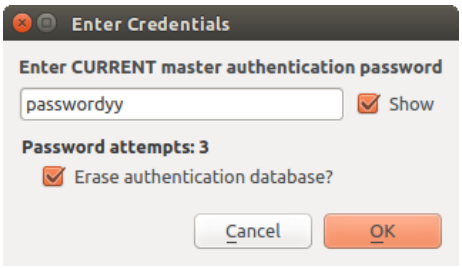


Figura21.4: Solicitação de senha após três tentativas inválidas

21.1.4 Configurações de Autenticação

Você pode gerenciar as configurações de autenticação de *Configurações* na guia *Autenticação* da caixa de diálogo Opções do QGIS (*Configurações* ► *Opções*).

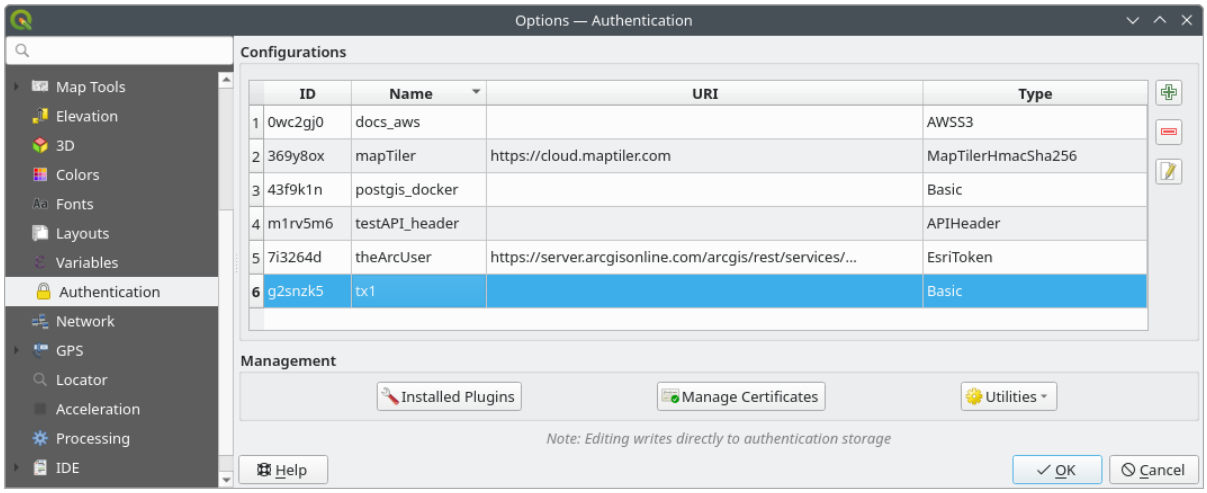


Figura21.5: Editor de configurações

To create a new authentication configuration:

1. Press the Add a new authentication configuration
2. Provide a *Name* and optional *Resource* URL
3. A seven-characters *ID* will be assigned to the configuration for identification purpose. It can be customized after you pressed the Unlock to edit ID button. Be aware that updating that ID may break things.
4. Select the *target method* and fill the corresponding connection details
5. Once done, click *Save* to save the configuration. The saved configuration appears in the list of authentication configurations and can be used with supported connections.

The saved configuration can later be edited using Edit selected configuration or removed using Delete selected configuration.

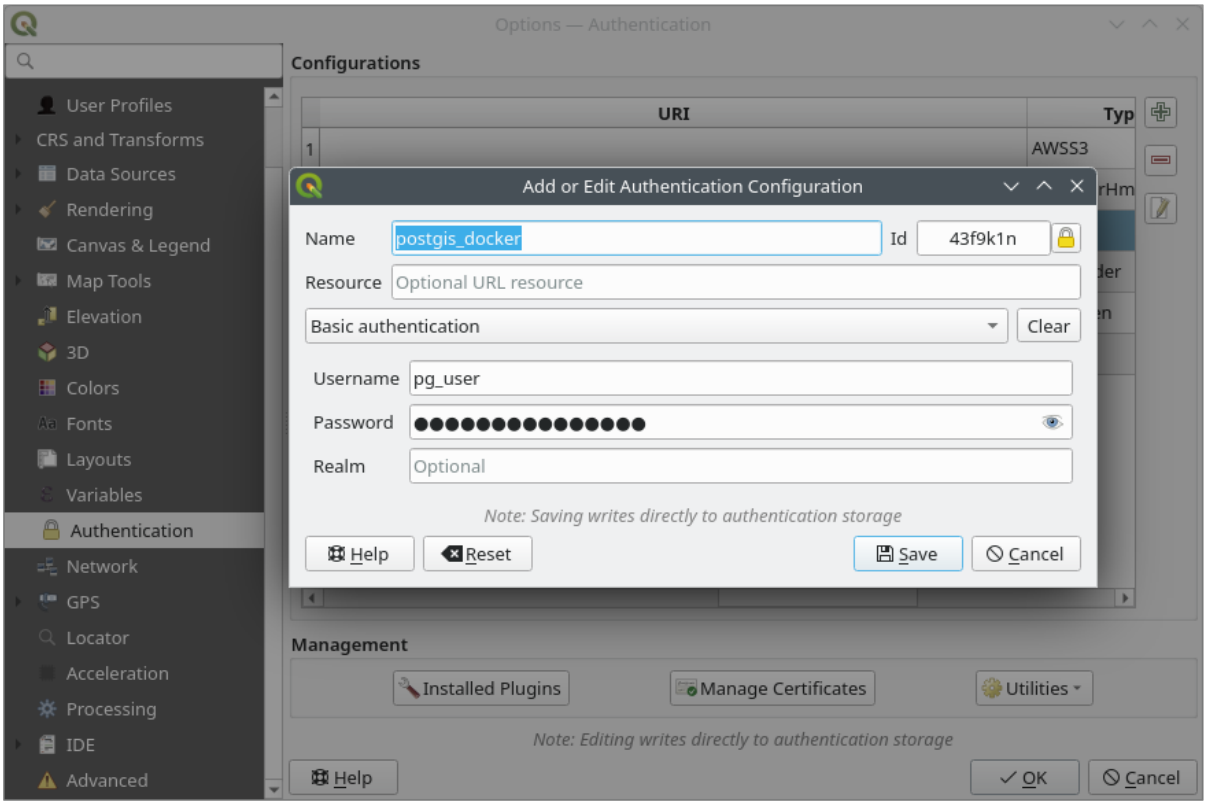


Figura21.6: Adicionando configuração de dentro do editor de configuração

The same type of operations for authentication configuration management (Add, Edit and Remove) can be done when configuring a given service connection, such as configuring an OWS service connection. For that, there are action buttons within the *configuration selector* for fully managing configurations found within the authentication database. In this case, there is no need to go to the *configurations* in *Authentication* tab of QGIS options unless you need to do more comprehensive configuration management.

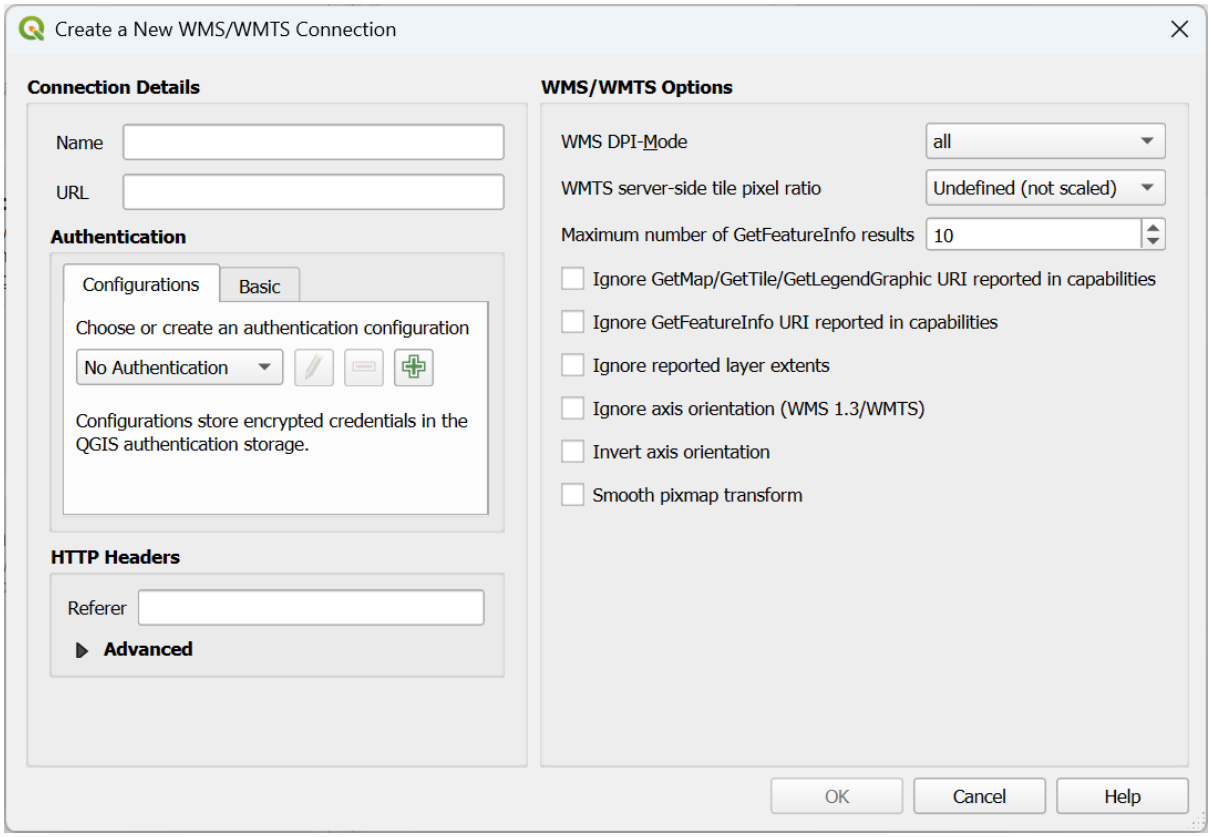


Figura21.7: WMS connection dialog showing authentication configuration widget

Ao criar ou editar uma configuração de autenticação, a informação necessária é um nome, um método de autenticação e qualquer outra informação que o método de autenticação requeira (veja mais sobre os tipos de autenticação disponíveis em:ref:*autenticação_métodos*).

21.1.5 Métodos de autenticação

As autenticações disponíveis são fornecidas pelos plugins C++ da mesma forma que os plugins do provedor de dados são suportados pelo QGIS. O método de autenticação que pode ser selecionado é relativo ao acesso necessário para o recurso/provedor, por exemplo. HTTP(S) ou banco de dados, e se há suporte no código QGIS e em um plugin. Como tal, alguns plugins de métodos de autenticação podem não ser aplicáveis em todos os lugares em que um seletor de configuração de autenticação é mostrado. Uma lista de plugins de métodos de autenticação disponíveis e seus recursos/provedores compatíveis podem ser acessados em: seleção de menu:*Configurações –> Opções* e, na aba: guilabel:*Autenticação*, clique em |opções|:guilabel:botão *Plugins Instalados*.

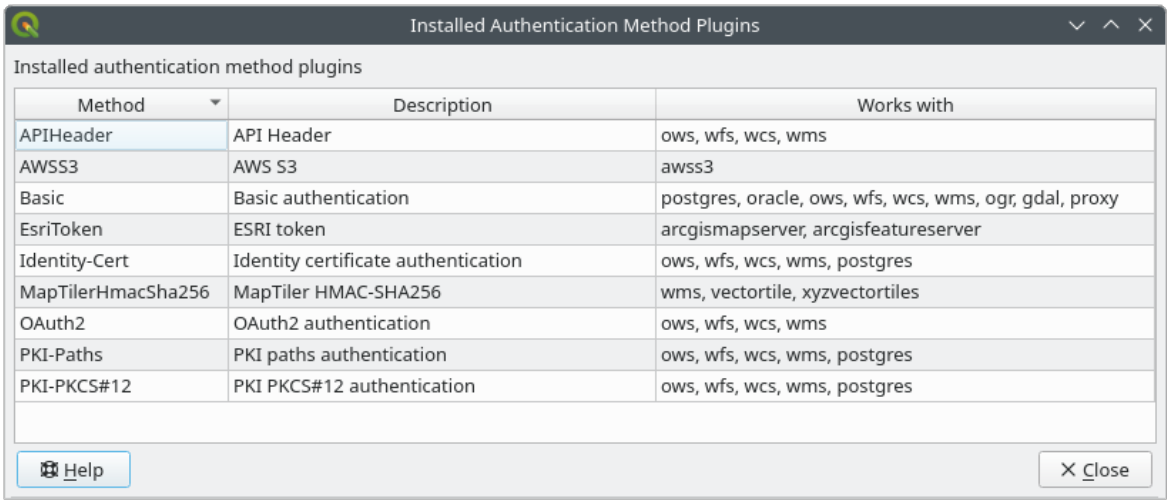


Figura21.8: Lista de plugins de métodos disponíveis

Plugins podem ser criados para novos métodos de autenticação que não requerem que o QGIS seja recompilado. Como o suporte para plugins é atualmente somente C++, o QGIS precisará ser reiniciado para que o novo plugin fique disponível para o usuário. Certifique-se de que seu plug-in seja compilado na mesma versão de destino do QGIS se você pretende adicioná-lo a uma instalação de destino existente.

API Header Authentication

The API Header authentication method allows you to connect to services that require custom HTTP headers, such as API keys. For example, if a WMS service requires an API key sent as a header (e.g. `X-API-KEY`), you can use this method to provide it.

You can add one or more headers with the button; provide a *Header Name* and *Header Value* for each entry. Use to delete the selected header or the *Clear* button to remove all entries.

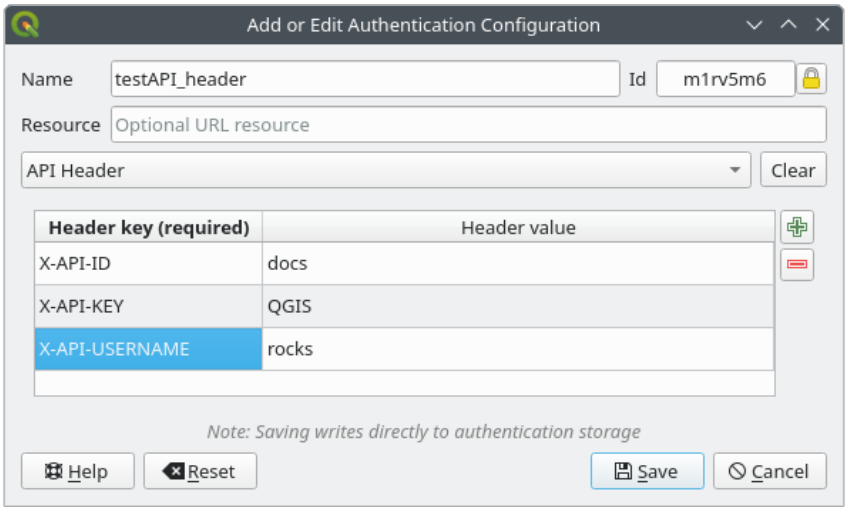


Figura21.9: API Header authentication configs

AWS S3 Authentication

The AWS S3 authentication method allows you to connect to Amazon S3 resources by providing a *Name*, *Resource*, *Username*, *Password*, and *Region*.

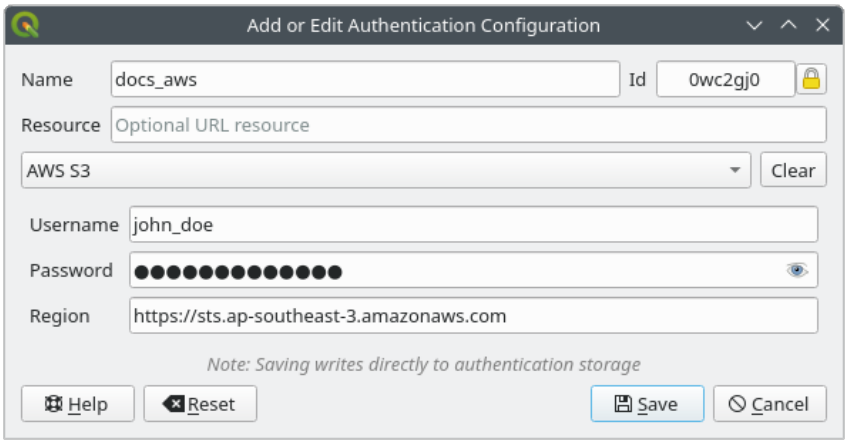


Figura21.10: AWS S3 authentication configs

Basic Authentication

The Basic authentication method is used for services that require standard HTTP authentication with a *Username*, *Password* and *Realm*.

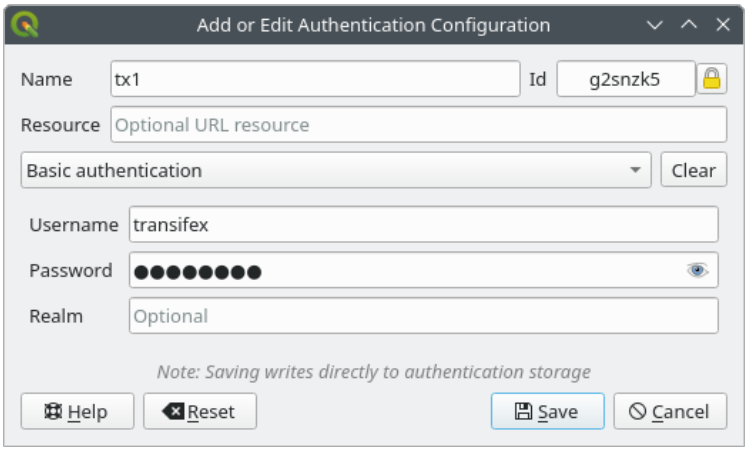


Figura21.11: Configurações básicas de autenticação HTTP

ESRI Token Authentication

The ESRI token authentication method is used for *ArcGIS REST Servers* that require token-based authentication.

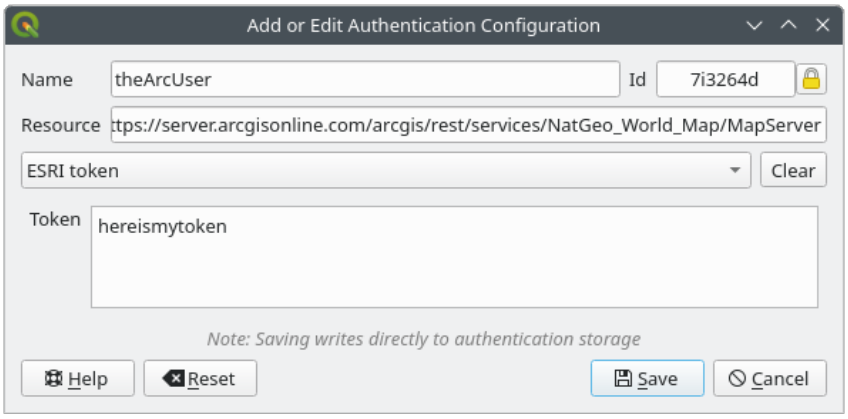


Figura21.12: Configurações de autenticação de símbolo ESRI

Identity certificate authentication

The Identity certificate authentication method allows you to connect using a client *identity certificate*.

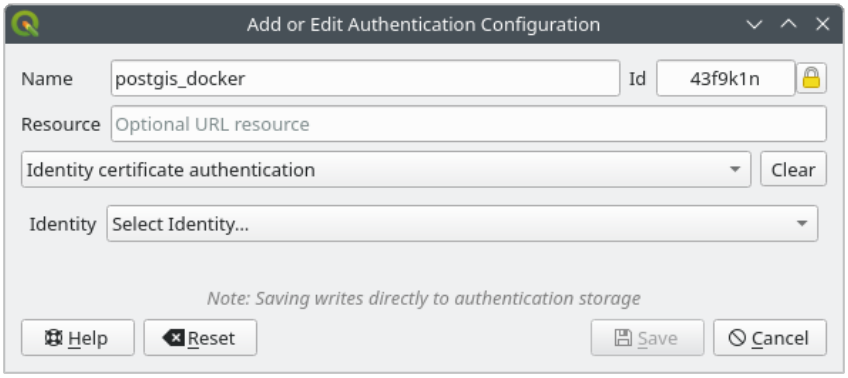


Figura21.13: Identity-cert authentication configs

MapTiler HMAC-SHA256 Authentication

The MapTiler HMAC-SHA256 authentication method is used to connect to MapTiler services that require HMAC-SHA256 authorization.

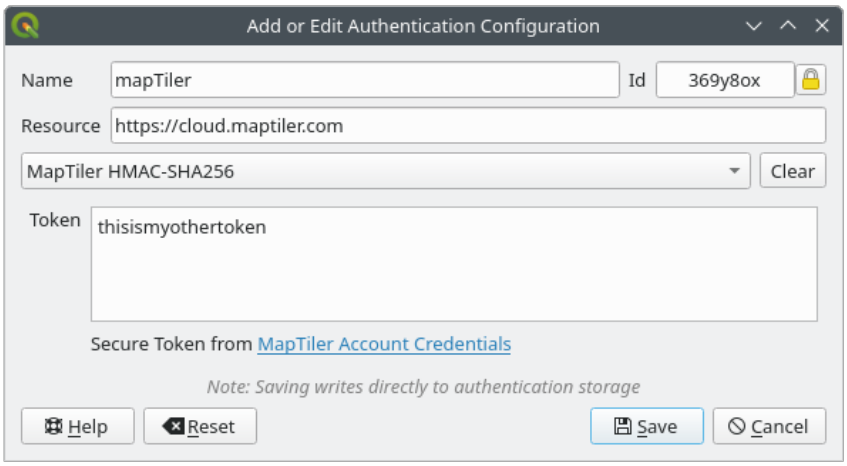


Figura21.14: MapTiler HmacSha256 authentication configs

OAuth2 Authentication

The OAuth2 authentication method is used to connect to services that require OAuth2 2.0 authorization, allowing secure access using client credential and token-based authentication.

It supports using `localhost` as the redirect host, providing compatibility with services that do not accept `http://127.0.0.1` redirects (for example, Microsoft SharePoint).

An advanced OAuth2 configuration option, *Extra token header(s)*, allows attaching additional tokens returned by the OAuth2 token endpoint as HTTP(S) request headers.

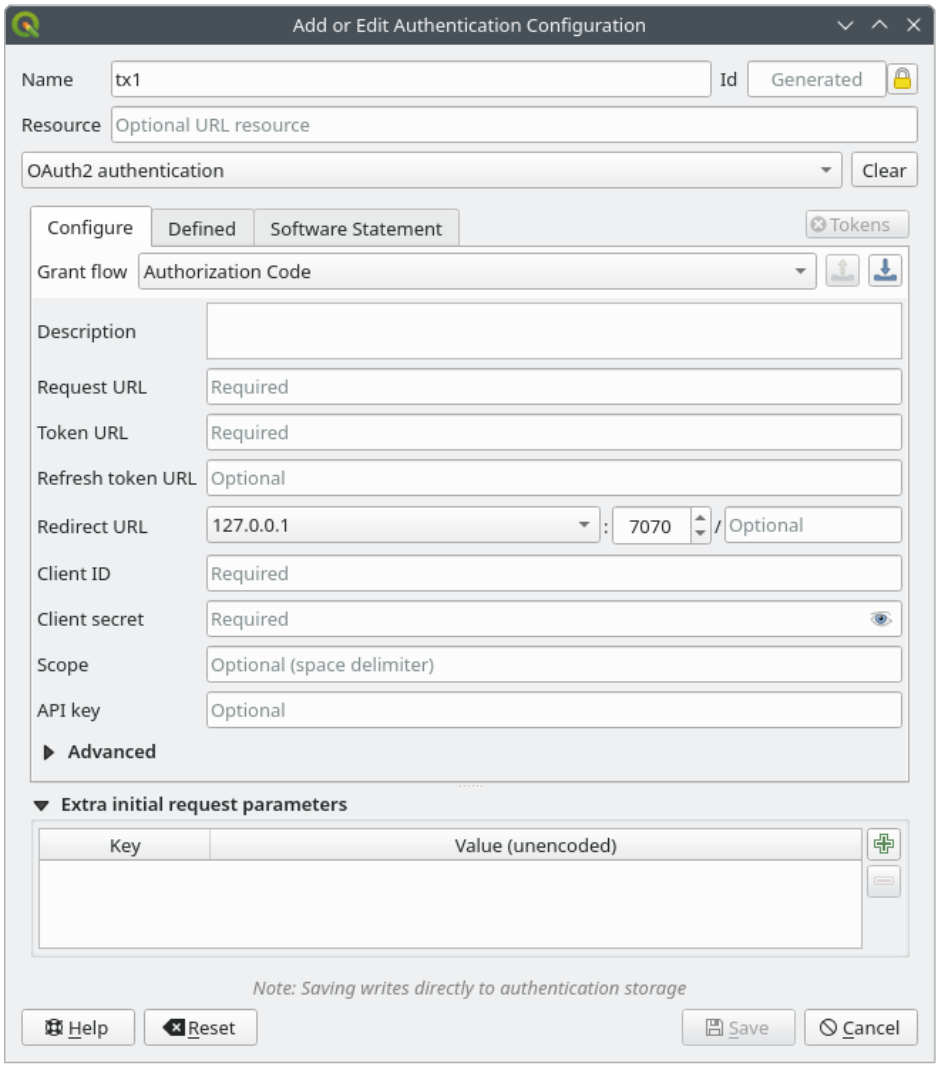


Figura21.15: Configurações de autenticação OAuth2

PKI-Paths Authentication

The PKI-Paths authentication method allows you to connect using separate certificate and key files stored on your system.

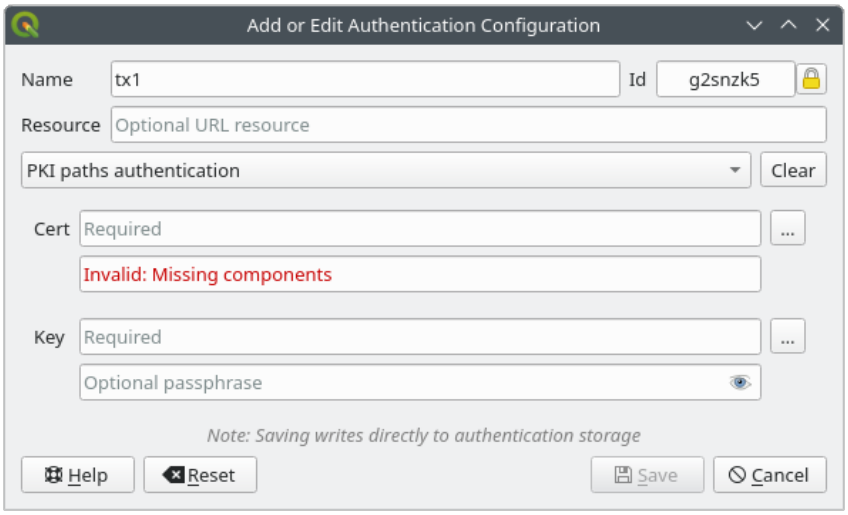


Figura21.16: Configurações de autenticação de caminhos PKI

PKI-PKCS#12 Authentication

The PKI-PKCS#12 authentication method allows you to connect using a single bundle file containing both the certificate and private key, with an optional passphrase for the key.

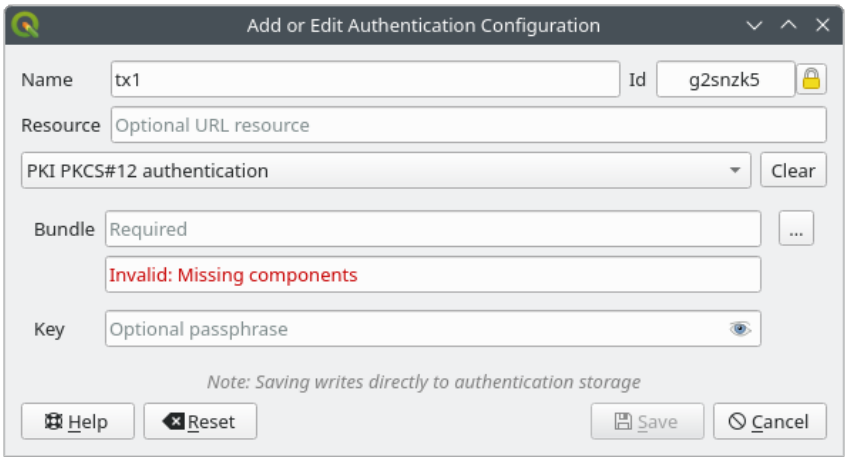


Figura21.17: Configurações de autenticação de caminhos de arquivo PKI PKCS#12

21.1.6 Utilitários de configuração de senha mestra e autenticação

No menu Opções (:seleção de menus: *Configurações –> Opções*) na aba: guilabel: *Autenticação*, existem várias ações de utilitários para gerenciar o banco de dados de autenticação e as configurações:

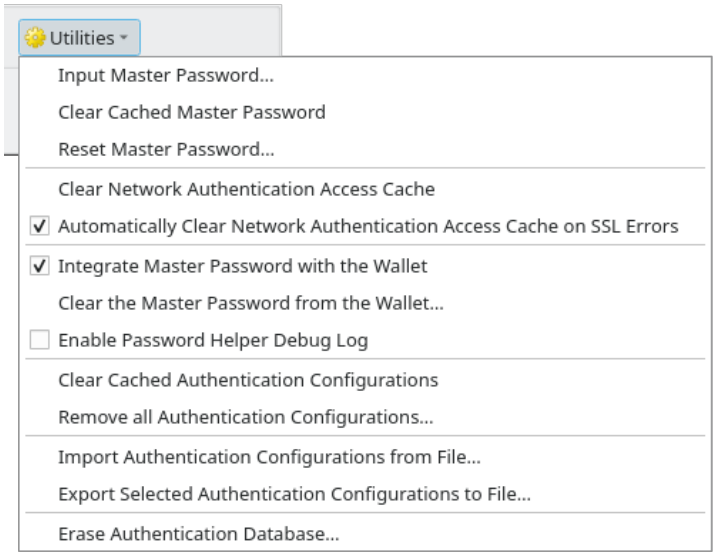


Figura21.18: Menu de utilitários

- *Input master password…*: opens the master password input dialog, independent of performing any authentication database command
- *Clear cached master password*: unsets the master password if it has been set
- *Reset master password…*: opens a dialog to change the master password (the current password must be known) and optionally back up the current database
- *Clear network authentication access cache*: clears the authentication cache of all connections
- *Automatically clear network authentication access cache on SSL errors*: the connection cache stores all authentication data for connections, also when the connection fails. If you change authentication configurations or certification authorities, you should clear the authentication cache or restart QGIS. When this option is checked, the authentication cache will be automatically cleared every time an SSL error occurs and you choose to abort the connection
- *Integrate master password with the Wallet*: adds the master password to your personal Wallet/Keyring
- *Clear the master password from the Wallet…*: deletes the master password from your Wallet/Keyring
- *Enable password helper debug log*: enables a debug tool that will contain all the log information of the authentication methods
- *Clear cached authentication configurations*: clears the internal lookup cache for configurations, used to speed up network connections. This does not clear QGIS's core network access manager's cache, which requires a relaunch of QGIS.
- *Remove all authentication configurations…*: clears the database of all configuration records, without removing other stored records.
- *Import authentication configurations from file…*: imports from an `.XML` file details for creating custom authentication configurations.
- *Export selected authentication configurations to file…*: exports the selected items to a possibly encrypted `.XML` file.

- *Erase authentication database…*: schedules a backup of the current database and complete rebuild of the database table structure. The actions are scheduled for a later time, to ensure that other operations, like project loading, do not interrupt the operation or cause errors due to a temporarily missing database.

21.1.7 Usando configurações de autenticação

Normalmente, uma configuração de autenticação é selecionada em uma caixa de diálogo de configuração para serviços de rede (como WMS). No entanto, a ferramenta seletora pode ser incorporada em qualquer lugar que a autenticação seja necessária ou em funcionalidades não essenciais, como em plug-ins PyQGIS ou C++ de terceiros.

When using the selector, *No authentication* is displayed in the pop-up menu control when nothing is selected, when there are no configurations to choose from, or when a previously assigned configuration can no longer be found in the database. Use the drop-down menu to select an existing authentication configuration or press

to create a configuration you could use. More details at *Autenticação*.

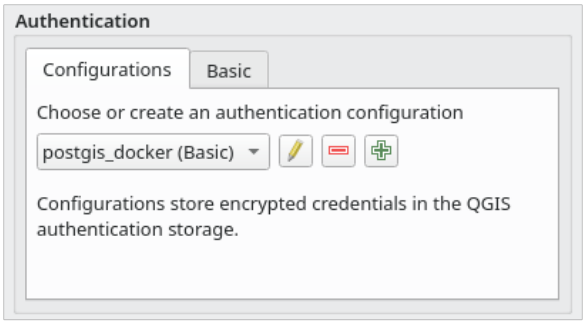


Figura21.19: Seletor de configuração de autenticação com configuração selecionada

21.1.8 Vinculações do Python

All classes and public functions have sip bindings, except QgsAuthCrypto, since management of the master password hashing and auth database encryption should be handled by the main app, and not via Python.

Once the master password is entered, the API is open to access authentication configs in the authentication database, similar to how Firefox works. However, no wall against PyQGIS access has been defined. This may lead to issues where a user downloads/installs a malicious PyQGIS plugin or standalone app that gains access to authentication credentials.

21.2 Fluxos de trabalho de autenticação de usuário

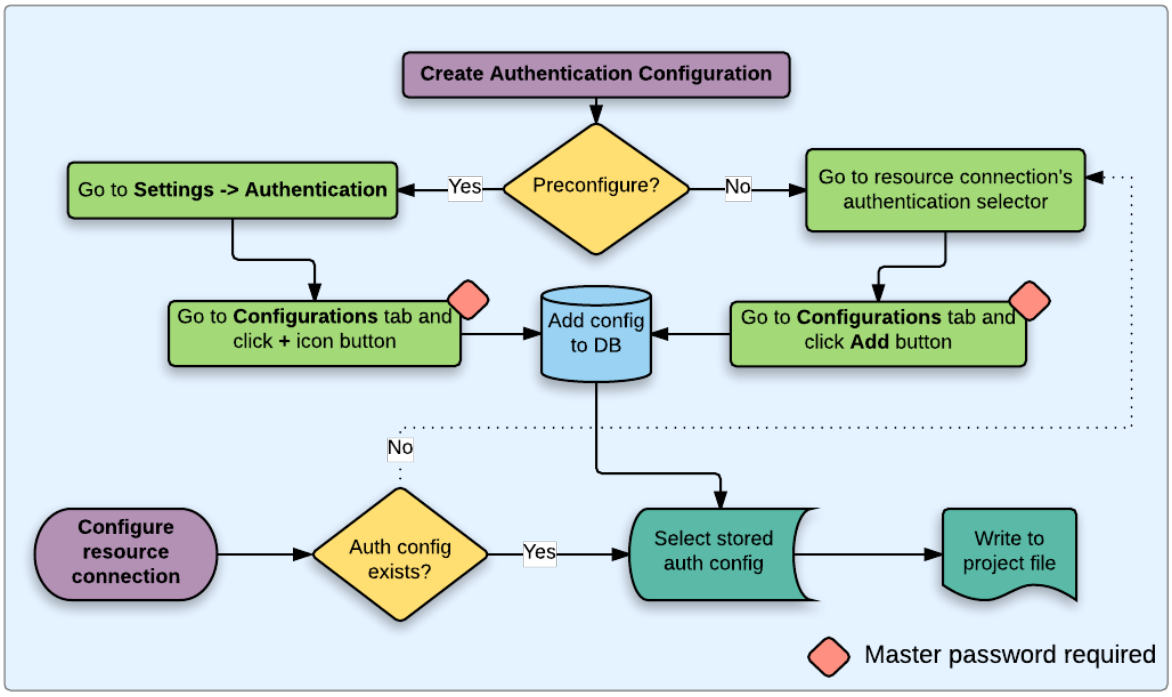


Figura21.20: Fluxos de trabalho de usuário genérico

21.2.1 Autenticação HTTP(S)

Uma das conexões de recursos mais comuns é via HTTP(S), por exemplo. servidores de mapeamento da web e plug-ins de métodos de autenticação geralmente funcionam para esses tipos de conexões. Os plug-ins de método têm acesso ao objeto de solicitação HTTP e podem manipular tanto a solicitação quanto seus cabeçalhos. Isso permite muitas formas de autenticação baseada na Internet. Ao conectar via HTTP(S) usando o método de autenticação padrão de nome de usuário/senha, tentará a autenticação HTTP Básico na conexão.

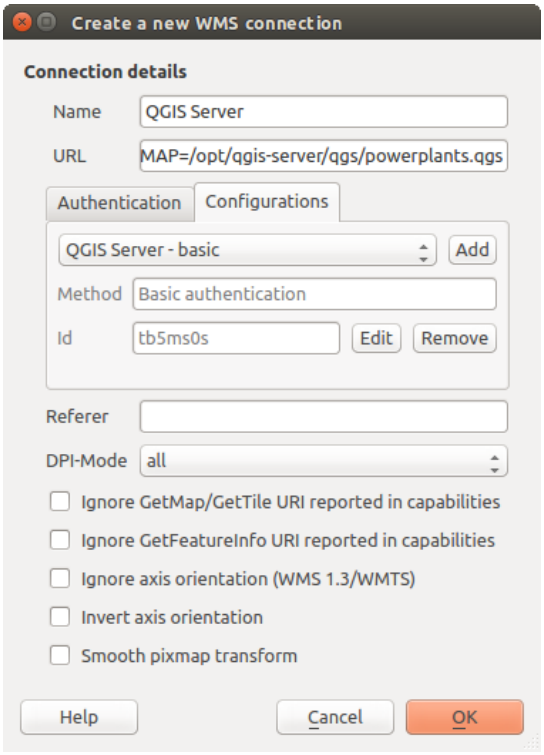


Figura21.21: Configurando uma conexão WMS para um HTTP básico

21.2.2 Autenticação de Base de Dados

Conexões com recursos de banco de dados são geralmente armazenadas como pares `chave=valor`, que irão expor nomes de usuários e (opcionalmente) senhas, se *não* estiver usando uma configuração de autenticação. Ao configurar com o sistema de autenticação, o `chave=valor` será uma representação abstrata das credenciais, por exemplo, `authfg=81t21b9`.

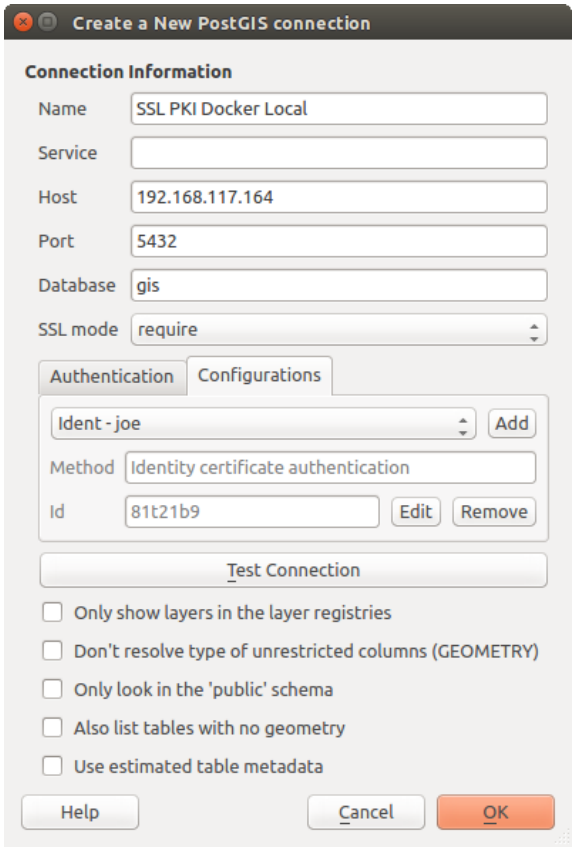


Figura21.22: Configurando uma conexão Postgres SSL com PKI

21.2.3 Autenticação de PKI

Ao configurar componentes PKI no sistema de autenticação, você tem a opção de importar componentes para o banco de dados ou referenciar arquivos de componentes armazenados em seu sistema de arquivos. O último pode ser útil se esses componentes forem alterados com frequência ou quando os componentes forem substituídos por um administrador do sistema. Em qualquer caso, você precisará armazenar qualquer senha necessária para acessar as chaves privadas no banco de dados.

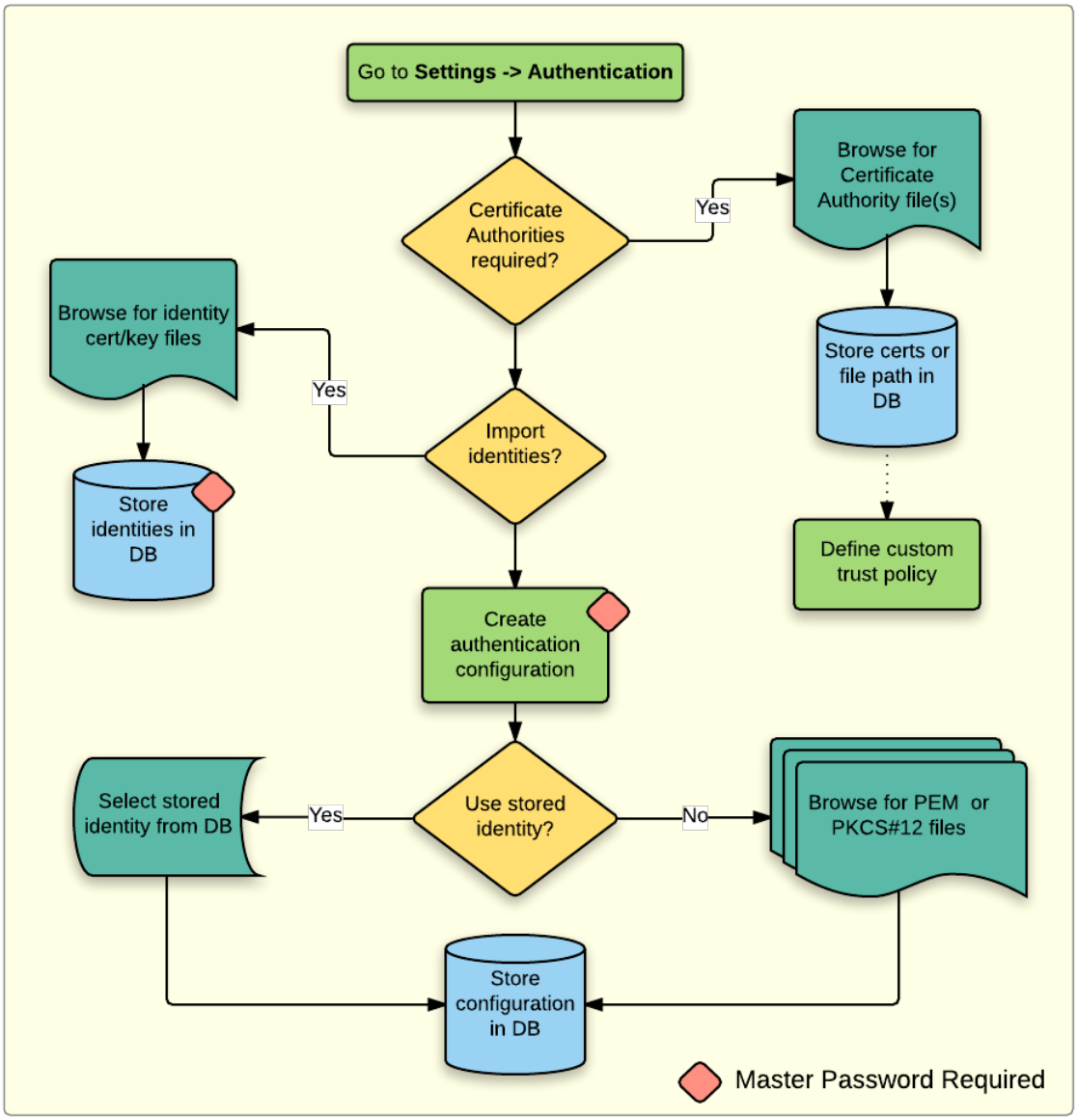


Figura21.23: Fluxo de trabalho de configuração de PKI

Todos os componentes PKI podem ser gerenciados em editores separados dentro do **Gerenciador de Certificados**, que pode ser acessado na aba:guilabel:*Autenticação* na caixa de diálogo *Opções* do QGIS (:seleção de menus: *Configurações –> Opções*) por clicando no botão:guilabel:*Gerenciar Certificados*

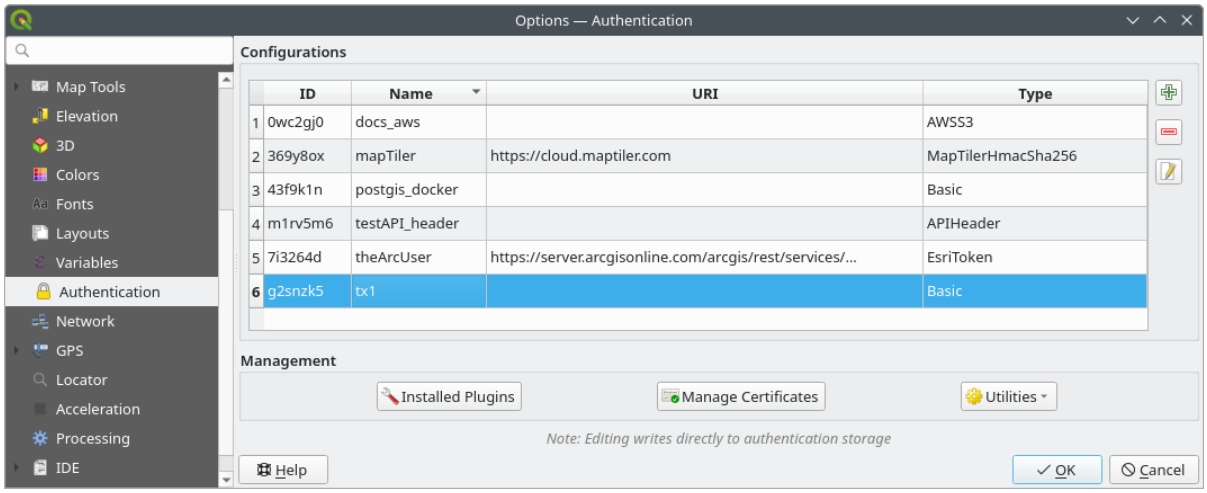


Figura21.24: Abrindo o Gerenciador de Certificados

No:guilabel:*Gerenciador de certificados*, existem editores para **Identidades**, **Servidores** e **Autoridades**. Cada um deles está contido em suas próprias guias e são descritos abaixo na ordem em que são encontrados no gráfico de fluxo de trabalho acima. A ordem de tabulação é relativa aos editores acessados com frequência quando você estiver acostumado com o fluxo de trabalho.

> **Nota**
>
> Como todas as edições do sistema de autenticação são gravadas imediatamente no banco de dados de autenticação, não há necessidade de clicar na caixa de diálogo:guilabel:*Opções OK* para que quaisquer alterações sejam salvas. Isso é diferente de outras configurações na caixa de diálogo Opções.

Autoridades

Você pode gerenciar as Autoridades de Certificação (CAs) disponíveis na guia **Autoridades** no **Gerenciador de certificados** na guia **Autenticação** da caixa de diálogo **Opções** do QGIS.

Conforme mencionado no gráfico de fluxo de trabalho acima, a primeira etapa é importar ou fazer referência a um arquivo de CAs. Esta etapa é opcional e pode ser desnecessária se sua cadeia de confiança de PKI se originar de CAs raiz já instaladas em seu sistema operacional (SO), como um certificado de um fornecedor de certificado comercial. Se sua autoridade de certificação raiz de autenticação não estiver nas autoridades de certificação raiz confiáveis do sistema operacional, ela precisará ser importada ou ter o caminho do sistema de arquivos referenciado. (Entre em contato com o administrador do sistema se não tiver certeza.)

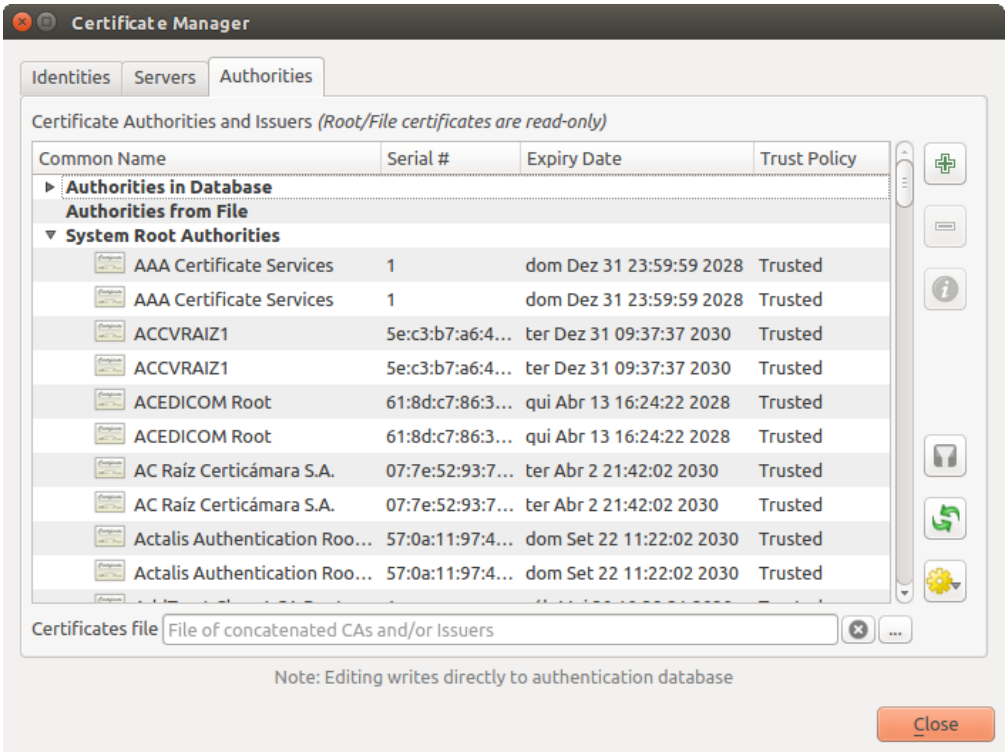


Figura21.25: Editor de autoridades

Por padrão, as CAs raiz do seu sistema operacional estão disponíveis; no entanto, suas configurações de confiança não são herdadas. Você deve revisar as configurações da política de confiança do certificado, especialmente se as CAs raiz do SO tiveram suas políticas ajustadas. Qualquer certificado expirado será definido como não confiável e não será usado em conexões de servidor seguras, a menos que você substitua especificamente sua política de confiança. Para ver a cadeia de confiança detectável pelo QGIS para qualquer certificado, selecione-o e clique em **|metadados|** Mostrar informações do certificado.

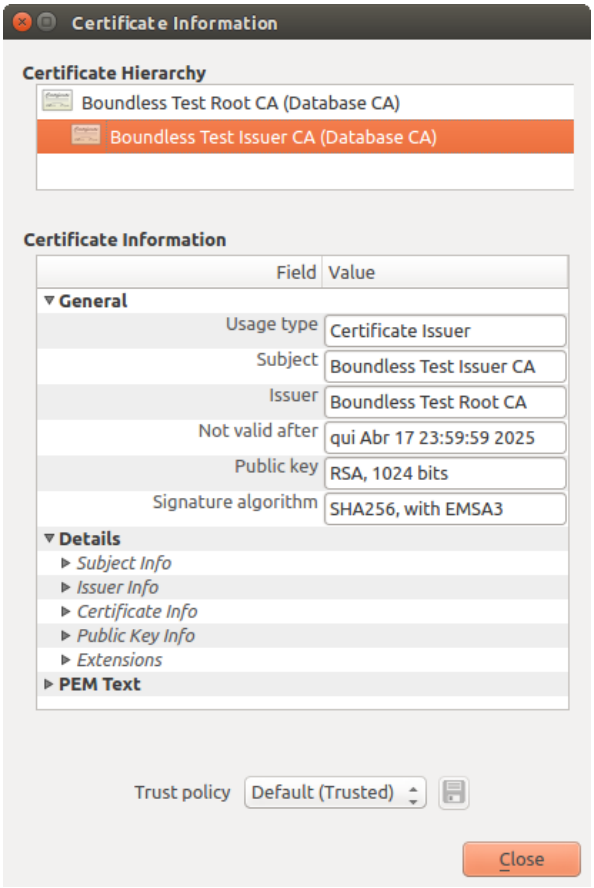


Figura21.26: Caixa de diálogo de informações do certificado

Você pode editar a:guilabel:*Política de confiança* **|selecione String|** para qualquer certificado selecionado dentro da cadeia. Qualquer alteração na política de confiança para um certificado selecionado não será salva no banco de dados, a menos que **|Salvar arquivo|**:sup:O botão *Salvar alteração da política de confiança do certificado no banco de dados* é clicado *por* certificação selecionada. Fechar a caixa de diálogo **não** aplicará as alterações de política.

Você pode revisar as CAs filtradas, certificados intermediários e raiz, que serão confiáveis para conexões seguras ou alterar a política de confiança padrão clicando em **|Configurações de transformação|** botão **Opções**.

> **Aviso**
>
> Alterar a política de confiança padrão pode resultar em problemas com conexões seguras.

Você pode importar CAs ou salvar um caminho do sistema de arquivos de um arquivo que contém várias CAs ou importar CAs individuais. O formato PEM padrão para arquivos que contêm várias certificações de cadeia de CA tem o certificado raiz na parte inferior do arquivo e todos os certificados filho assinados posteriormente acima, no início do arquivo.

A caixa de diálogo de importação de certificados de CA encontrará todos os certificados de CA dentro do arquivo, independentemente da ordem, e também oferece a opção de importar certificados que são considerados inválidos (caso você queira substituir sua política de confiança). Você pode substituir a política de confiança na importação ou fazê-lo posteriormente no editor **Autoridades**.

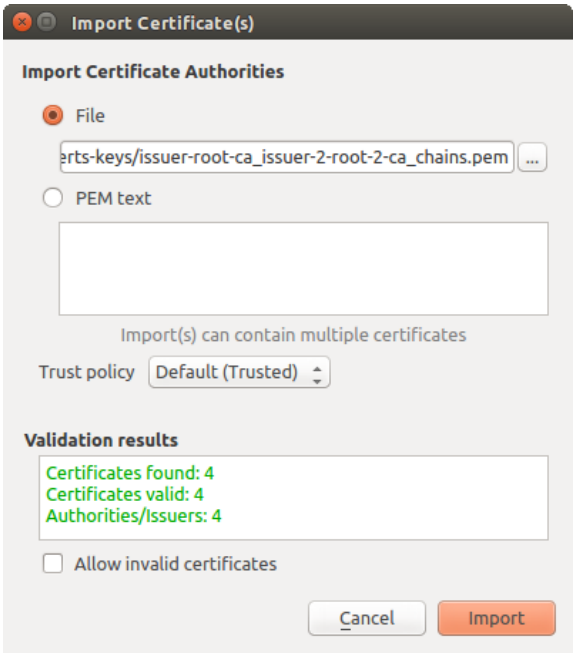


Figura21.27: Caixa de diálogo Importar certificados

Nota

Se você estiver colando informações de certificado no campo:guilabel:*PEM texto*, observe que certificados criptografados não são suportados.

Identidades

Você pode gerenciar pacotes de identidade de cliente disponíveis na guia: guilabel: *Identidades* no:guilabel:*Gerenciador de certificados* na guia **Autenticação** da caixa de diálogo **Opções** do QGIS. Uma identidade é o que o autentica em relação a um serviço habilitado para PKI e geralmente consiste em um certificado de cliente e uma chave privada, como arquivos separados ou combinados em um único arquivo "agregado". O pacote ou chave privada geralmente é protegido por senha.

Depois de importar quaisquer Autoridades de Certificação (CAs), você pode importar opcionalmente quaisquer pacotes de identidade para o banco de dados de autenticação. Se você não desejar armazenar as identidades, poderá fazer referência aos caminhos do sistema de arquivos do componente em uma configuração de autenticação individual.

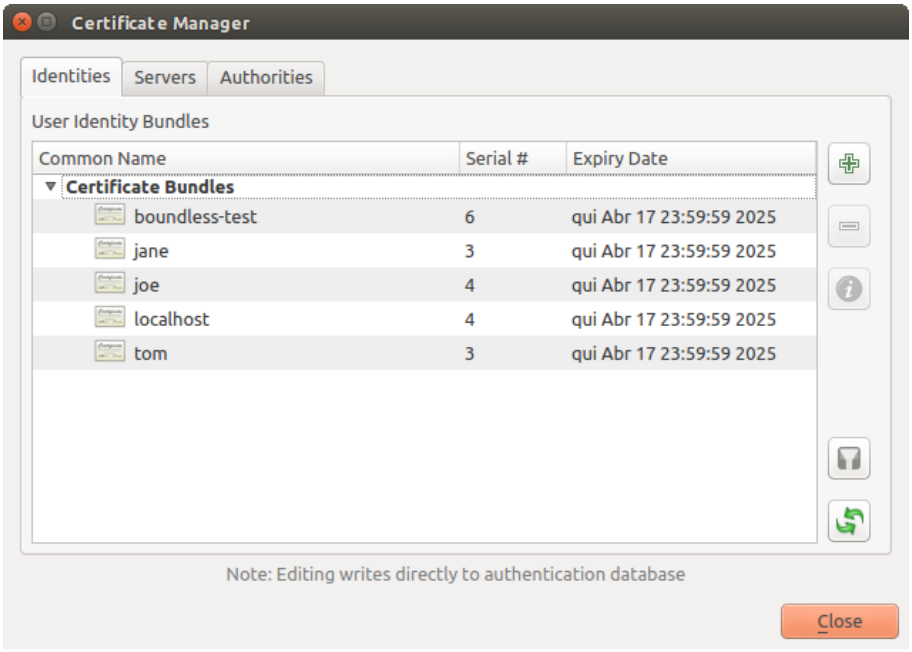


Figura21.28: Editor de identidades

Ao importar um pacote de identidade, ele pode ser protegido por senha ou desprotegido e pode conter certificados de CA formando uma cadeia de confiança. As certificações da cadeia de confiança não serão importadas aqui; eles podem ser adicionados separadamente na aba *Autoridades*.

Após a importação, o certificado e a chave privada do pacote serão armazenados no banco de dados, com o armazenamento da chave criptografado usando a senha mestra do QGIS. O uso subsequente do pacote armazenado do banco de dados exigirá apenas a entrada da senha mestra.

Os pacotes de identidade pessoal que consistem em componentes PEM/DER (.pem/.der) e PKCS#12 (.p12/.pfx) são suportados. Se uma chave ou pacote for protegido por senha, a senha será necessária para validar o componente antes da importação. Da mesma forma, se o certificado do cliente no pacote for inválido (por exemplo, sua data de vigência ainda não foi iniciada ou expirou), o pacote não poderá ser importado.

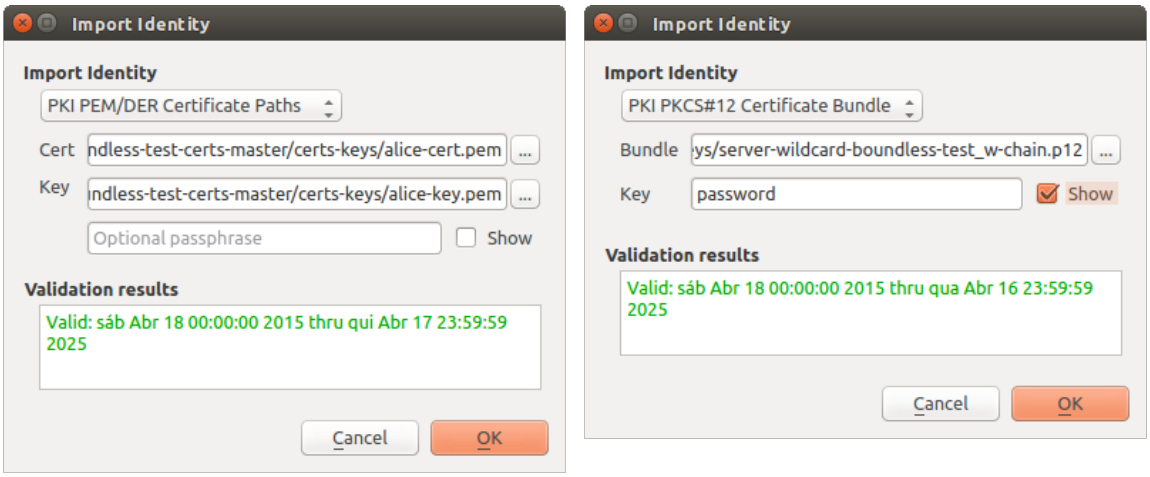


Figura21.29: PEM/DER and PKCS#12 identities import

21.2.4 Lidando com camadas ruins

Ocasionalmente, o ID de configuração de autenticação salvo com um arquivo de projeto não é mais válido, possivelmente porque o banco de dados de autenticação atual é diferente de quando o projeto foi salvo pela última vez ou devido a uma incompatibilidade de credenciais. Nesses casos, a caixa de diálogo *Manusear camadas ruins* será apresentada na inicialização do QGIS.

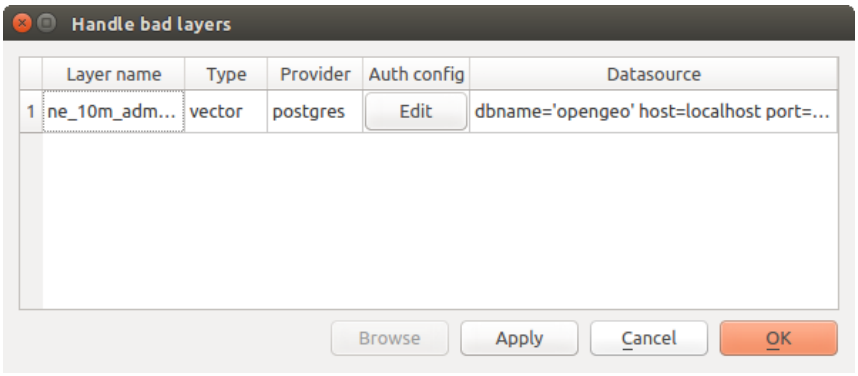


Figura21.30: Lidar com camadas ruins com autenticação

Se uma fonte de dados tiver um ID de configuração de autenticação associado a ela, você poderá editá-la. Fazer isso editará automaticamente a cadeia de origem da fonte de dados, da mesma forma que abrir o arquivo de projeto em um editor de texto e editar a cadeia de origem.

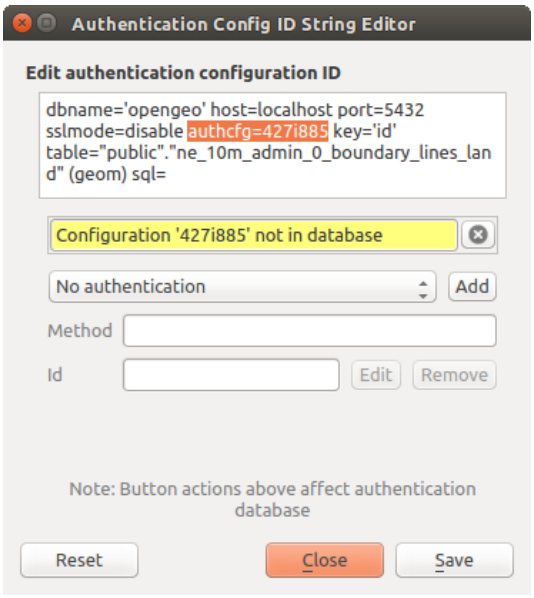


Figura21.31: Edite o ID de configuração de autenticação da camada inválida

21.2.5 Alterando o ID de configuração de autenticação

Ocasionalmente, você precisará alterar o ID de configuração de autenticação associado ao acesso a um recurso. Há casos em que isso é útil:

- **O ID de configuração de autenticação do recurso não é mais válido**: isso pode ocorrer quando você alterna bancos de dados de autenticação e precisa *alinhar* uma nova configuração para o ID já associado a um recurso.
- **Arquivos de projeto compartilhados**: se você pretendia compartilhar projetos entre usuários, por exemplo, por meio de um servidor de arquivos compartilhado, você pode *predefinir* um caractere de 7 caracteres (contendo **a-z** e/ou **0-9**) associado ao recurso. Em seguida, os usuários individuais alteram o ID de uma configuração de autenticação específica para suas credenciais do recurso. Quando o projeto é aberto, o ID é encontrado no banco de dados de autenticação, mas as credenciais são diferentes por usuário.

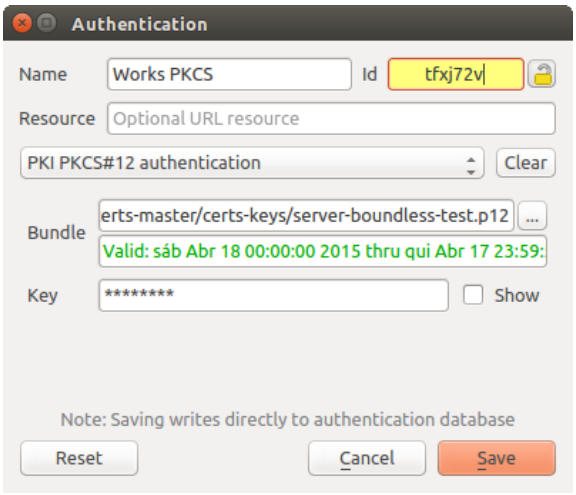


Figura21.32: Alterando o ID de configuração de autenticação de uma camada (campo de texto amarelo desbloqueado)

Aviso

Alterar o ID de configuração de autenticação é considerado uma operação avançada e só deve ser feito com total conhecimento do motivo pelo qual é necessário. É por isso que há um botão de bloqueio que precisa ser clicado para desbloquear o campo de texto do ID antes de editar o ID.

21.2.6 Suporte ao servidor QGIS

Ao usar um arquivo de projeto, com camadas que possuem configurações de autenticação, como base para um mapa no QGIS Servidor, há algumas etapas de configuração adicionais necessárias para o QGIS carregar os recursos:

- O banco de dados de autenticação precisa estar disponível
- A senha mestra do banco de dados de autenticação precisa estar disponível

Ao instanciar o sistema de autenticação, o Servidor irá criar ou usar o arquivo `qgis-auth.db` no ativo perfil de usuário, ou no diretório definido pela variável de ambiente `QGIS_AUTH_DB_DIR_PATH`. Pode ser que o usuário do Servidor não tenha um diretório HOME, nesse caso, use a variável de ambiente para definir um diretório que o usuário do Servidor tenha permissão de leitura/gravação e não esteja localizado dentro dos diretórios acessíveis pela web.

Para passar a senha mestra para o Servidor, escreva-a na primeira linha do arquivo em um caminho no sistema de arquivos legível pelo usuário de processos do Servidor e definido usando a variável de ambiente `QGIS_AUTH_SENHA_ARQUIVO`. Certifique-se de limitar o arquivo como legível apenas pelo usuário do processo do servidor e não armazene o arquivo em diretórios acessíveis pela web.

Nota

A variável `QGIS_AUTH_SENHA_ARQUIVO` será removida do ambiente do Servidor imediatamente após o acesso.

21.2.7 Exceções do servidor SSL

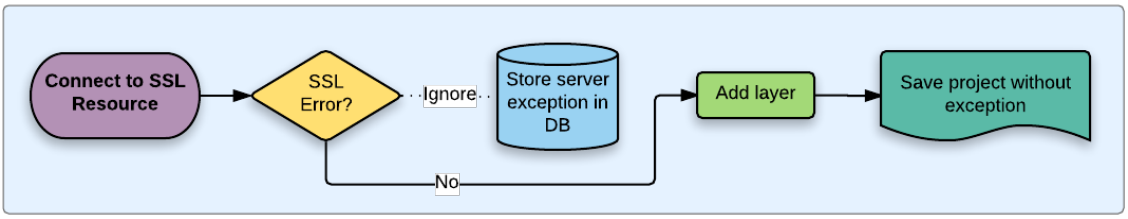


Figura21.33: Exceção do servidor SSL

Você pode gerenciar as configurações e exceções do servidor SSL na guia **Servidores** na seção **Autenticação** da caixa de diálogo **Opções** do QGIS

Às vezes, ao se conectar a um servidor SSL, ocorrem erros com o "aperto de mão" SSL ou com o certificado do servidor. Você pode ignorar esses erros ou criar uma configuração de servidor SSL como exceção. Isso é semelhante a como os navegadores da Web permitem que você substitua erros de SSL, mas com um controle mais granular.

Aviso

Você não deve criar uma configuração de servidor SSL a menos que tenha conhecimento completo de toda a configuração SSL entre o servidor e o cliente. Em vez disso, relate o problema ao administrador do servidor.

Nota

Algumas configurações de PKI usam uma cadeia confiável de CA completamente diferente para validar identidades de clientes da cadeia usada para validar o certificado do servidor SSL. Em tais circunstâncias, qualquer configuração criada para o servidor de conexão não corrigirá necessariamente um problema com a validação da identidade do cliente, e somente o emissor da identidade do cliente ou o administrador do servidor podem corrigir o problema.

Você pode pré-configurar uma configuração de servidor SSL clicando em |**Adicionar simbologia**| botão. Como alternativa, você pode adicionar uma configuração quando ocorrer um erro de SSL durante uma conexão e for apresentada uma caixa de diálogo **Erro SSL** (onde o erro pode ser ignorado temporariamente ou salvo no banco de dados e ignorado):

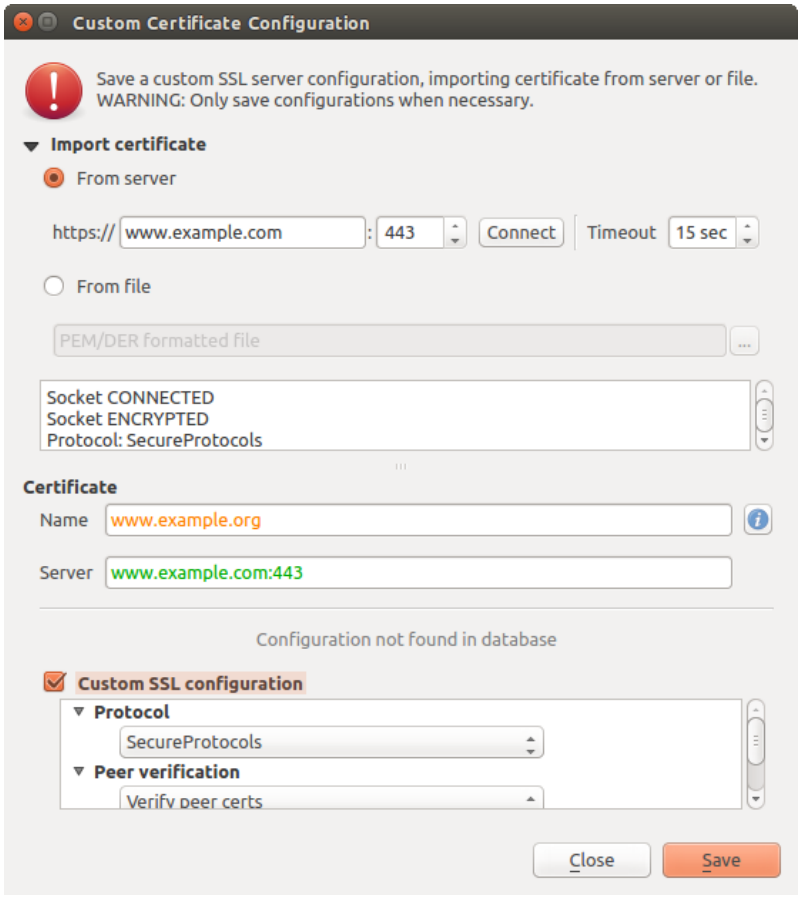


Figura21.34: Adicionando configuração manualmente

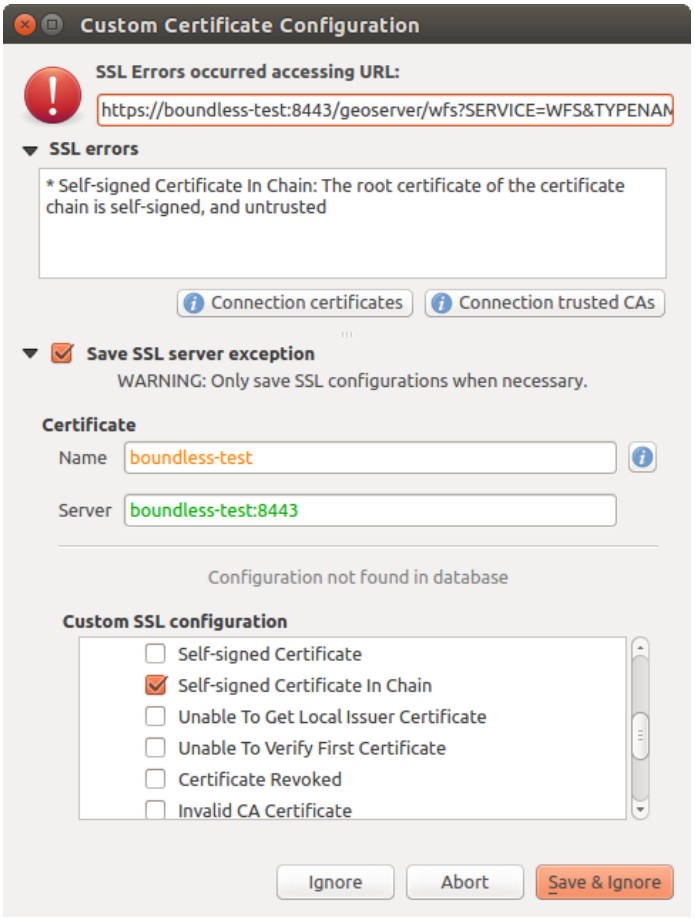


Figura21.35: Adicionando configuração durante erro de SSL

Depois que uma configuração SSL é salva no banco de dados, ela pode ser editada ou excluída.

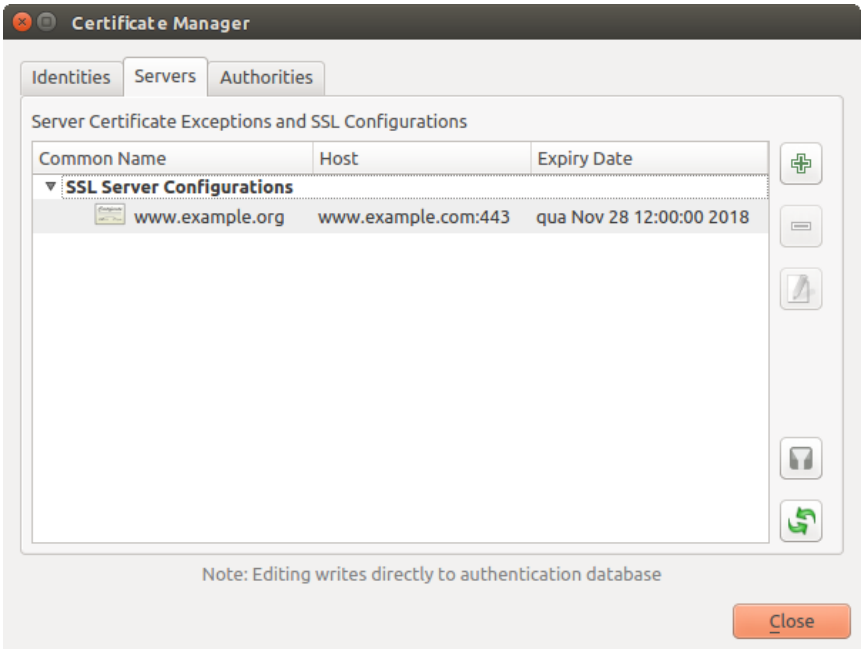


Figura21.36: Configuração SSL existente

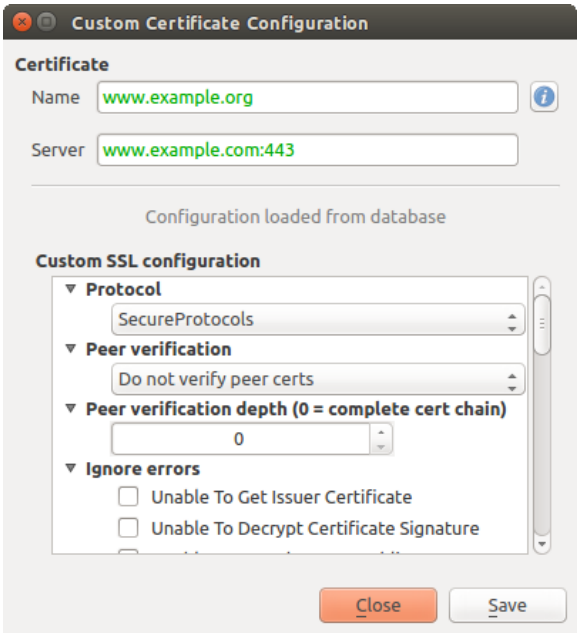


Figura21.37: Editando uma configuração SSL existente

Se você quiser pré-configurar uma configuração SSL e a caixa de diálogo de importação não estiver funcionando para a conexão do seu servidor, você pode acionar manualmente uma conexão através do **Console Python** executando o seguinte código (substituir `https://bugreports.qt-project.org` com a URL do seu servidor):

```
from qgis.PyQt.QtNetwork import QNetworkRequest
from qgis.PyQt.QtCore import QUrl
from qgis.core import QgsNetworkAccessManager

req = QNetworkRequest(QUrl('https://bugreports.qt-project.org'))
reply = QgsNetworkAccessManager.instance().get(req)
```

Isso abrirá uma caixa de diálogo de erro SSL se ocorrer algum erro, onde você pode optar por salvar a configuração no banco de dados.

CAPÍTULO 22

Integração com SIG GRASS

A integração do GRASS fornece acesso aos bancos de dados e funcionalidades do GRASS GIS (veja GRASS-PROJECT em *Referências Bibliográficas e Web*). A integração consiste em duas partes: provedor e complemento. O provedor permite navegar, gerenciar e visualizar camadas raster e vetor GRASS. O complemento pode ser usado para criar novos locations e mapasets GRASS, alterar a região GRASS, criar e editar camadas vetoriais e analisar dados GRASS 2-D e 3-D com mais de 400 módulos GRASS. Nesta seção, apresentaremos as funcionalidades do provedor e do complemento e forneceremos alguns exemplos de gerenciamento e trabalho com dados GRASS.

The provider supports GRASS version 6 and 7, the plugin supports GRASS 6 and 7 (starting from QGIS 2.12). QGIS distribution may contain provider/plugin for either GRASS 6 or GRASS 7 or for both versions at the same time (binaries have different file names). Only one version of the provider/plugin may be loaded on runtime however.

22.1 Demo dataset

As an example, we will use the QGIS Alaska dataset (see section *Baixando dados de amostra*). It includes a small sample GRASS `LOCATION` with three vector layers and one raster elevation map. Create a new folder called `grassdata`, download the QGIS 'Alaska' dataset `qgis_sample_data.zip` from https://qgis.org/downloads-list/#data and unzip the file into `grassdata`.

More sample GRASS `LOCATIONs` are available at the GRASS website.

22.2 Carregando camadas raster e vetorial GRASS

If the provider is loaded in QGIS, the location item with GRASS icon is added in the browser tree under each folder item which contains GRASS location. Go to the folder `grassdata` and expand location `alaska` and mapset `demo`.

Você pode carregar as camadas raster e vetorial GRASS como qualquer outra camada do navegador, clicando duas vezes no item da camada ou arrastando e soltando na tela ou legenda do mapa.

Dica

Carregando dados GRASS

If you don't see GRASS location item, verify in *Help* ► *About* ► *Providers* if GRASS vector provider is loaded.

22.3 Importing data into a GRASS LOCATION via drag and drop

This section gives an example of how to import raster and vector data into a GRASS mapset.

1. In QGIS browser navigate to the mapset you want to import data into.
2. In QGIS browser find a layer you want to import to GRASS, note that you can open another instance of the browser (*Browser Panel (2)*) if source data are too far from the mapset in the tree.
3. Drag a layer and drop it on the target mapset. The import may take some time for larger layers, you will see animated icon in front of new layer item until the import finishes.

When raster data are in different CRS, they can be reprojected using an *Approximate* (fast) or *Exact* (precise) transformation. If a link to the source raster is created (using `r.external`), the source data are in the same CRS and the format is known to GDAL, the source data CRS will be used. You can set these options in the *Browser* tab in *Opções do GRASS*.

If a source raster has more bands, a new GRASS map is created for each layer with **.<band number>** suffix and group of all maps with icon is created. External rasters have a different icon .

22.4 Managing GRASS data in QGIS Browser

- Copying maps: GRASS maps may be copied between mapsets within the same location using drag and drop.
- Deleting maps: Right click on a GRASS map and select *Delete* from context menu.
- Renaming maps: Right click on a GRASS map and select *Rename* from context menu.

22.5 Opções do GRASS

As opções do GRASS podem ser definidas na caixa de diálogo *Opções do GRASS*, que pode ser aberta clicando com o botão direito do mouse no local ou no item do conjunto de mapas no navegador e escolhendo :guilabel:`Opções do GRASS '.

22.6 Iniciando o complemento GRASS

To use GRASS functionalities in QGIS, you must select and load the GRASS plugin using the Plugin Manager. To do this, go to the menu *Plugins* ► *Manage and Install Plugins…*, select *GRASS* and click *OK*.

The following main features are provided with the GRASS menu (*Plugins* ► *GRASS*) when you start the GRASS plugin:

- Open Mapset
- New Mapset
- Close Mapset
- Open GRASS Tools
- Display Current GRASS Region

- GRASS Options

22.7 Abrindo o mapset GRASS

A GRASS mapset must be opened to get access to GRASS Tools in the plugin (the tools are disabled if no mapset is open). You can open a mapset from the browser: right click on mapset item and then choose *Open mapset* from context menu.

22.8 Localização e MAPSET GRASS

GRASS data are stored in a directory referred to as GISDBASE. This directory, often called `grassdata`, must be created before you start working with the GRASS plugin in QGIS. Within this directory, the GRASS GIS data are organized by projects stored in subdirectories called `LOCATIONs`. Each `LOCATION` is defined by its coordinate system, map projection and geographical boundaries. Each `LOCATION` can have several `MAPSETs` (subdirectories of the `LOCATION`) that are used to subdivide the project into different topics or sub-regions, or as workspaces for individual team members (see Neteler & Mitasova 2008 in *Referências Bibliográficas e Web*). In order to analyze vector and raster layers with GRASS modules, you generally have to import them into a GRASS `LOCATION`. (This is not strictly true – with the GRASS modules `r.external` and `v.external` you can create read-only links to external GDAL-supported datasets without importing them. This is not the usual way for beginners to work with GRASS, therefore this functionality will not be described here.)

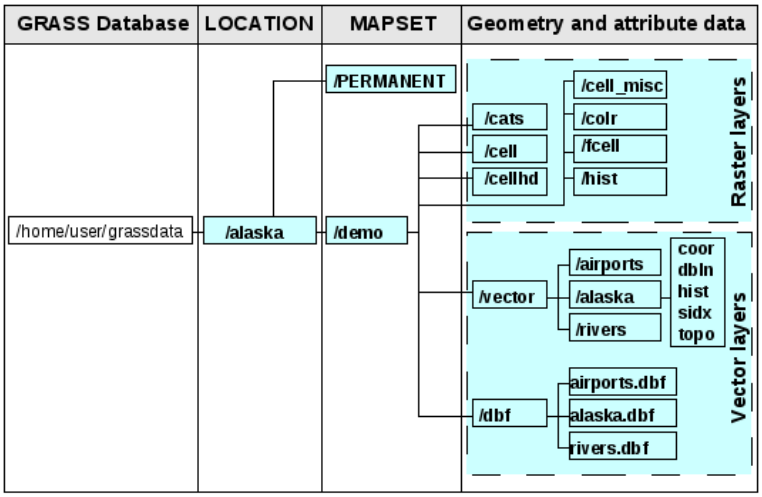


Figura22.1: Dados GRASS em LOCATION alaska

22.9 Importando dados para uma localização GRASS

See section *Importing data into a GRASS LOCATION via drag and drop* to find how data can be easily imported by dragging and dropping in the browser.

This section gives an example of how to import raster and vector data into the 'alaska' GRASS `LOCATION` provided by the QGIS 'Alaska' dataset in traditional way, using standard GRASS modules. Therefore, we use the landcover raster map `landcover.img` and the vector GML file `lakes.gml` from the QGIS 'Alaska' dataset (see *Baixando dados de amostra*).

1. Inicie o QGIS e verifique se o complemento GRASS está carregado.
2. In the GRASS toolbar, click the Open MAPSET icon to bring up the *MAPSET* wizard.

3. Select as GRASS database the folder `grassdata` in the QGIS Alaska dataset, as `LOCATION` 'alaska', as `MAPSET` 'demo' and click *OK*.
4. Now click the Open GRASS tools icon. The GRASS Toolbox (see section *Ferramentas GRASS*) dialog appears.
5. Para importar o mapa raster: arquivo: *landcover.img*, clique no módulo: arquivo:` r.in.gdal` no:guilabel:guia módulos em árvore. Este módulo GRASS permite importar arquivos raster GDAL que suporta GRASS: arquivo: 'local'. O módulo para: arquivo: *aparece r.in.gdal*.
6. Navegue até a pasta `raster` no conjunto de dados QGIS 'Alaska' e selecione o arquivo `landcover.img`.
7. As raster output name, define `landcover_grass` and click *Run*. In the *Output* tab, you see the currently running GRASS command `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass`.
8. When it says **Successfully finished**, click *View Output*. The `landcover_grass` raster layer is now imported into GRASS and will be visualized in the QGIS canvas.
9. Para importar o arquivo vetorial GML: arquivo *lakes.gml*, clique no módulo: arquivo:` v.in.ogr` no:guilabel:*módulos em árvores*. Este módulo GRASS permite importar arquivos vetoriais OGR apoiados na GRASS: arquivo: *local*. A caixa de diálogo para o arquivo *aparece v.in.ogr*.
10. Browse to the folder `gml` in the QGIS 'Alaska' dataset and select the file `lakes.gml` as OGR file.
11. As vector output name, define `lakes_grass` and click *Run*. You don't have to care about the other options in this example. In the *Output* tab you see the currently running GRASS command `v.in.ogr -o dsn=/path/to/lakes.gml output=lakes\_grass`.
12. When it says **Succesfully finished**, click *View Output*. The `lakes_grass` vector layer is now imported into GRASS and will be visualized in the QGIS canvas.

22.9.1 Criando uma nova Localização GRASS

As an example, here is the sample GRASS `LOCATION alaska`, which is projected in the Albers Equal Area projection using feet as units. This sample GRASS `LOCATION alaska` will be used for all examples and exercises in the following GRASS-related sections. It is useful to download and install the dataset on your computer (see *Baixando dados de amostra*).

1. Inicie o QGIS e verifique se o complemento GRASS está carregado.
2. Visualize o shapefile `alaska.shp` (veja a seção *Carregando uma camada de um arquivo*) do conjunto de dados do QGIS Alaska (veja *Baixando dados de amostra*).
3. In the GRASS toolbar, click on the New mapset icon to bring up the *MAPSET* wizard.
4. Select an existing GRASS database (GISDBASE) folder `grassdata`, or create one for the new `LOCATION` using a file manager on your computer. Then click *Next*.
5. We can use this wizard to create a new `MAPSET` within an existing `LOCATION` (see section *Adicionando um novo MAPSET*) or to create a new `LOCATION` altogether. Select *Create new location* (see Fig.22.2).
6. Digite um nome para `LOCATION` – usamos 'alaska' - e clique em *Próximo*.
7. Define the projection by clicking on the radio button *Projection* to enable the projection list.
8. We are using Albers Equal Area Alaska (feet) projection. Since we happen to know that it is represented by the EPSG ID 2964, we enter it in the search box. (Note: If you want to repeat this process for another `LOCATION` and projection and haven't memorized the EPSG ID, click on the CRS Status icon in the lower right-hand corner of the status bar (see section *Trabalhando com Projeções*)).
9. Na *Filtro*, insira 2964 para selecionar a projeção.
10. Haz clic en *Siguiente*.

11. To define the default region, we have to enter the LOCATION bounds in the north, south, east, and west directions. Here, we simply click on the button *Set Current QGIS Extent*, to apply the extent of the loaded layer `alaska.shp` as the GRASS default region extent.
12. Haz clic en *Siguiente*.
13. We also need to define a MAPSET within our new LOCATION (this is necessary when creating a new LOCATION). You can name it whatever you like - we used 'demo'. GRASS automatically creates a special MAPSET called PERMANENT, designed to store the core data for the project, its default spatial extent and coordinate system definitions (see Neteler & Mitasova 2008 in *Referências Bibliográficas e Web*).
14. Check out the summary to make sure it's correct and click *Finish*.
15. O novo: arquivo 'LOCALIZAÇÃO': 'Alaska', e dois: arquivos: *CONJUNTO DE MAPAS*, demonstração e" permanente ", são criados. O conjunto de trabalho aberto no momento é 'demonstração', como você definiu.
16. Note que algumas das ferramentas na barra de ferramentas GRASS que foram desativadas estão agora habilitadas.

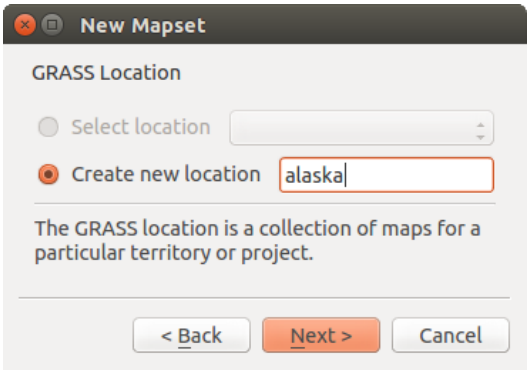


Figura22.2: Creating a new GRASS LOCATION or a new MAPSET in QGIS

If that seemed like a lot of steps, it's really not all that bad and a very quick way to create a LOCATION. The LOCATION 'alaska' is now ready for data import (see section *Importando dados para uma localização GRASS*). You can also use the already-existing vector and raster data in the sample GRASS LOCATION 'alaska', included in the QGIS 'Alaska' dataset *Baixando dados de amostra*, and move on to section *O modelo de dados vetorial do GRASS*.

22.9.2 Adicionando um novo MAPSET

A user has write access only to a GRASS MAPSET which he or she created. This means that besides access to your own MAPSET, you can read maps in other users' MAPSETs (and they can read yours), but you can modify or remove only the maps in your own MAPSET.

Todos: arquivos: 'CONJUNTO DE MAPAS' inclui um arquivo 'WIND' que armazena as coordenadas dos valores e a resolução selecionada (ver Neteler & Mitasova de 2008, em: ref:` literatura web` e seção:ref:*sec_regiao_grass*) .

1. Inicie o QGIS e verifique se o complemento GRASS está carregado.
2. In the GRASS toolbar, click on the icon to bring up the *MAPSET* wizard.
3. Selecione a pasta do banco de dados GRASS (GISDBASE): arquivo *grassdata* com o arquivo:` local` 'Alaska', onde queremos adicionar mais um: arquivo: *CONJUNTO DE MAPAS* chamado 'teste'.
4. Haz clic en *Siguiente*.
5. We can use this wizard to create a new MAPSET within an existing LOCATION or to create a new LOCATION altogether. Click on the radio button *Select location* (see Fig.22.2) and click *Next*.
6. Enter the name `test` for the new MAPSET. Below in the wizard, you see a list of existing MAPSETs and corresponding owners.

7. Click *Next*, check out the summary to make sure it's all correct and click *Finish*.

22.10 O modelo de dados vetorial do GRASS

It is important to understand the GRASS vector data model prior to digitizing. In general, GRASS uses a topological vector model. This means that areas are not represented as closed polygons, but by one or more boundaries. A boundary between two adjacent areas is digitized only once, and it is shared by both areas. Boundaries must be connected and closed without gaps. An area is identified (and labelled) by the **centroid** of the area.

Além de limites e centróides, um mapa vetor também pode conter pontos e linhas. Todos estes elementos de geometria podem ser misturados em um vetor e será representado em diferentes 'camadas' dentro de um GRASS vetor. Então, no GRASS, uma camada não é um vetor ou raster, mas um nível dentro de uma camada vetorial. Isso é importante para distinguir cuidadosamente. (Embora seja possível misturar elementos de geometria, é incomum e, mesmo no GRASS, utilizado somente em casos especiais, tais como rede de análise vetorial. Normalmente, você deve preferir armazenar diferentes elementos geométricos em diferentes camadas.)

É possível armazenar várias "camadas" em um conjunto de dados vetoriais. Por exemplo, campos, florestas e lagos pode ser armazenado em um vetor. Uma floresta adjacente e um lago podem compartilhar o mesmo limite, mas eles têm tabelas de atributos separados. Também é possível fixar atributos limites. Um exemplo pode ser o caso em que a fronteira entre um lago e uma floresta é um caminho, para que ele possa ter uma tabela de atributos diferentes.

A "camada" do recurso é definido pela "camada" dentro do GRASS. 'Camada' é o número que define, se houver mais do que uma camada interior do conjunto de dados (por exemplo, se a geometria for uma floresta ou lago). Por enquanto, ele pode ser apenas um número. No futuro, o GRASS também apoiará nomes como campos na interface do usuário.

Os atributos podem ser armazenados dentro do GRASS `LOCATION` como dBase, SQLite3 ou em tabelas externas de banco de dados, por exemplo, PostgreSQL, MySQL, Oracle, etc.

Atributos em tabelas de base de dados estão ligados a elementos geométricos usando um valor 'categoria'.

'Categoria' (chave, ID) é um número inteiro ligado a geometrias primitivas, e que é usada como ligação a uma coluna chave na tabela de base de dados.

> **Dica**
>
> **Aprendendo um Modelo Vetor GRASS**
>
> A melhor maneira de aprender o modelo vetorial GRASS e suas capacidades é baixar um dos muitos tutoriais GRASS onde o modelo vetorial é descrito mais profundamente. Veja https://grass.osgeo.org/learn/manuals/ para mais informações.

22.11 Criando uma nova camada vetorial GRASS

To create a new GRASS vector layer, select one of following items from mapset context menu in the browser:

- Nova Camada de Ponto
- Nova Camada de Linha
- Nova Camada de Polígono

and enter a name in the dialog. A new vector map will be created and layer will be added to canvas and editing started. Selecting type of the layer does not restrict geometry types which can be digitized in the vector map. In GRASS, it is possible to organize all sorts of geometry types (point, line and polygon) in one vector map. The type is only used to add the layer to the canvas, because QGIS requires a layer to have a specific type.

It is also possible to add layers to existing vector maps selecting one of the items described above from context menu of existing vector map.

In GRASS, it is possible to organize all sorts of geometry types (point, line and area) in one layer, because GRASS uses a topological vector model, so you don't need to select the geometry type when creating a new GRASS vector. This is different from shapefile creation with QGIS, because shapefiles use the Simple Feature vector model (see section *Criando novas camadas vetoriais*).

22.12 Digitalizando e editando uma camada vetorial GRASS

As camadas de vetor GRASS podem ser digitalizadas usando as ferramentas de digitalização QGIS padrão. No entanto, existem algumas particularidades que você deve conhecer devido a

- GRASS topological model versus QGIS simple feature
- complexidade do modelo GRASS
  - multiple layers in single maps
  - vários tipos de geometria em um único mapa
  - compartilhamento de geometria por várias feições de várias camadas

As particularidades são discutidas nas seções a seguir.

Save, discard changes, undo, redo

> **Aviso**
>
> Todas as alterações feitas durante a edição são imediatamente gravadas no mapa vetorial e nas tabelas de atributos relacionadas.

Changes are written after each operation, it is however, possible to do undo/redo or discard all changes when closing editing. If undo or discard changes is used, original state is rewritten in vector map and attribute tables.

Existem duas razões principais para este comportamento:

- It is the nature of GRASS vectors coming from conviction that user wants to do what he is doing and it is better to have data saved when the work is suddenly interrupted (for example, blackout)
- A necessidade de edição eficaz dos dados topológicos é uma informação visualizada sobre a correção topológica; essas informações só podem ser adquiridas no mapa vetorial GRASS se as alterações forem gravadas no mapa.

Barra de Ferramentas

The 'Digitizing Toolbar' has some specific tools when a GRASS layer is edited:

Tabela22.1: GRASS Digitizing Tools

Ícone	Ferramenta	Finalidade
	Novo Ponto	Digitaliza novo ponto
	Nova Linha	Digitaliza nova linha
	Nova Fronteira	Digitalizar novo limite
	Novo Centróide	Digitaliza novo centróide (etiqueta da área existente)
	New Closed Boundary	Digitalizar novo limite fechado

> **Dica**
>
> **Digitalizando polígonos no GRASS**
>
> If you want to create a polygon in GRASS, you first digitize the boundary of the polygon. Then you add a centroid (label point) into the closed boundary. The reason for this is that a topological vector model links the attribute information of a polygon always to the centroid and not to the boundary.

Categoria

Category, often called cat, is sort of ID. The name comes from times when GRASS vectors had only singly attribute "category". Category is used as a link between geometry and attributes. A single geometry may have multiple categories and thus represent multiple features in different layers. Currently it is possible to assign only one category per layer using QGIS editing tools. New features have automatically assigned new unique category, except boundaries. Boundaries usually only form areas and do not represent linear features, it is however possible to define attributes for a boundary later, for example in different layer.

Novas categorias são sempre criadas apenas na camada que está sendo editada.

It is not possible to assign more categories to geometry using QGIS editing, such data are properly represented as multiple features, and individual features, even from different layers, may be deleted.

Atributos

Attributes of currently edited layer can only be modified. If the vector map contains more layers, features of other layers will have all attributes set to '<not editable (layer #)>' to warn you that such attribute is not editable. The reason is, that other layers may have and usually have different set of fields while QGIS only supports one fixed set of fields per layer.

If a geometry primitive does not have a category assigned, a new unique category is automatically assigned and new record in attribute table is created when an attribute of that geometry is changed.

> **Dica**
>
> If you want to do bulk update of attributes in table, for example using 'Field Calculator' (*Utilizando a Calculadora de Campo*), and there are features without category which you don't want to update (typically boundaries), you can filter them out by setting 'Advanced Filter' to `cat is not null`.

Editing style

The topological symbology is essential for effective editing of topological data. When editing starts, a specialized 'GRASS Edit' renderer is set on the layer automatically and original renderer is restored when editing is closed. The style may be customized in layer properties 'Style' tab. The style can also be stored in project file or in separate file as any other style. If you customize the style, do not change its name, because it is used to reset the style when editing is started again.

> **Dica**
>
> Não salve o arquivo do projeto quando a camada for editada, a camada será armazenada com 'Editar estilo', o que não terá significado se a camada não for editada.

The style is based on topological information which is temporarily added to attribute table as field 'topo_symbol'. The field is automatically removed when editing is closed.

> **Dica**
>
> Não remova o campo 'topo_symbol' da tabela de atributos, isso tornaria os recursos invisíveis porque o renderizador é baseado nessa coluna.

Ajuste

To form an area, vertices of connected boundaries must have **exactly** the same coordinates. This can be achieved using snapping tool only if canvas and vector map have the same CRS. Otherwise, due conversion from map coordinates to canvas and back, the coordinate may become slightly different due to representation error and CRS transformations.

> **Dica**
>
> Use o SRC da camada também para a tela ao editar.

Limitações

Simultaneous editing of multiple layers within the same vector at the same time is not supported. This is mainly due to the impossibility of handling multiple undo stacks for a single data source.

On Linux and macOS only one GRASS layer can be edited at time. This is due to a bug in GRASS which does not allow to close database drivers in random order. This is being solved with GRASS developers.

> **Dica**
>
> **GRASS Edita Permissões**
>
> Você deve ser o proprietário do GRASS: arquivo: 'CONJUNTO DE MAPAS' que deseja editar. É impossível editar camadas de dados em: arquivo: 'CONJUNTO DE MAPAS' que não é seu, mesmo que você tenha permissão de gravação.

22.13 A ferramenta região GRASS

A região (definição de uma janela de trabalho espacial) no GRASS é importante para trabalhar com camadas. Análise Vetorial é, por padrão, não se limitando a quaisquer definições de região definida. Mas todos os vetores recém-criados terá a extensão espacial e a resolução da região GRASS atualmente definido, independentemente da sua extensão e da resolução original. A região GRASS atual é armazenado no: arquivo: '$ LOCAL / $ mapset / WIND', e define o norte, sul, leste e oeste como limites, o número de colunas e linhas, resolução espacial horizontal e vertical.

It is possible to switch on and off the visualization of the GRASS region in the QGIS canvas using the Display current GRASS region button.

The region can be modified in 'Region' tab in 'GRASS Tolls' dock widget. Type in the new region bounds and resolution, and click *Apply*. If you click on *Select the extent by dragging on canvas* you can select a new region interactively with your mouse on the QGIS canvas dragging a rectangle.

O módulo GRASS: arquivo: *g.região* fornece muito mais parâmetros para definir uma medida para a região e resolução adequados para sua análise. Você pode usar esses parâmetros com as Ferramentas GRASS, descrito na seção: ref: *subseção_grass_ferramentas*.

22.14 Ferramentas GRASS

The Open GRASS Tools box provides GRASS module functionalities to work with data inside a selected GRASS `LOCATION` and `MAPSET`. To use the GRASS Toolbox you need to open a `LOCATION` and `MAPSET` that you have write permission for (usually granted, if you created the `MAPSET`). This is necessary, because new raster or vector layers created during analysis need to be written to the currently selected `LOCATION` and `MAPSET`.

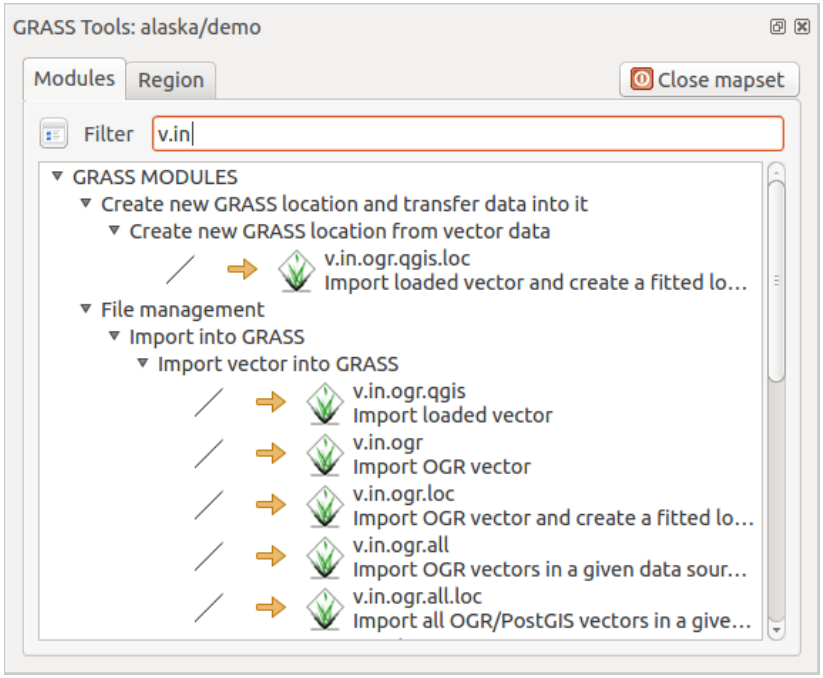


Figura22.3: GRASS Toolbox and Module Tree

22.14.1 Trabalhando com módulos GRASS

The GRASS plugin provides a user-friendly toolbox with about 200 of the available GRASS modules and functionalities.

Uma lista completa dos módulos GRASS disponíveis na Caixa de Ferramentas gráficas na versão QGIS 3.44 está disponível no wiki do GRASS em https://grasswiki.osgeo.org/wiki/GRASS-QGIS_relevant_module_list.

Também é possível personalizar o conteúdo Ferramentas GRASS. Este procedimento é descrito na seção:ref:*sec_customizacao_ferramentas*.

As shown in Fig.22.3, you can look for the appropriate GRASS module using the thematically grouped *Modules Tree* or the searchable *Modules List* tab.

Clicando em um ícone do módulo gráfico, uma nova aba será adicionada ao diálogo da caixa de ferramentas, oferecendo três novas sub-abas :guilabel:'Opções', :guilabel:'Saída' e :guilabel:'Manual'.

Opções

The *Options* tab provides a simplified module dialog where you can usually select a raster or vector layer visualized in the QGIS canvas and enter further module-specific parameters to run the module.

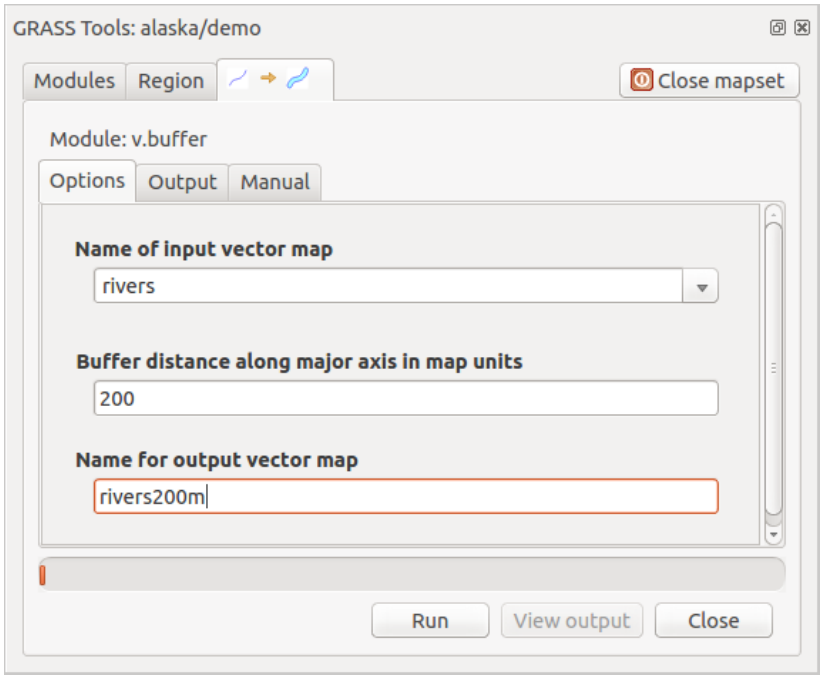


Figura22.4: Opções do módulo Caixa de Ferramentas GRASS

Os parâmetros do módulo fornecidos geralmente não estão completos para manter a caixa de diálogo simples. Se você deseja usar parâmetros e sinalizadores adicionais do módulo, é necessário iniciar o shell GRASS e executar o módulo na linha de comando.

Um novo recurso do QGIS 1.8 é o suporte ao botão *Mostrar Opções Avançadas* abaixo da caixa de diálogo do módulo simplificado na guia *Opções*. No momento, ele é adicionado apenas ao módulo `v.in.ascii` como exemplo de uso, mas provavelmente fará parte de mais ou de todos os módulos da Caixa de Ferramentas GRASS em versões futuras do QGIS. Isso permite que você use as opções completas do módulo GRASS sem a necessidade de alternar para o shell GRASS.

Saída

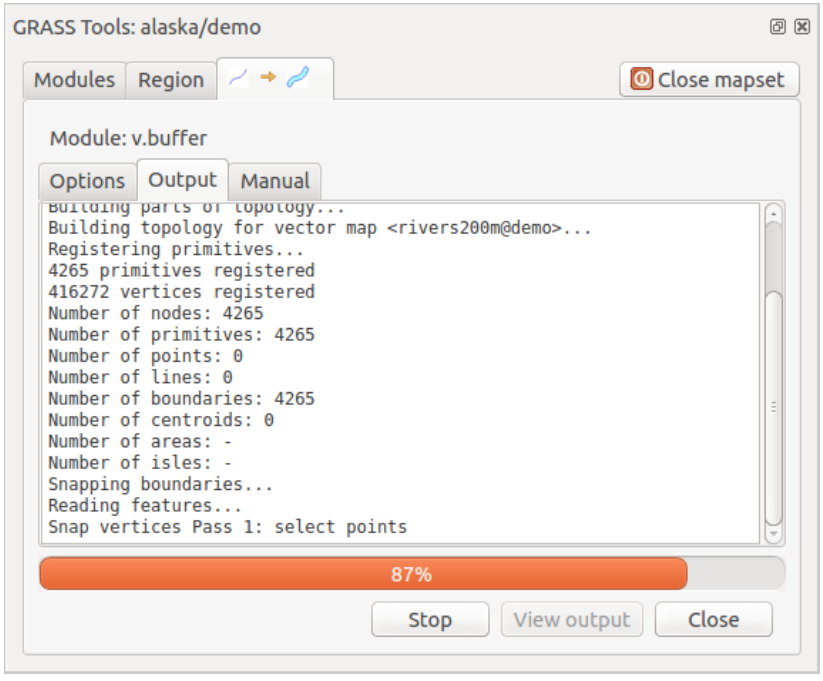


Figura22.5: GRASS Toolbox Module Output

The *Output* tab provides information about the output status of the module. When you click the *Run* button, the module switches to the *Output* tab and you see information about the analysis process. If all works well, you will finally see a `Successfully finished` message.

Manual

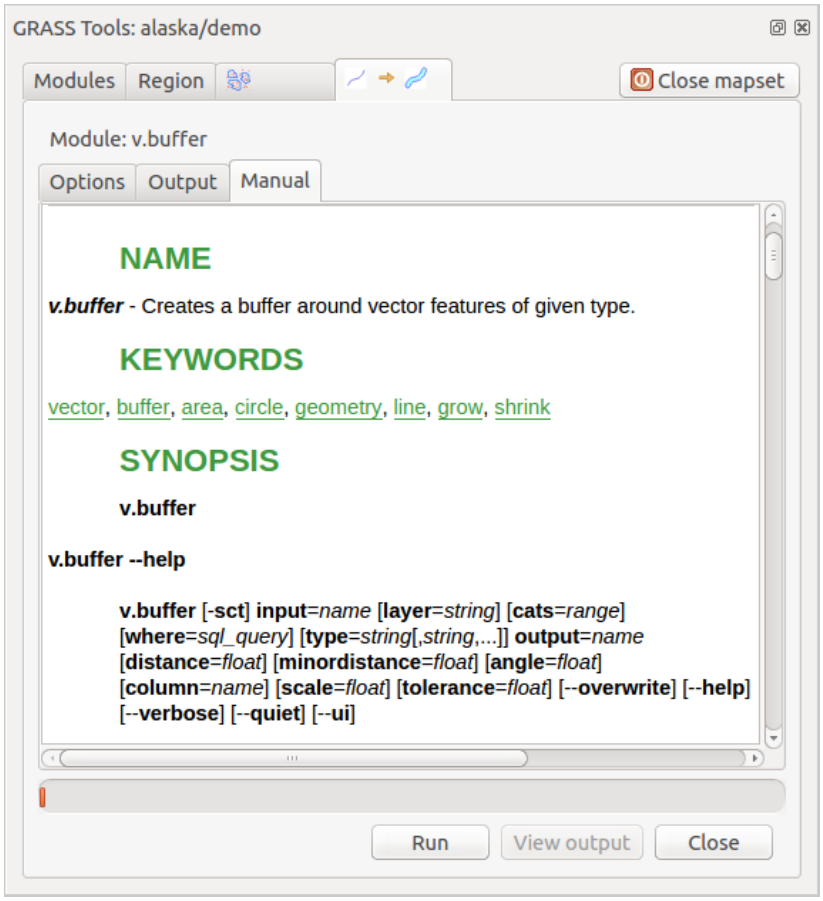


Figura22.6: GRASS Toolbox Module Manual

A:guilabel:'Manual' mostra a página de ajuda em HTML do módulo grass. Você pode usá-lo para verificar outros parâmetros do módulo e sinalizadores ou para obter um conhecimento mais profundo sobre o objetivo do módulo. No final de cada página do manual, você vê outras ligações a: arquivo: 'Ajuda', o arquivo: 'Temática' e do: arquivo: 'Completo'. Estas ligações fornecem a mesma informação que o módulo: arquivo: *g.manual*.

Dica

Mostra Resultados Imediatamente

Se você quiser exibir os resultados de cálculo imediatamente em sua tela do mapa, você pode usar o botão "Ver Saída" na parte inferior da aba módulo.

22.14.2 Exemplos de módulos GRASS

Os exemplos seguintes demonstrarão o poder de alguns dos módulos grama.

Criando linhas de contorno

O primeiro exemplo cria um mapa de contorno vetorial de elevação (DEM). Aqui, presume-se que você tem o Alasca: arquivo: 'local' configurado como explicado na seção: ref:` sec_import_loc_data

- Primeiro, abra o local clicando no botão Abrir mapset e escolher a localização do Alasca.
- Now open the Toolbox with the Open GRASS tools button.
- Na lista de categorias de ferramentas, clique duas vezes no: menu: 'Raster -> Gestão de Superfície -> Gerar linhas de contorno do vetor'.
- Now a single click on the tool **r.contour** will open the tool dialog as explained above (see *Trabalhando com módulos GRASS*).
- In the *Name of input raster map* enter `gtopo30`.
- Type into the *Increment between Contour levels* the value 100. (This will create contour lines at intervals of 100 meters.)
- Digite no *Nome para a saída do vetor mapa* o nome `ctour_100`.
- Click *Run* to start the process. Wait for several moments until the message `Successfully finished` appears in the output window. Then click *View Output* and *Close*.

Como se trata de uma grande região, vai demorar um pouco para exibir. Depois de terminar a renderização, você pode abrir a janela de propriedades da camada para alterar a cor da linha para que os contornos apareçam claramente sobre o raster elevação, como em:ref:*vector_properties_dialog*.

Em seguida, amplie uma pequena área montanhosa no centro do Alasca. Aproxima-se, você vai notar que os contornos têm cantos afiados. GRASS oferece a ferramenta **v.generalize** para alterar ligeiramente mapas vetoriais, mantendo sua forma original. A ferramenta usa diversos algoritmos diferentes, com finalidades diferentes. Alguns dos algoritmos (ie, Douglas Peuker e redução da Vertex) simplificam a linha removendo alguns dos vértices. O vetor resultante irá carregar mais rápido. Este processo é útil quando você tem um vetor altamente detalhado, mas você está criando um mapa muito pequena em escala, de modo que o detalhe é desnecessário.

> **Dica**
>
> **A ferramenta simplificar**
>
> Note que o QGIS possui uma *Vetor* ► *Ferramentas de Geometria* ► *Simplificar geometrias* que funciona exatamente como o algoritmo GRASS **v.generalize** Douglas-Peuker.

No entanto, o objetivo do exemplo é diferente. As linhas de contorno criado pelo "r.contour" têm ângulos agudos que devem ser suavizadas. Entre o algoritmo ** v.generalize ** , há Chaiken que faz exatamente isso (também ranhuras de Hermite). Esteja ciente de que estes algoritmos podem ** adicionar vértices** ao vetor, fazendo-a carregar ainda mais lentamente.

- Abra o GRASS Ferramentas e clique duas vezes na categoria: menu: 'Vetor -> Desenvolver mapa -> Geral', em seguida, clique no módulo **v.generalize** para abrir a janela de opções.
- Verifique se o vetor 'ctour_100' aparece como *Nome do vetor de entrada*.
- From the list of algorithms, choose Chaiken's. Leave all other options at their default, and scroll down to the last row to enter in the field *Name for output vector map* 'ctour_100_smooth', and click *Run*.
- The process takes several moments. Once `Successfully finished` appears in the output windows, click *View Output* and then *Close*.

- É possível mudar a cor do vetor para apresentar claramente o fundo matricial e para contrastar com as linhas de contorno originais. Você notará que as novas linhas de contorno têm cantos mais suaves do que o original, enquanto permanecer fiel à forma geral de origem.

Figura22.7: Módulo GRASS v.generalize para suavizar um mapa vetorial

Dica

Outros usos para r.contour

O processo descrito acima pode ser utilizado em outras situações equivalentes. Se você tem um mapa de dados de precipitação, por exemplo, então o mesmo método será utilizado para criar um mapa de vetor de linhas (chuvas constantes).

Criando um efeito de sombreamento 3D

Vários métodos são usados para exibir as camadas de elevação e dar um efeito 3-D para mapas. O uso de curvas de nível, como mostrado acima, é um método popular, muitas vezes escolhido para produzir mapas topográficos. Outra forma de apresentar um efeito 3-D é de sombreamento. O efeito Sombra é criado a partir de um DEM (elevação) que calcula a inclinação e o aspecto de cada célula, em seguida, simula a posição do sol no céu dando um valor de reflexo para cada célula. Assim, você tem pistas virada para o sol; as encostas viradas longe do sol (na sombra) são escurecidas.

- Comece este exemplo carregando o 'raster de elevação gtopo30'. Inicie o GRASS Ferramentas, e sob a categoria Raster, clique duas vezes para abrir: menu: 'Análise espacial -> Análise do Terreno'.
- Então, clique **r.shaded.relief** para abrir o módulo.
- Change the *azimuth angle* 1,00 270 to 315.
- Enter `gtopo30_shade` for the new hillshade raster, and click *Run*.
- Quando o processo termina, adicione o raster sombreado ao mapa. Você deve vê-lo exibido em escala de cinza.
- Para visualizar tanto o sombreamento da colina e as cores do "gtopo30", mova o mapa sombreado abaixo "mapa gtopo30" na tabela de conteúdo, em seguida, abra o: menu: 'Propriedades de' `"GTOPO30", mude para o: guilabel: 'transparência': e defina seu nível de transparência a cerca de 25%.

Agora você deve ter a elevação `gtopo30` com o seu mapa de cores e configuração de transparência apresentada **acima** o tons de cinza mapa sombreado. Para ver os efeitos visuais do sombreamento, desligue o mapa `gtopo30_shade` em seguida, ligue-o novamente.

Estatísticas Raster em um mapa vetor

O próximo exemplo mostra como um módulo GRASS pode agregar dados raster e adicionar colunas de estatísticas para cada polígono em um mapa vetor.

- Again using the Alaska data, refer to *Importando dados para uma localização GRASS* to import the `shapefiles/trees.shp` file into GRASS.
- Agora, um passo intermediário é necessária: centroides deve ser adicionado ao mapa importando as árvores para torná-lo num vetor de área de grass completo (incluindo os limites e centróide).
- Na caixa de ferramentas, escolha: menu: `Vetor -> Gerenciar recursos, e abra o módulo ** v.centroids **.
- Digite como *mapa vetor de saída* 'forest_areas' e rode o módulo.
- Now load the `forest_areas` vector and display the types of forests - deciduous, evergreen, mixed - in different colors: In the layer *Properties* window, *Symbology* tab, choose from *Legend type* 'Unique value' and set the *Classification field* to 'VEGDESC'. (Refer to the explanation of the symbology tab in *Propriedades da simbologia* of the vector section.)
- Em seguida, reabrir o GRASS Ferramentas no: menu: 'Vetor -> Atualização do Vetor' para outros mapas.
- Clique no módulo **v.rast.stats**. Digite "gtopo30" e "forest_areas".
- Only one additional parameter is needed: Enter *column prefix* `elev`, and click *Run*. This is a computationally heavy operation, which will run for a long time (probably up to two hours).
- Finalmente, abra o *"areas_floresta" na tabela de atributos, e verifique que várias novas colunas foram adicionadas, incluindo* `elev_min``, `` elev_max``, `` elev_média``, etc, para cada polígono florestal.

22.14.3 Personalizando a caixa de ferramentas GRASS

Quase todos os módulos de GRASS podem ser adicionados a caixa de ferramentas. Uma interface XML é fornecido para analisar os arquivos mais simples que configuram a aparência e os parâmetros dos módulos dentro da caixa de ferramentas.

Um exemplo de arquivo XML para gerar o módulo `v.buffer` (v.buffer.qgm) se parece com isto:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE qgisgrassmodule SYSTEM "http://mrcc.com/qgisgrassmodule.dtd">

<qgisgrassmodule label="Vector buffer" module="v.buffer">
        <option key="input" typeoption="type" layeroption="layer" />
        <option key="buffer"/>
        <option key="output" />
</qgisgrassmodule>
```

The parser reads this definition and creates a new tab inside the Toolbox when you select the module.

CAPÍTULO 23

QGIS estrutura de processamento

23.1 Introducción

This chapter introduces the QGIS processing framework, a geoprocessing environment that can be used to call native and third-party algorithms from QGIS, making your spatial analysis tasks more productive and easy to accomplish.

As a *Core plugin*, Processing is installed by default but you need to activate it:

1. Go to *Plugins* ► *Manage and install plugins…*
2. Click on the *Installed* tab at the left
3. Check the box next to the *Processing* entry
4. Cierra el cuadro de diálogo.

   A *Processing* menu is now available in the top menu bar. From there you can reach the main components of this framework.

Nas seções seguintes, vamos analisar como usar os elementos gráficos deste quadro e tirar o máximo proveito de cada um deles.

There are four basic elements in the framework GUI, which are used to run algorithms for different purposes. Choosing one tool or another will depend on the kind of analysis that is to be performed and the particular characteristics of each user and project. All of them (except for the batch processing interface, which is called from the toolbox or the algorithm execution dialog, as we will see) can be accessed from the *Processing* menu item (you will see more entries; the remaining ones are not used to execute algorithms and will be explained later in this chapter).

- The *Toolbox*: The main element of the GUI, it is used to execute a single algorithm or run a batch process based on that algorithm.

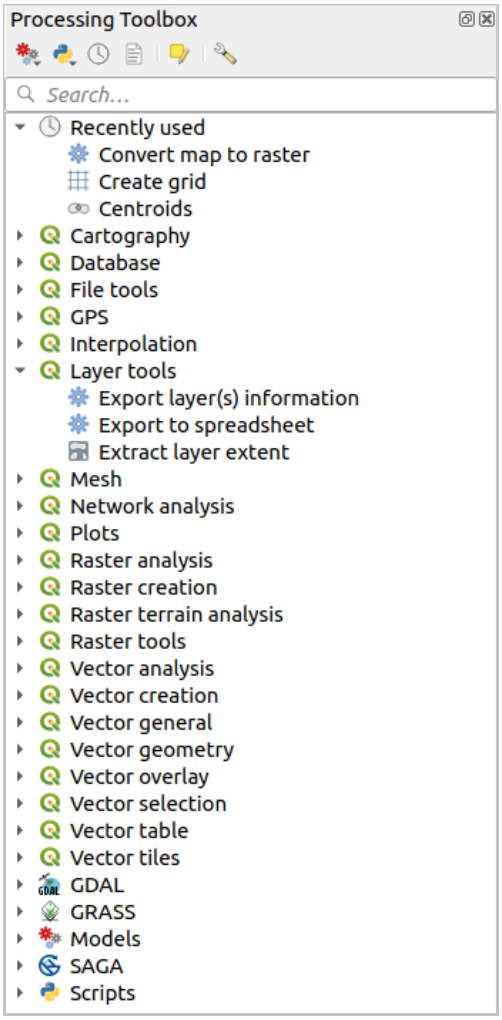


Figura23.1: Processamento

- The *Model Designer*: Several algorithms can be combined graphically using the modeler to define a workflow, creating a single process that involves several subprocesses.

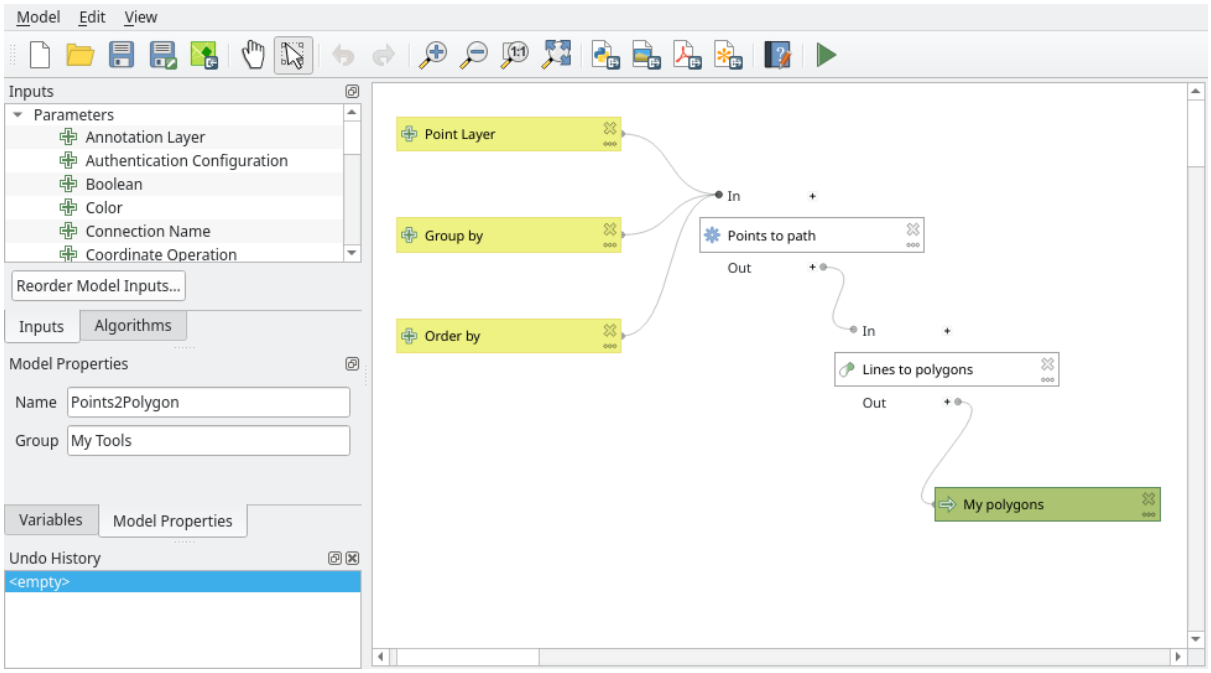


Figura23.2: Modelador de processamento

- The *History* manager: All actions performed using any of the aforementioned elements are stored in a history file and can be later easily reproduced using the history manager.

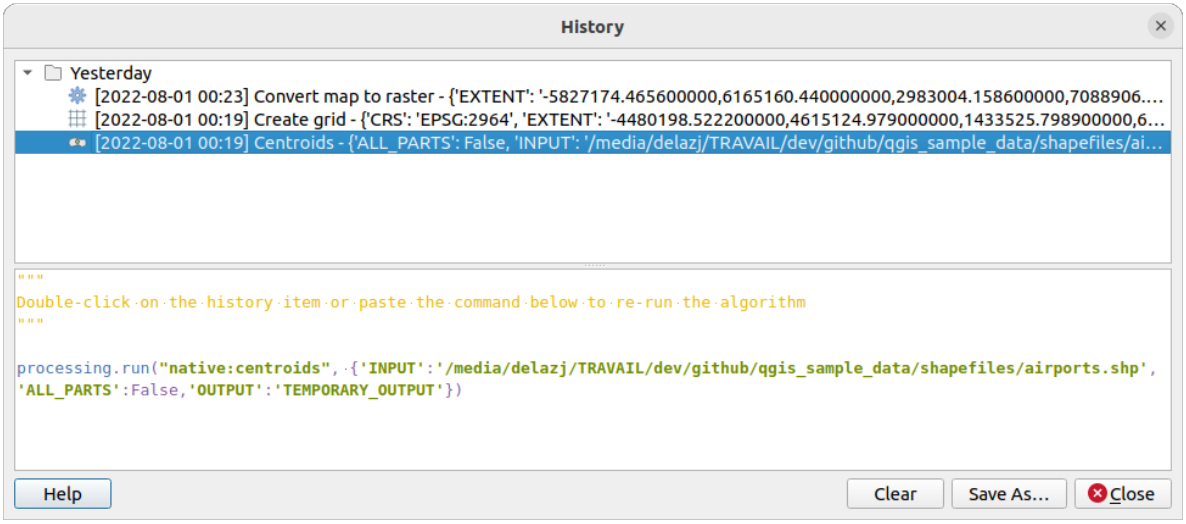


Figura23.3: Histórico de Processamento

- The *Batch Processing* interface: This interface allows you to execute batch processes and automate the execution of a single algorithm on multiple datasets.

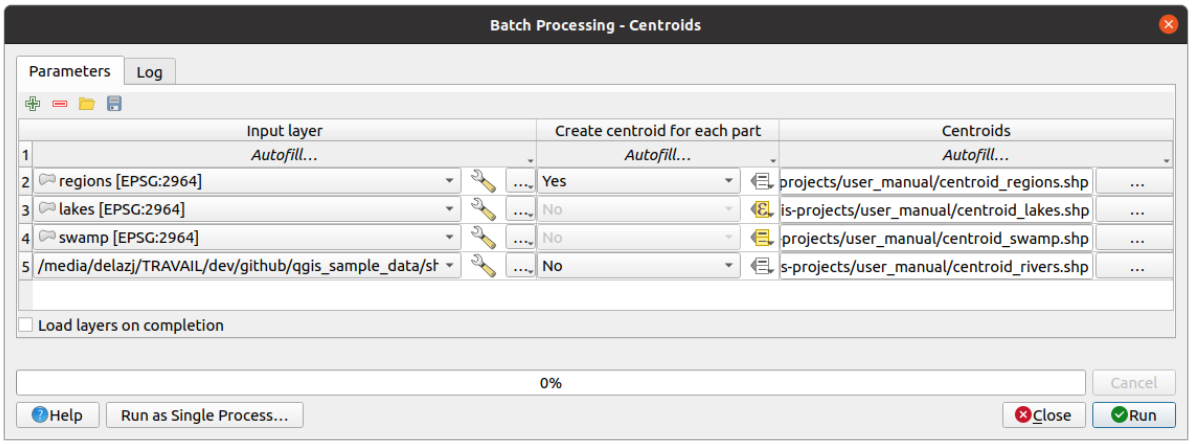


Figura23.4: Batch Processing interface

Nas seções seguintes, vamos analisar cada um desses elementos em detalhe.

23.2 Configuring the Processing Framework

The Processing Options menu (*Settings*► *Options* ► *Processing* tab) allows you to configure how algorithms work. Configuration parameters are structured in separate blocks that you can select on the left-hand side of the dialog.

23.2.1 General

The *General* block contains the default settings to control how the algorithm dialog as well as input or output parameters should behave. Some of the settings can however be overridden at the algorithm level, per *algorithm run* or for *individual parameters*.

- *Default output raster layer extension* is by default `tif`
- *Default output vector layer extension* is by default `gpkg`
- *Invalid features filtering* when executing algorithm:
  - *Do not filter (better performance)*: all the features (with valid and invalid geometries) are processed, but the result may be erroneous depending on how the geometry invalidity affects the operations
  - *Skip (ignore) features with invalid geometries*, meaning that only a subset of your dataset (the valid geometry features) will be processed
  - *Stop algorithm execution when a geometry is invalid*: you'll need to track and fix the invalid geometries if you want the algorithm to process the whole layer. Algorithms like *Verificar a validade* or *Corrigir geometrias* can help you achieve this.

  The *Invalid features filtering* setting can be overridden on a per-input basis, at algorithm runtime.
- *Keep dialog open after running algorithm*. Once an algorithm has finished execution and its output layers are loaded into the QGIS project, the algorithm dialog is closed. If you want to keep it open (to run the algorithm again with different parameters, or to better check the output that is written to the log tab), check this option.
- *Max Threads*
- *Output folder* for non temporary outputs: If no folder path is provided for the Processing execution outputs, this is the folder in which they will be saved. Default is `processing` under the user home directory (e.g., `C:\Users\<YOURUSERNAME>\processing\` on Windows).

- *Override temporary output folder path*: Temporary outputs are saved by default in the `tmp` folder on the machine. This option helps you set a different place for storage.
- *Pre-execution script* and *Post-execution script*. These parameters point to files that contain scripts written using the processing scripting functionality, explained in the section covering scripting and the console.
- *Prefer output filename for layer names*. The name of each resulting layer created by an algorithm is defined by the algorithm itself. In some cases, a fixed name might be used, meaning that the same output name will be used, no matter which input layer is used. In other cases, the name might depend on the name of the input layer or some of the parameters used to run the algorithm. If this checkbox is checked, the name will be taken from the output filename instead. Notice that, if the output is saved to a temporary file, the filename of this temporary file is usually a long and meaningless one intended to avoid collision with other already existing filenames.
- *Results group name*. If you want to obtain all processing result layers in a group in the *Layers* panel, set a group name for this parameter. The group may exist already or not. QGIS will add all output layers to such a group. By default, this parameter is empty, so all output layers are added to different places in the *Layers* panel, depending on the item that is active when running an algorithm. Note that output layers will be loaded to the *Layers* panel only if *Open output file after running algorithm* is checked in the algorithm dialog.
- *Show algorithms with known issues*: By default, QGIS avoids display of broken algorithms (generally from third-party providers). If checked, they will be available in the Processing toolbox, with a warning icon and a tooltip explaining they have issues. Use at your own risks.
- *Show feature count for output vector layers*. Since calculating the feature count might take some time in certain data formats, this option is off by default.
- *Show layer CRS definition in selection boxes*
- *Show tooltip when there are disabled providers*
- *Style for line layers*, *Style for point layers*, *Style for polygons layers* and *Style for raster layers* are used for setting the default rendering style for output layers (that is, layers generated by processing algorithms). Just create the style you want using QGIS, save it to a file, and then enter the path to that file in the settings so the algorithms can use it. Whenever a layer is loaded by Processing and added to the QGIS canvas, it will be rendered with that style.

  Rendering styles can be configured individually for each algorithm and each one of its outputs. Just right-click on the name of the algorithm in the toolbox and select *Edit rendering styles for outputs*. You will see a dialog like the one shown next.

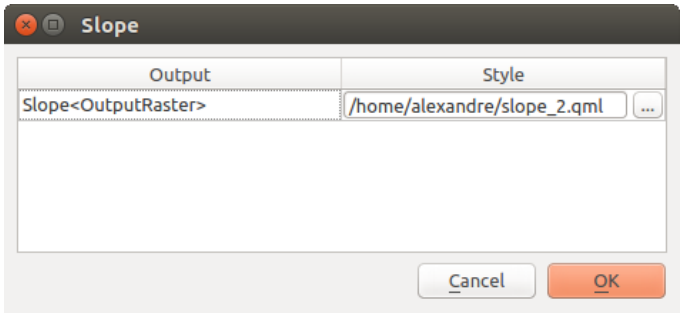


Figura23.5: Rendering Styles

  Select the style file (`.qml`) that you want for each output and press *OK*.
- *Warn before executing if parameter CRS's do not match*: By default, QGIS native algorithms (i.e. the ones listed under the *Menus* ► *QGIS (native C++)* group) transparently reproject the input layers to the first one's CRS before execution. Check this option to get a notification from the other tools that do not suppport reprojection, when the inputs CRS are not identical. Third-party providers are not concerned.

23.2.2 Menus

The *Menus* block controls whether an algorithm, script or model (built-in or provided by plugins) should be made available through a dedicated menu or toolbar (along with the Processing Toolbox). For each item of each provider, you can:

- *Add button in toolbar*, making it available in the *Processing Algorithms* toolbar
- assign an *Icon* to the algorithm
- set a *Menu path*: the algorithm will then be available through an existing or a custom menu, e.g. `Vect&or/MyTopAlgorithms` will add a shortcut in a *MyTopAlgorithms* sub-menu of the default *Vector* menu.

Reinicie o QGIS para aplicar as configurações. A qualquer momento, suas alterações podem ser *Resetadas para as configurações padrão*.

23.2.3 Models and Scripts

In the *Models* and *Scripts* blocks, a default path, set under the active *User profile* folder, is provided for storing models and scripts respectively. As for the other options, you can modify the path, e.g., to lead to a remote or shared folder for all your users.

23.2.4 Providers

You will also find a block for algorithm *Providers*. This is the place where installed providers expose their settings.

By default, QGIS is installed with two algorithm providers:

- *GDAL* whose algorithms you may *Activate* (or not) in this dialog. Read more about the *GDAL algorithms*.
- *GRASS* whose algorithms require the installation of the *GRASS GIS Processing Provider* from the *Plugins Manager*. You could then set some dedicated settings such as:
  - *For raster layers, use r.external (faster) instead of r.in.gdal*
  - *For vector layers, use v.external (faster) instead of v.in.ogr*
  - *Location of GRASS docs*: by default, and depending on your OS, QGIS utilizes a hardcoded list of paths for finding the local GRASS documentation and opening it whenever you hit the *Help* button of a GRASS algorithm dialog. Filling this option will override the default path, allowing you to use a custom location, or remote documentation.
  - *Log console output*
  - *Log execution commands*

Installed plugins that provide Processing algorithms may also have their provider listed in this group, with custom settings.

23.3 A Barra de Ferramentas

The *Processing Toolbox* is the main element of the processing GUI, and the one that you are more likely to use in your daily work. It shows the list of all available **algorithms** grouped in different blocks called *Providers*, and custom **models** and **scripts** you can add to extend the set of tools. Hence the toolbox is the access point to run them, whether as a single process or as a batch process involving several executions of the same algorithm on different sets of inputs.

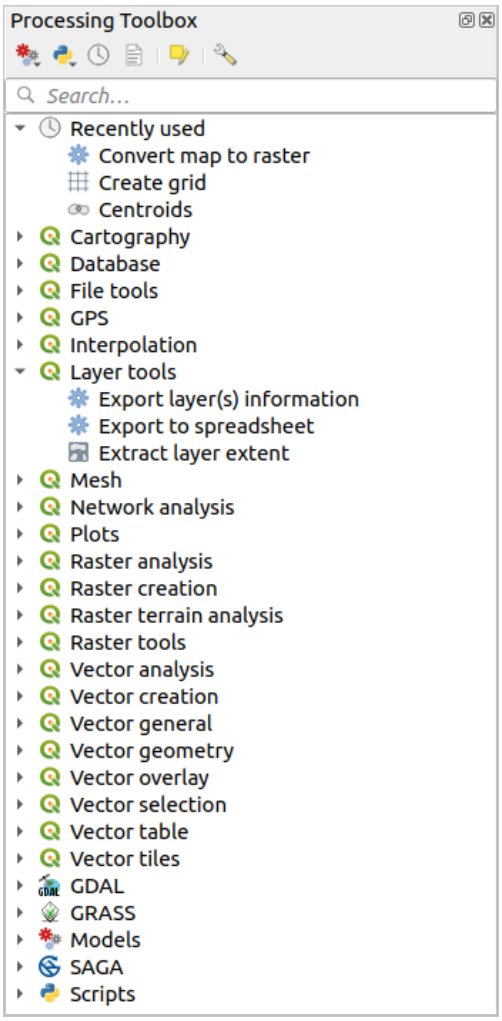


Figura23.6: Processamento

Providers can be (de)activated in the *Processing settings dialog*. By default, only providers that do not rely on third-party applications (that is, those that only require QGIS elements to be run) are active. Algorithms requiring external applications might need additional configuration. Configuring providers is explained in a *later chapter* in this manual.

In the upper part of the toolbox dialog, you will find a set of tools to:

- work with Models: *Create New Model…*, *Open Existing Model…* and *Add Model to Toolbox…*;
- work with Scripts: *Create New Script…*, *Create New Script from Template…*, *Open Existing Script…* and *Add Script to Toolbox…*;
- open the History panel;
- open the Results Viewer panel;

- toggle the toolbox to the *in-place modification mode* using the Edit Features In-Place button: only the algorithms that are suitable to be executed on the active layer without outputting a new layer are displayed;
- open the Options dialog.

Below this toolbar is a *Search…* box to help you easily find the tools you need. You can enter any word or phrase on the text box. Notice that, as you type, the number of algorithms, models or scripts in the toolbox is reduced to just those that contain the text you have entered in their names or keywords.

The top of the algorithm list shows the most Recently Used tools and the Favorites tools, making it convenient to reexecute them. The Recently Used list is there by default, showing the algorithms you have recently executed. The Favorites list is created when you manually add an algorithm to it.

You can right-click on any algorithm in the toolbox and choose from the contextual menu to:

- *Execute…* to run the algorithm.
- *Execute as Batch Process…* to run the algorithm in batch mode.
- *Edit Rendering Styles for Outputs…* to customize the rendering styles for the algorithm's outputs.
- *Add to Favorites* to add the algorithm to the Favorites list.
- *Remove from Favorites* to remove the algorithm from the Favorites list.

Unlike the Recently Used list, the Favorites list remains static and is not affected by which algorithms were executed.

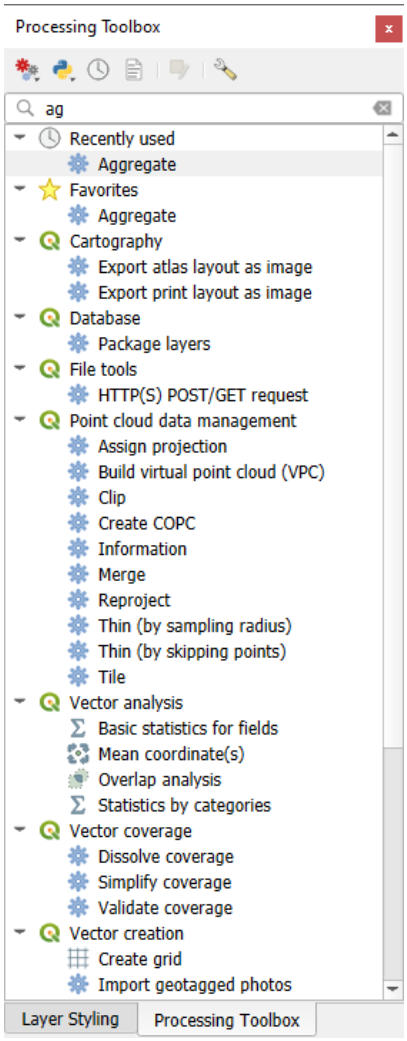


Figura23.7: Caixa de ferramentas de processamento mostrando os resultados da pesquisa

Para executar uma ferramenta, basta clicar duas vezes sobre seu nome na caixa de ferramentas.

23.3.1 O diálogo do algoritmo

Once you double-click on the name of the algorithm that you want to execute, a dialog similar to that in the Fig.23.8 below is shown (in this case, the dialog corresponds to the `Centroids` algorithm). Dialog title will include the group name from which the algorithm originates.

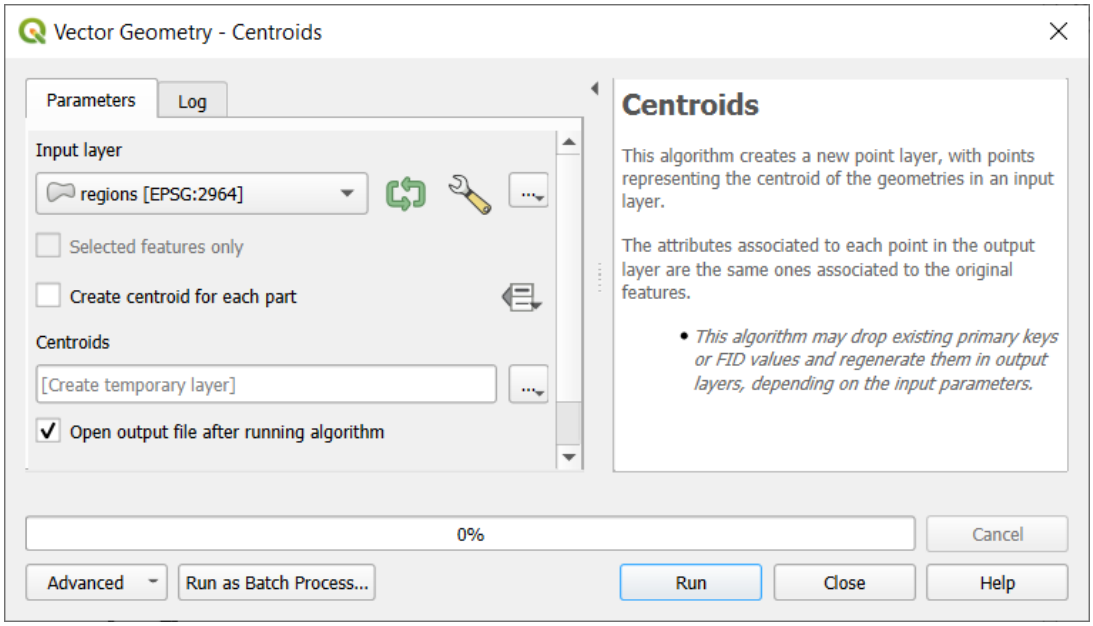


Figura23.8: Diálogo Algoritmo - Parâmetros

The dialog shows two tabs (*Parameters* and *Log*) on the left part, the algorithm description on the right, and a set of buttons at the bottom.

Parameter types

The *Parameters* tab is used to set the input values that the algorithm needs to be executed. It shows a list of input values and configuration parameters to be set. It of course has a different content, depending on the requirements of the algorithm to be executed, and is created automatically based on those requirements.

> **Dica**
>
> **Setting your own default values for algorithm parameters**
>
> Algorithm dialogs open with some parameters prefilled with values from QGIS installation. It is however possible to set *your own default values* for specific algorithm parameters so that they are used at algorithm startup.

Embora o número e tipo de parâmetros dependerem das características do algoritmo, a estrutura é semelhante para todos eles. Os parâmetros encontrados na tabela podem ser de um dos seguintes tipos.

- A **vector layer**, to select from a list of all vector layers available (currently opened) in QGIS. You can also use unloaded layers: press the … button on the widget right-hand side, and select:
  - *Select file…*: selects file on disk using the Operating System file explorer
  - *Browse for layer…*: opens the *Browser panel*, allowing to take the layers directly from database sources (PostgreSQL, SQL Server, Oracle, …), web services (WFS, AFS, …) or files on disk.

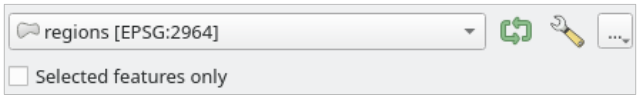


Figura23.9: Vector input widget

Dica

Drag-and drop as convenient alternative for layer selection

Besides selecting from the drop-down menu or browsing sources, you can also drag and drop a layer from the *Layers* or *Browser* panel, or from the Operating System file explorer directly into the combobox.

Nota

By default, the layer widget shows the CRS of the layer along with its name. If you do not want to see this additional information, you can disable this functionality in the Processing Settings dialog, unchecking the *General* ► *Show layer CRS definition in selection boxes* option.

The vector input widget also has following features:

- an iterator button: If toggled, the algorithm will be executed iteratively on each one of its features, instead of just once for the whole layer, producing as many outputs as times the algorithm is executed. This allows for automating the process when all features in a layer have to be processed separately. If the algorithm contains several input vectors you can iterate over, the iteration will be processed only on the first toggled parameter, in the order parameters are declared in the algorithm.
- Advanced options button to adjust settings to use for that specific parameter. These settings concern:
  - *Invalid feature filtering*: allows the *default method* for handling features with invalid geometries to be overridden
  - *Limit features processed*: optional limit on number of features processed from the source
  - *Feature filter*: allows to enter an expression to subset the layer dynamically when running the tool, avoiding the need for separate steps to set layer filters or create layer subsets.

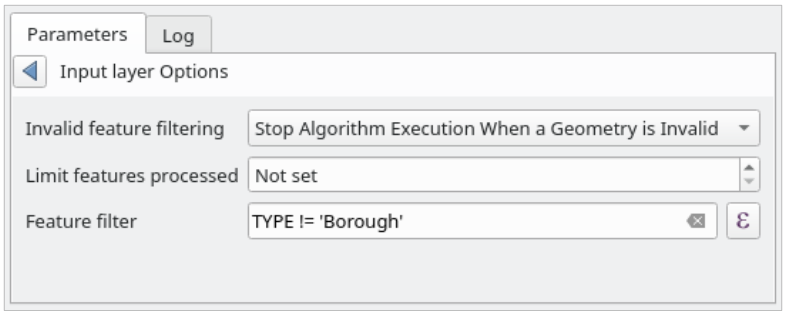


Figura23.10: Advanced options for vector input widget

  - It is also possible to limit the algorithm execution on the vector layer to its *Selected features only*.

- A **table**, to select from a list of all available in QGIS. Non-spatial tables are loaded into QGIS like vector layers, and use the *same widget*.
- A **raster layer**, to select from a list of all raster layers available in QGIS. The selector contains as well a … button on its right-hand side, to let you select filenames that represent layers currently not loaded in QGIS.

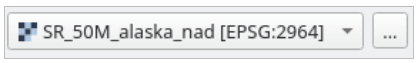


Figura23.11: Raster input widget

- Uma **opção**, para escolher a partir de uma lista de seleção de opções possíveis.

- A **numerical value**, to be introduced in a spin box. In some contexts (when the parameter applies at the feature level and not at the layer's), you will find a Data-defined override button by its side, allowing you to open the *expression builder* and enter a mathematical expression to generate variable values for the parameter. Some useful variables related to data loaded into QGIS can be added to your expression, so you can select a value derived from any of these variables, such as the cell size of a layer or the northernmost coordinate of another one.

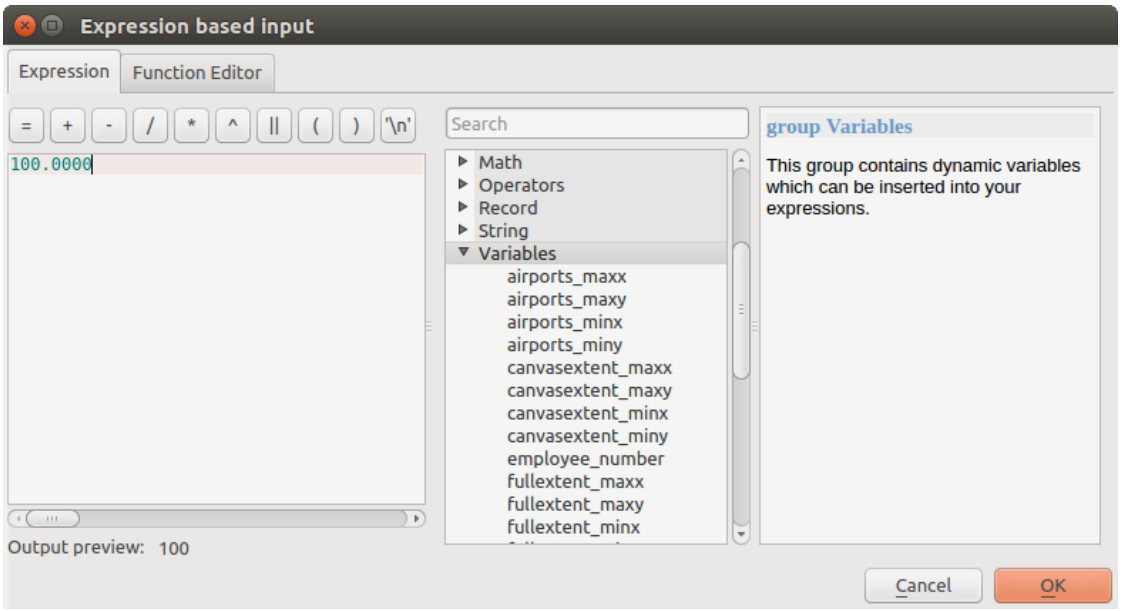


Figura23.12: Entrada baseada em expressão

- Um **intervalo**, com valores mínimo e máximo a serem introduzidos em duas caixas de texto.
- A **text string**, to be introduced in a text box.
- A **field**, to choose from the attributes table of a vector layer or a single table selected in another parameter.
- A **coordinate reference system**. You can select it among the recently used ones from the drop-down list or from the *CRS selection* dialog that appears when you click on the button on the right-hand side.

> **Dica**
>
> **Drag-and-drop as convenient shortcut for CRS selection**
>
> You can also drag a layer from the Layers panel and drop it onto the CRS widget. The widget will automatically set its CRS to match the dropped layer's.

- An **extent**, a text box defining a rectangle through its corners coordinate in the format `xmin, xmax, ymin, ymax`. Press the Set to current map canvas extent button to use the map canvas extent. Clicking the arrow on the right-hand side of the value selector, a pop-up menu will appear, giving you options to:
  - *Calculate from layer* ►: fills the text box with the coordinates of the bounding box of a layer to select among the loaded ones
  - *Calculate from layout map* ►: fills the text box with the coordinates of a map item selected from a layout in the current project
  - *Calculate from bookmark* ►: fills the text box with the coordinates of a saved bookmark
  - *Use current map canvas extent*

– *Draw on canvas*: the parameters window will hide itself, so you can click and drag onto the canvas. Once you have defined the extent rectangle, the dialog will reappear, containing the values in the extent text box.

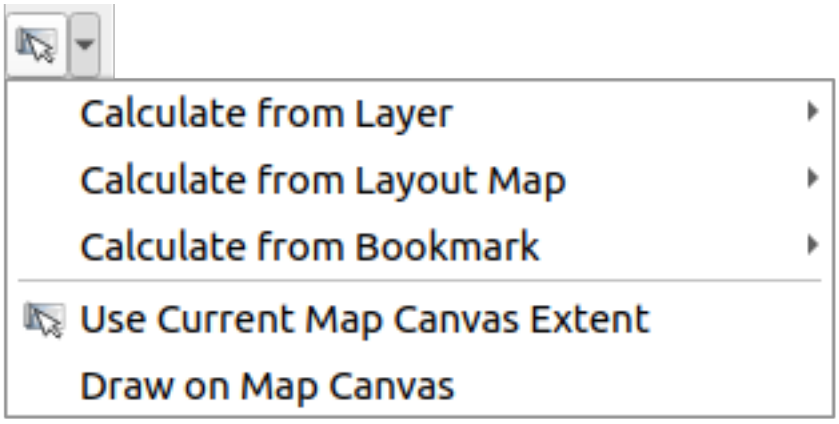


Figura23.13: Seletor de extensão

- A **list of elements** (whether raster or vector layers, tables, fields) to select from. Click on the … button at the left of the option to see a dialog like the following one. Multiple selection is allowed and when the dialog is closed, number of selected items is displayed in the parameter text box widget.

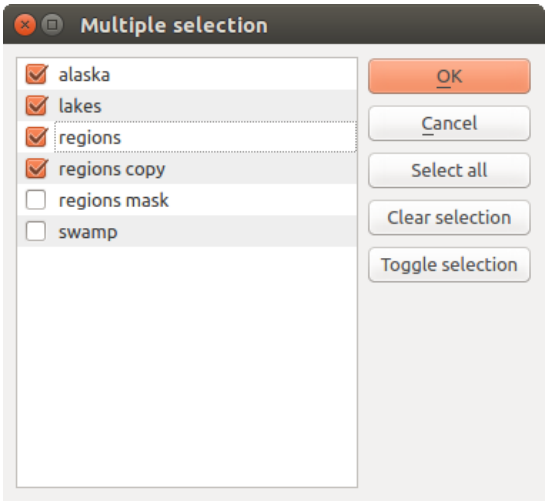


Figura23.14: Seleção Múltipla

- A **tabela pequena** a ser editada pelo usuário. Estes são usados para definir parâmetros como tabelas de pesquisa ou kernels de convolução, entre outros.

  Clique no botão do lado direito para ver a tabela e editar os seus valores.

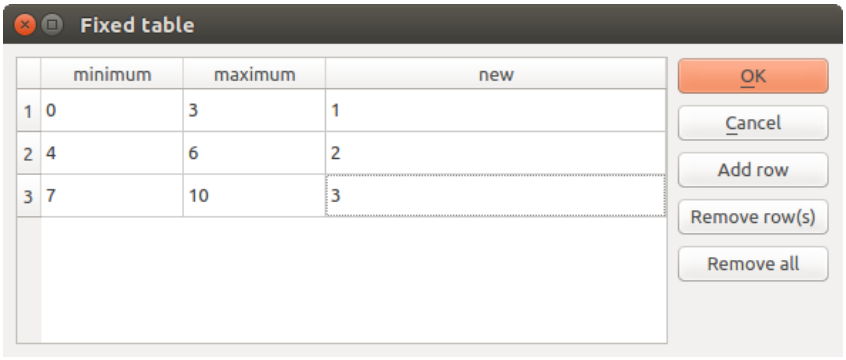


Figura23.15: Tabela fixa

Dependendo do algoritmo, o número de linhas pode ser modificado ou não, utilizando os botões do lado direito da janela.

Nota

Some algorithms require many parameters to run, e.g. in the *Calculadora Raster* you have to specify manually the cell size, the extent and the CRS. You can avoid to choose all the parameters manually when the algorithm has the `Reference layers` parameter. With this parameter you can choose the reference layer and all its properties (cell size, extent, CRS) will be used.

Logging the execution

Along with the *Parameters* tab, there is another tab named *Log* (see Fig.23.16 below). Information provided by the algorithm during its execution is written in this tab, allowing you to track the execution as well as being aware and having more details about the algorithm as it runs. You can directly click on the names of output files, folders, or HTML files listed in the log. Doing so will open the folder containing the generated file and automatically select it. Information on algorithm execution is also output in the *View* ► *Panels* ► *Log Messages Panel*.

Notice that not all algorithms write information to the *Log* tab, and many of them might run silently without producing any output other than the final files. Check the *Log Messages Panel* in that case.

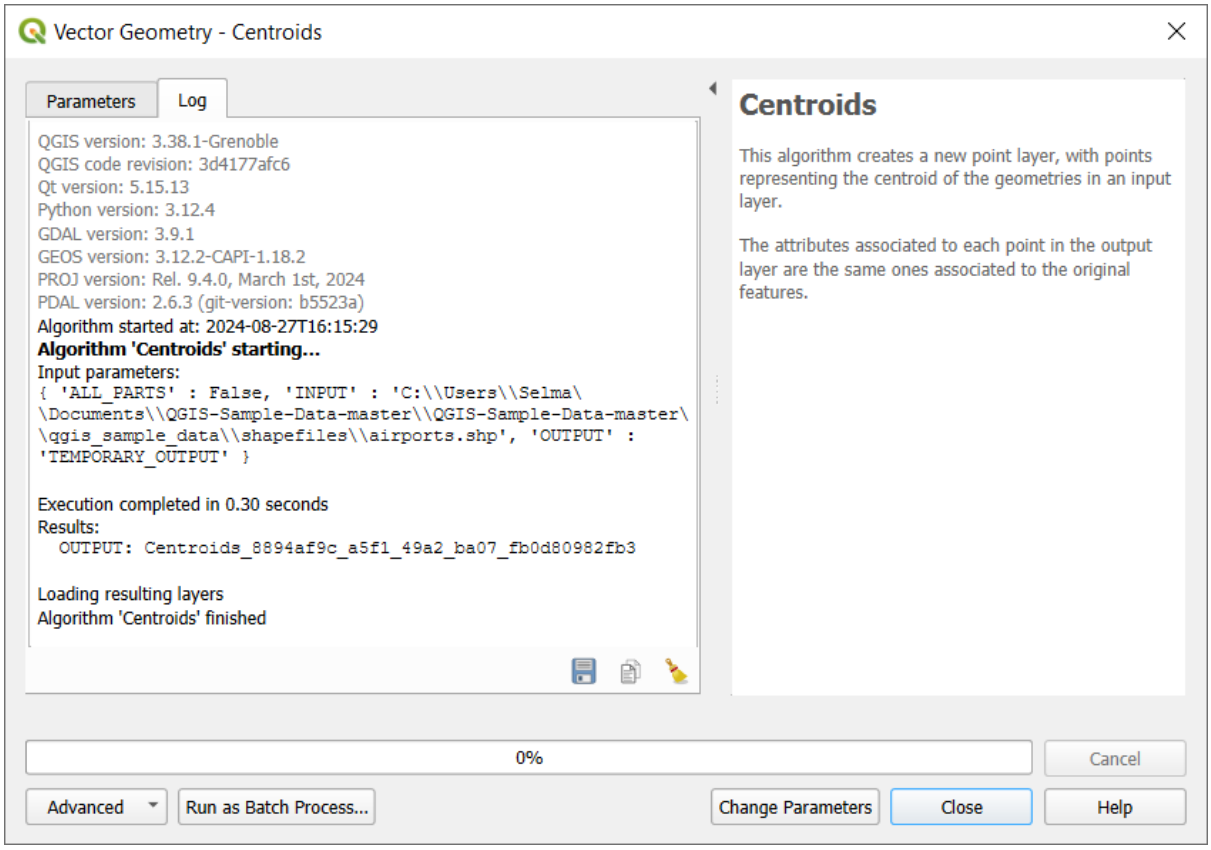


Figura23.16: Algorithm Dialog - Log

At the bottom of the *Log* tab you will find buttons to *Save Log to File*, *Copy Log to Clipboard* and *Clear Log*. These are particularly handy when you have checked the *Keep dialog open after running algorithm* in the *General* part of the Processing options.

Other tools

No lado direito da caixa de diálogo, você encontrará uma breve descrição do algoritmo, que o ajudará a entender seu propósito e suas ideias básicas. Se tal descrição não estiver disponível, o painel de descrição não será exibido.

For a more detailed help file, which might include description of every parameter it uses, or examples, you will find a *Help* button at the bottom of the dialog bringing you to the *Processing algorithms documentation* or to the provider documentation (for some third-party providers).

The *Advanced* ► menu provides functions to reuse the configuration defined in the dialog without running the algorithm:

- *Algorithm Settings...*: allows to override processing settings for the current algorithm execution. More details at *Override algorithm settings*.
- *Copy as Python Command*: allows for easy copying of the equivalent *PyQGIS command* to run the tool using the parameters defined in the dialog
- *Copy as qgis_process Command*: allows for easy generation of *qgis_process command*, including its environment settings like the distance units, area units, ellipsoid, and any tricky parameter values like GeoPackage outputs with specific layers
- *Copy as JSON*: all the settings of the command are copied in a `JSON` format, ready to be consumed by qgis_process. This is a convenient way to see the expected format of the commands, even for complex

parameters (like TIN interpolation parameters). You can store these easily and then restore them later by pasting the values.

- *Paste Settings* in a `JSON` format

The *Run as Batch Process…* button triggers the *batch processing mode* allowing to configure and run multiple instances of the algorithm with a variety of parameters. A *Run as Single Process…* button helps you switch back from the batch mode.

When an algorithm execution finishes (either successfully or not), a new button *Change Parameters* is shown as long as the *Log* tab is active.

Override algorithm settings

Triggered from within the *Advanced* drop-down menu at the bottom of an algorithm dialog, the *Algorithm Settings…* shows a panel allowing users to control general processing settings which apply to that algorithm execution only. It is intended to be a place where a user can override their *global processing settings* on an ad-hoc basis without having to change their usual default settings.

Settings that can be overridden are:

- *Invalid feature filtering*: unlike the existing per-parameter setting override for this, setting the handling method here will apply to **ALL inputs** for the algorithm
- *Calculation settings*, such as *Distance units* and *Area units* to use for distance/area measurements
- *Environment settings*, such as *Temporary folder* and *Number of threads to use*

Uma nota nas projeções

Processing algorithm execution are always performed in the input layer coordinate reference system (CRS). Due to QGIS's on-the-fly reprojecting capabilities, although two layers might seem to overlap and match, that might not be true if their original coordinates are used without reprojecting them onto a common coordinate system. Whenever you use more than one layer as input to a *QGIS native algorithm*, whether vector or raster, the layers will all be reprojected to match the coordinate reference system of the first input layer.

No entanto, isso é menos verdadeiro para a maioria dos aplicativos externos cujos algoritmos são expostos por meio da estrutura de processamento, pois assumem que todas as camadas já estão em um sistema de coordenadas comum e prontas para serem analisadas.

By default, the parameters dialog will show a description of the CRS of each layer along with its name, making it easy to select layers that share the same CRS to be used as input layers. If you do not want to see this additional information, you can disable this functionality in the Processing settings dialog, unchecking the *Show layer CRS definition in selection boxes* option.

If you try to execute an algorithm using as input two or more layers with unmatching CRSs, a warning dialog will be shown. This occurs thanks to the *Warn before executing if layer CRS's do not match* option.

Pode continuar a executar o algoritmo, mas tenha atenção que na maioria dos casos irá produzir resultados errados, tais como, camadas vazias devido à falta de sobreposição das camadas usadas como arquivos de entrada.

Dica

Use algoritmos de processamento para fazer reprojeção intermediária

Quando um algoritmo não pode executar com sucesso em várias camadas de entrada devido a SRC's não correspondentes, utilize o algoritmo interno QGIS como :ref:`qgisreprojetarcamada' para executar a reprojeção de camadas para a mesma SRC antes de executar o algoritmo utilizando estas saídas.

23.3.2 Objetos de dados gerados por algoritmos

Data objects generated by an algorithm can be of different types. They can be a layer (vector with or without geometry, raster, mesh, point cloud, ...), stored as a plain file on disk or in a database, text-based files such as `qml` style file or graphics `.html` file, a folder, ...

The parameters dialog will contain a widget corresponding to the needed output, where you can type the output channel to use for saving it. An output channel contains the information needed to save the resulting object somewhere. In the most usual case, you will save it to a file, but in the case of vector layers, and when they are generated by native algorithms (algorithms not using external applications) you can also save to a PostgreSQL, GeoPackage or SpatiaLite database, or a memory layer.

The **output** parameter for setting the result of the algorithm execution provides a text box for filling the path to the output result, and a drop-down ... button for additional options (the availability depends on the algorithm you run):

- *Skip Output*: if you are not interested in a given output of the algorithm
- *Create Temporary Layer* (`TEMPORARY_OUTPUT`): the output is stored in a vector *temporary scratch layer*.
- *Save to File...*: you will be prompted with a save file dialog, where you can select the desired file path. Supported file extensions are shown in the file format selector of the dialog, depending on the kind of output and the algorithm.

  The format of the output is defined by the filename extension. The supported formats depend on what is supported by the algorithm itself. To select a format, just select the corresponding file extension (or add it, if you are directly typing the file path instead). If the extension of the file path you entered does not match any of the supported formats, a default extension will be appended to the file path, and the file format corresponding to that extension will be used to save the layer or table. Default extensions are `.dbf` for tables, `.tif` for raster layers and `.gpkg` for vector layers. These can be modified in the setting dialog, selecting any other of the formats supported by QGIS.

  You can set a default folder for output data objects. Go to the settings dialog (you can open it from the *Settings* ► *Options* ► *Processing* menu), and in the *General* group, you will find a parameter named *Output folder*. This output folder is used as the default path in case you type just a filename with no path (i.e., `myfile.shp`) when executing an algorithm.
- *Save to a Temporary File*: if you do not enter any filename in the output text box (or select the corresponding option in the context menu), the result will be saved as a temporary file in the corresponding default file format, and it will be deleted once you exit QGIS (take care with that, in case you save your project and it contains temporary layers). The file is saved in the default `TEMP` folder on the machine, if *not overridden*.
- *Save to Geopackage...*: the output layer will be saved as a named table of a new or existing GeoPackage database
- *Save to Database Table...*: the output layer will be saved as a new table in a database such as GeoPackage, SpatiaLite, PostgreSQL or MS SQL Server. A connection should already exist. You'll need to provide the name of the new table and depending on the type of target database, select a schema,
- *Append to Layer...*: adds features output by the algorithm to an existing layer. Upon selecting the target layer, a field mapping panel opens, allowing to manually set how fields from the output layer are mapped to the target layer's fields.
- *Save to Directory*: Use a new or existing directory for storing the algorithm outputs
- *Save to a Temporary Directory*: creates a directory within the set `TEMP` folder for storing the algorithm outputs. The files will be deleted once you exit QGIS.

Ao executar um algoritmo que usa uma camada de vetor no modo interativo, o caminho do arquivo digitado é utilizado como o caminho de base para todos os arquivos gerados, que são nomeados com o nome de base e acrescentando-se um número que representa o índice da iteração. A extensão do arquivo (e formato) é usado para todos esses arquivos gerados.

Algorithms also generate graphics and text as HTML files. These results are shown at the end of the algorithm execution in the *Results Viewer* dialog. This dialog will keep the results produced by any algorithm during the current session, and can be shown at any time by selecting *Processing* ► *Results Viewer* from the QGIS main menu.

For layer-based outputs, it is possible to *Open output file after running algorithm* directly in QGIS, as a new entry in the *Layers* panel. By default, the files are opened.

23.4 Gerenciador do histórico

23.4.1 O histórico do processamento

Toda vez que você executa um algoritmo, as informações sobre o processo são armazenadas no gerenciador de histórico. A data e hora da execução são salvas, juntamente com os parâmetros utilizados, facilitando o rastreamento e controle de todo o trabalho que foi desenvolvido utilizando a estrutura de Processamento, e a sua reprodução.

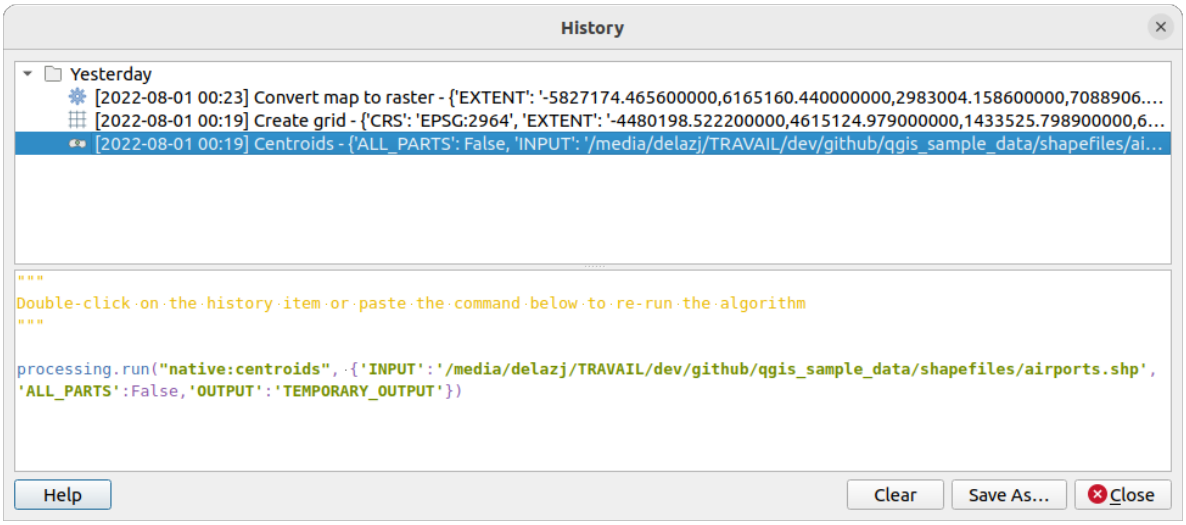


Figura23.17: História

As informações do processo são mantidas como uma expressão de linha de comando, mesmo que o algoritmo tenha sido lançado da caixa de ferramentas. Isto o torna útil para aqueles que estão aprendendo como usar a interface da linha de comando, já que podem chamar um algoritmo usando a caixa de ferramentas e depois verificar o gerenciador de histórico para ver como ele poderia ser chamado a partir da linha de comando.

Right-click on a row and you should be able to:

- *Copy as Python Command*: allows for easy copying of the equivalent *PyQGIS command* run from the dialog. Same as the code displayed below the commands list.
- *Copy as qgis_process Command*: allows for easy generation of *qgis_process command*, including its environment settings like the distance units, area units, ellipsoid, and any tricky parameter values like GeoPackage outputs with specific layers
- *Copy as JSON*: all the settings of the command are copied in a `JSON` format, ready to be consumed by qgis_process. This is a convenient way to see the expected format of the commands, even for complex parameters (like TIN interpolation parameters). You can store these easily and then restore them later by pasting the values over an algorithm dialog.
- *Create Test...* using the concerned algorithm and parameters, following instructions at Processing README file.

Além de navegar pelas entradas no registro, você também pode reexecutar os processos simplesmente clicando duas vezes sobre a entrada. A janela de do algoritmo abre-se então com parâmetros já definidos, e você pode alterar qualquer um deles para atender às suas necessidades e executar novamente o algoritmo.

The *History* dialog also provides a convenient way to contribute to the consolidation of the testing infrastructure of QGIS Processing algorithms and scripts.

23.4.2 O log do processamento

The history dialog only contains the execution calls, but not the information produced by the algorithm when executed. That information is written to the QGIS log (*View* ► *Panels* ► *Log Messages Panel*).

Third-party algorithms are usually executed by using their command-line interfaces, which communicate with the user via the console. Although that console is not shown, usually a full dump of it is written to the log each time you run one of those algorithms. To avoid cluttering the log with that information, you can disable it for each provider in the settings dialog.

Some algorithms, even if they can produce a result with the given input data, output comments or additional information to log when they detect potential problems with the data, in order to warn you. Make sure you check those messages in the log if you get unexpected results.

23.5 The model designer

The *model designer* allows you to create complex models using a simple and easy-to-use interface. When working with a GIS, most analysis operations are not isolated, rather part of a chain of operations. Using the model designer, that chain of operations can be wrapped into a single process, making it convenient to execute later with a different set of inputs. No matter how many steps and different algorithms it involves, a model is executed as a single algorithm, saving time and effort.

The model designer can be opened from the Processing menu (*Processing* ► *Model Designer*).

23.5.1 The model designer interface

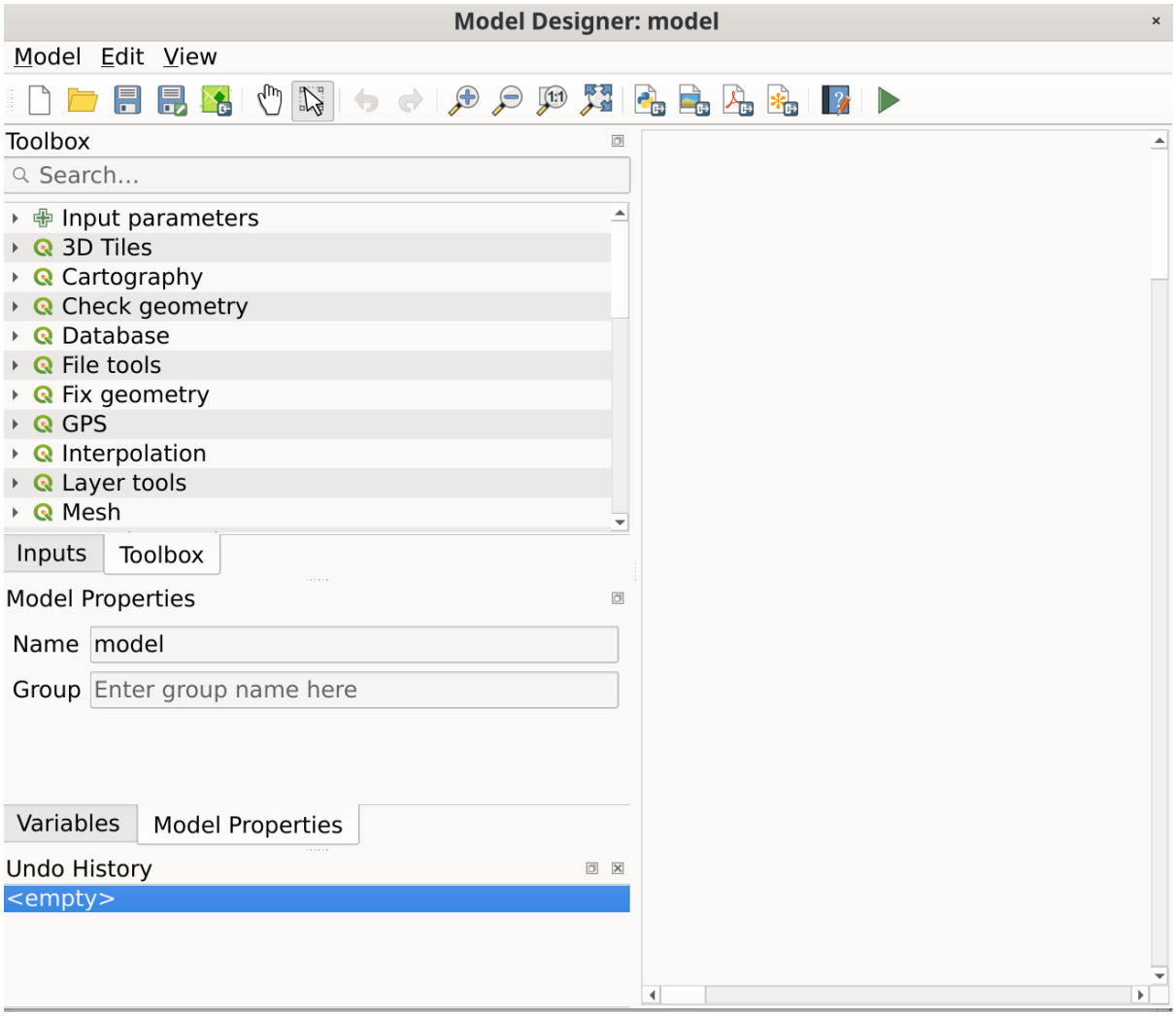


Figura23.18: Model designer

In its main part, the modeler has a working canvas where the structure of the model and the workflow it represents can be constructed.

At the top of the dialog, different menus and the *Navigation* toolbar give access to a variety of tools.

Menu modelo

Etiqueta	Atalho	Barra de ferramentas de navegação	Descrição
Validate Model			Checks whether the algorithms and inputs used in the model exist. Convenient before releasing a model.
Run Model…	F5	☑	Executa o modelo
Run Selected Steps…	Shift+F5		Runs only the selected steps in a model, allowing to run a subset of the model
guilabel:*Reordenar entradas do modelo…*			Sets the order in which inputs are presented to the user in the algorithm dialog.
Reorder Output Layers…			Sets a specific order which the outputs from the model must use when loading the results into a project.
Open Model…	Ctrl+O	☑	Opens a `.model3` file for edit or execution
Salvar Modelo	Ctrl+S	☑	Saves the model to disk as a `.model3` file
Salvar Modelo como…	Ctrl+Shi	☑	Saves the model to disk as a new `.model3` file
Save Model in project		☑	Incorpora o arquivo de modelo no arquivo de projeto, tornando-o disponível ao compartilhar o arquivo de projeto.
Edit Model Help…		☑	Uma interface para documentar o modelo, os algoritmos, os parâmetros e as saídas, bem como o autor e versão
Exportar ►			
► *Exportar como Imagem…*		☑	Salva o design gráfico do modelo em um formato de arquivo de imagem (para fins de ilustração)
► *Exportar como PDF…*			Saves the model's graphical design to a `PDF` file format (for illustration purpose)
► *Exportar como SVG…*			Saves the model's graphical design to an `SVG` file format (for illustration purpose)
► *Export as Script Algorithm…*		☑	Gera um arquivo de script python incluindo as instruções do modelo

Editar menu

Etiqueta	Atalho	Barra de ferramentas de navegação	Descrição
Selecionar tudo	`Ctrl+A`		Seleciona todos os componentes do modelo no design
Snap selected components to Grid			snaps and aligns the elements into a grid
Refazer	`Ctrl+Y`	☑	Rollback the latest canceled action. See also the *Undo/Redo* panel.
Desfazer	`Ctrl+Z`	☑	Cancel the previous change. See also the *Undo/Redo* panel.
Cortar	`Ctrl+X`		Corta uma seleção de componentes do modelo.
Copiar	`Ctrl+C`		Copia uma seleção de componentes do modelo.
Colar	`Ctrl+V`		Cola uma seleção recortada ou copiada de componentes de um modelo para outro ou dentro do mesmo modelo. Os componentes selecionados mantêm suas propriedades e comentários originais.
Delete selected components	`Del`		Remove um componente do modelo.
Add Group Box			Adds a box at the background of related components in order to visually group them. Particularly useful in big models to keep the workflow clean.

ver menu

Etiqueta	Atalho	Barra de ferramentas de navegação	Descrição
Zoom To ►			Zooms to the selected group box extent
Zoom In	`Ctrl++`	☑	
Reduzir Zoom	`Ctrl+-`	☑	
Zoom to 100%	`Ctrl+1`	☑	
Zoom Full	`Ctrl+0`	☑	Displays all the components in the designer current canvas
Mostrar comentários			Displays comments associated to every algorithm or input in the model designer
Enable Snapping			
Toggle Panel Visibility	`Ctrl+Tab`		Switches ON or OFF the *panels* in the designer

Painéis

The left part of the window is a section with five panels that can be used to add new elements to the model:

1. *Inputs*: all the *input parameters* that could shape your model
2. *Toolbox*: here you can find available *Processing algorithms*. It also includes input parameters. Use *Search...* box to find algorithms or inputs.
3. *Model Properties*: specify the name (required) of the model and the group in which it will be displayed in the *Processing Toolbox*
4. *Variables*: Models can contain dedicated *variables* that are unique and only available to them. These variables can be accessed by any expression used within the model. They are useful to control algorithms within a model and control multiple aspects of the model by changing a single variable. The variables can be viewed and modified in the *Variables* panel.
5. *Undo History*: this panel will register everything that happens in the modeler, making it easy to cancel things you did wrong.

Sobre os algoritmos disponíveis

Some algorithms that can be executed from the *Processing* ► *Toolbox* do not appear in the list of available algorithms when you are designing a model. To be included in a model, an algorithm must have the correct semantic. If an algorithm does not have such a well-defined semantic (for instance, if the number of output layers cannot be known in advance), then it is not possible to use it within a model, and it will not appear in the list of algorithms that you can find in the modeler dialog. On the other hand some algorithms are specific to the modeler. Those algorithms are located within the group 'Modeler Tools'.

23.5.2 Criando um modelo

A criação de um modelo envolve duas etapas básicas:

1. *Definition of necessary inputs*. These inputs will be added to the parameters window, so the user can set their values when executing the model. The model itself is an algorithm, so the parameters window is generated automatically as for all algorithms available in the Processing framework.
2. *Definition of the workflow*. Using the input data of the model, the workflow is defined by adding algorithms and selecting how they use the defined inputs or the outputs generated by other algorithms in the model.

Definição das entradas

The first step is to define the inputs for the model. They are found in the *Inputs* panel on the left side of the modeler window. Hovering with the mouse over the inputs will show a tooltip with additional information. For a full list of available parameters in modeler and their correspondance for scripting, please read *Tipos de entrada e saída para Algoritmos de Processamento*.

When double-clicking on an element, a dialog is shown that lets you define its characteristics. Depending on the parameter, the dialog will contain at least one element (the description, which is what the user will see when executing the model). For example, when adding a numerical value, as can be seen in the next figure, in addition to the description of the parameter, you have to set a default value and the range of valid values.

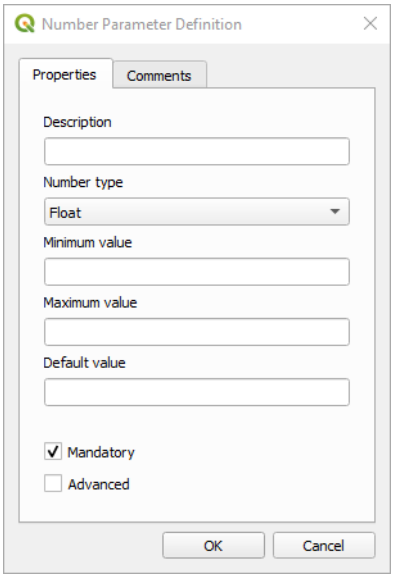


Figura23.19: Definição dos Parâmetros do Modelo

You can define your input as mandatory for your model by checking the Mandatory option and by checking the Advanced checkbox you can set the input to be within the Advanced section. This is particularly useful when the model has many parameters and some of them are not trivial, but you still want to choose them.

For each added input, a new element is added to the modeler canvas.

Figura23.20: Parâmetros do modelo

You can also add inputs by dragging the input type from the list and dropping it at the position where you want it in the modeler canvas. If you want to change a parameter of an existing input, just double click on it, and the same dialog will pop up.

When using a model within another model, the inputs and outputs necessary will be displayed in the canvas.

Definição do fluxo de trabalho

In the following example we will add two inputs and two algorithms. The aim of the model is to copy the elevation values from a DEM raster layer to a line layer using the Drape algorithm, and then calculate the total ascent of the line layer using the Climb Along Line algorithm.

In the *Inputs* tab, choose the two inputs as Vector Layer for the line and Raster Layer for the DEM. We are now ready to add the algorithms to the workflow.

Algorithms can be found in the *Toolbox* panel, grouped much in the same way as they are in the Processing toolbox.

Figura23.21: Model Inputs

To add an algorithm to a model, double-click on its name or drag and drop it, just like for inputs. As for the inputs you can change the description of the algorithm and add a comment. When adding an algorithm, an execution dialog

will appear, with a content similar to the one found in the execution panel that is shown when executing the algorithm from the toolbox. The following picture shows both the `Drape (set Z value from raster)` and the `Climb along line` algorithm dialogs.

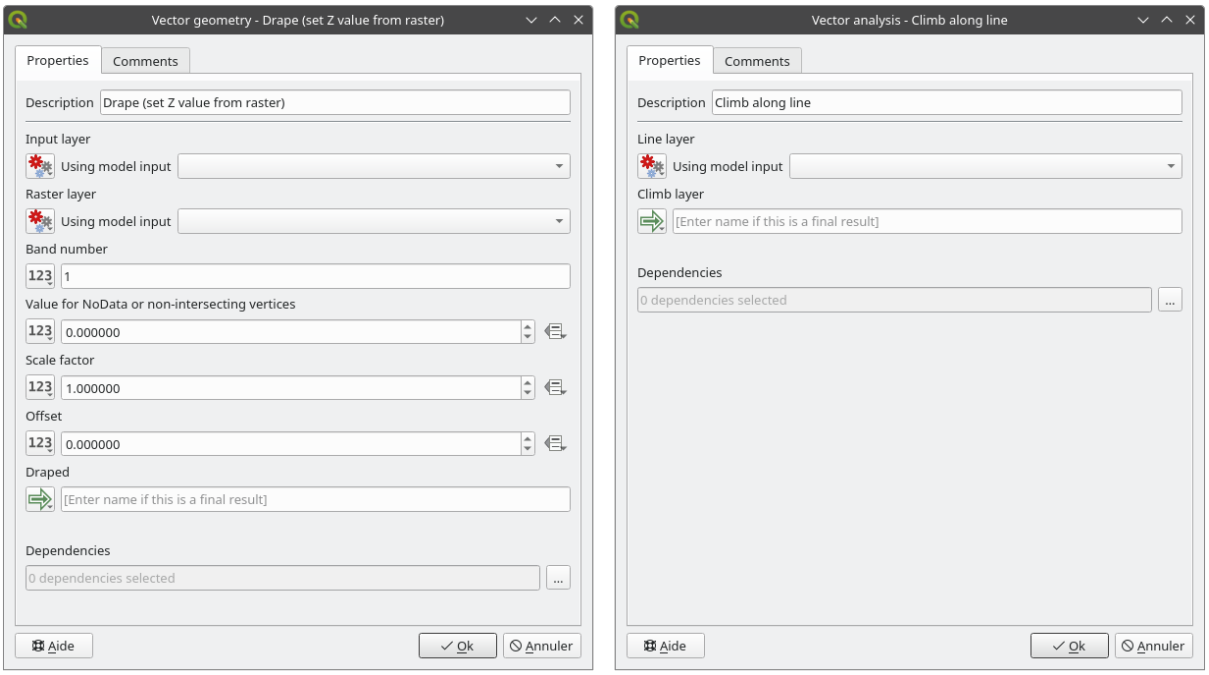


Figura23.22: Model Algorithm parameters

As you can see, there are however some differences. Each parameter has a drop-down menu next to it allowing to control how it will be served during the workflow:

- Value: allows you to assign a static value to the parameter. Depending on the parameter type, the widget will let you enter a number (`5.0`), a string (`mytext`), select layer(s) loaded in the QGIS project or from a folder, pick items from a list, …
- Pre-calculated Value: opens the *Expression Builder* dialog and lets you define an expression to fill the parameter. Model inputs together with some other layer statistics are available as **variables** and are listed at the top of the Search dialog of the Expression Builder. The expression is evaluated once before the child algorithm is executed and used during the execution of that algorithm.
- Model Input: allows to use an input added to the model as a parameter. Once clicked, this option will list all the suitable inputs for the parameter.
- Algorithm Output: allows to use the output of another algorithm as an input of the current algorithm. As of model inputs, this option will list all the suitable inputs for the parameter.
- The **output parameter** also has the above options in its drop-down menu:
  - add static outputs for child algorithms, e.g. always saving a child algorithm's output to a predefined geopackage or postgres layer
  - use an expression based output values for child algorithms, e.g. generating an automatic file name based on today's date and saving outputs to that file
  - use a model input, e.g. the *File/Folder* model input to specify an output file or folder
  - use another algorithm output, e.g. the output of the *Create directory* algorithm (from *Modeler tools*)
  - an addditional Model Output option makes the output of the algorithm available in the model. If a layer generated by the algorithm is only to be used as input to another algorithm, don't edit that text box.

In the following picture you can see the two input parameters defined as `Model Input` and the temporary output layer:

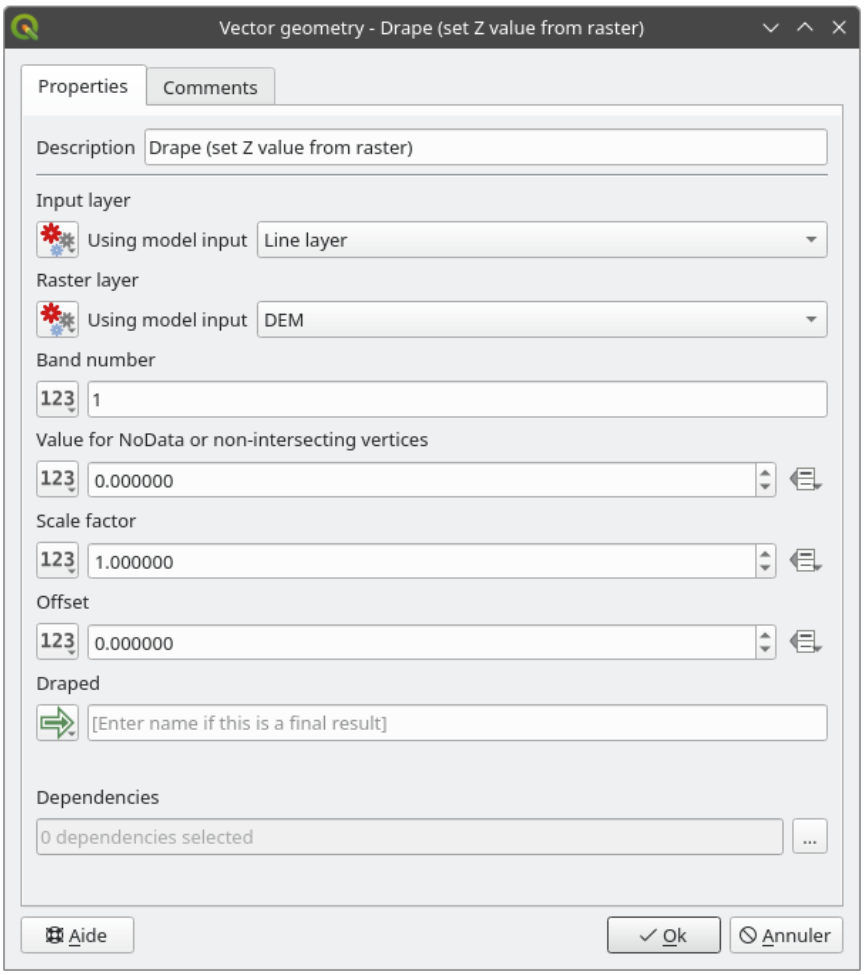


Figura23.23: Parâmetros de entrada e saída de algoritmo

You will also find an additional parameter named *Dependencies* that is not available when calling the algorithm from the toolbox. This parameter allows you to define the order in which algorithms are executed, by explicitly defining one algorithm as a *parent* of the current one. This will force the *parent* algorithm to be executed before the current one.

When you use the output of a previous algorithm as the input of your algorithm, that implicitly sets the previous algorithm as parent of the current one (and places the corresponding arrow in the modeler canvas). However, in some cases an algorithm might depend on another one even if it does not use any output object from it (for instance, an algorithm that executes a SQL sentence on a PostgreSQL database and another one that imports a layer into that same database). In that case, just select the previous algorithm in the *Dependencies* parameter and they will be executed in the correct order.

Once all the parameters have been assigned valid values, click on *OK* and the algorithm will be added to the canvas. It will be linked to the elements in the canvas (algorithms or inputs) that provide objects that are used as inputs for the algorithm.

Elements can be dragged to a different position on the canvas using the Select/Move/Link Item tool. This is useful to make the structure of the model clearer and more intuitive. You can also resize the elements, grasping their border. This is particularly useful if the description of the input or algorithm is long. With *View* ► *Enable snapping* option checked, items resizing or displacement can be bound to a virtual grid, for a more visually structured algorithm design.

Os links entre os elementos são atualizados automaticamente e você pode ver um botão + na parte superior e inferior

de cada algoritmo. Clicar no botão listará todas as entradas e saídas do algoritmo para que você possa ter uma visão geral rápida.

You can create or remove connections between model components using an interactive drag-and-drop interface. Each parameter and algorithm item on the model canvas includes small circular sockets, output sockets on the right and input sockets on the left. To connect components, click and drag from an output socket (either from a model input or an algorithm) to an input socket on another algorithm. A visual edge follows the cursor and completes the connection when released. To remove a connection, drag the edge away from the input socket. The algorithm dialogs update automatically to reflect changes made using the drag-and-drop interface.

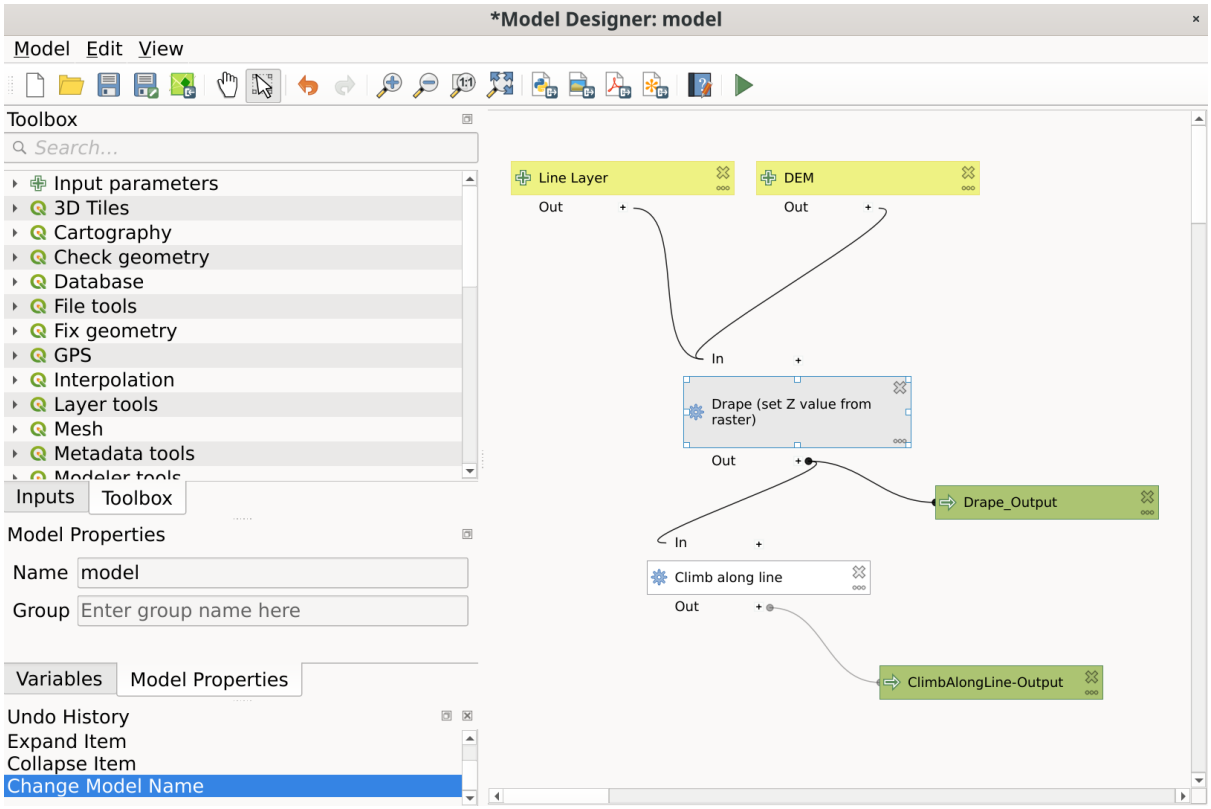


Figura23.24: Um modelo completo

With the *Edit* ► *Add Group Box* tool, you can add a draggable *box* to the canvas. This feature is very useful in big models to group related elements in the modeler canvas and to keep the workflow clean. For example we might group together all the algorithms of the example:

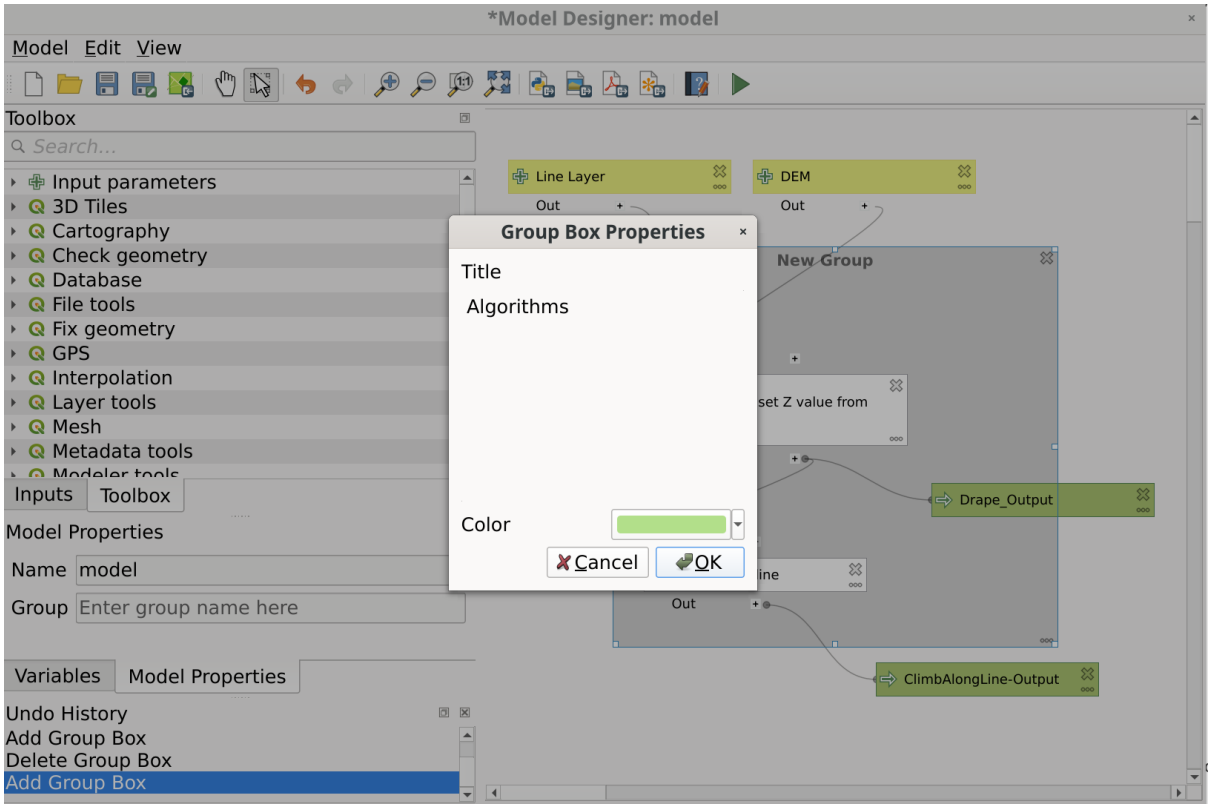


Figura23.25: Model Group Box

You can change the name and the color of the boxes. Group boxes are very useful when used together with *View* ► *Zoom To* ► tool, allowing you to zoom to a specific part of the model. You can also zoom in and out by using the mouse wheel.

You might want to change the order of the inputs and how they are listed in the main model dialog. At the bottom of the `Input` panel you will find the `Reorder Model Inputs...` button and by clicking on it a new dialog pops up allowing you to change the order of the inputs:

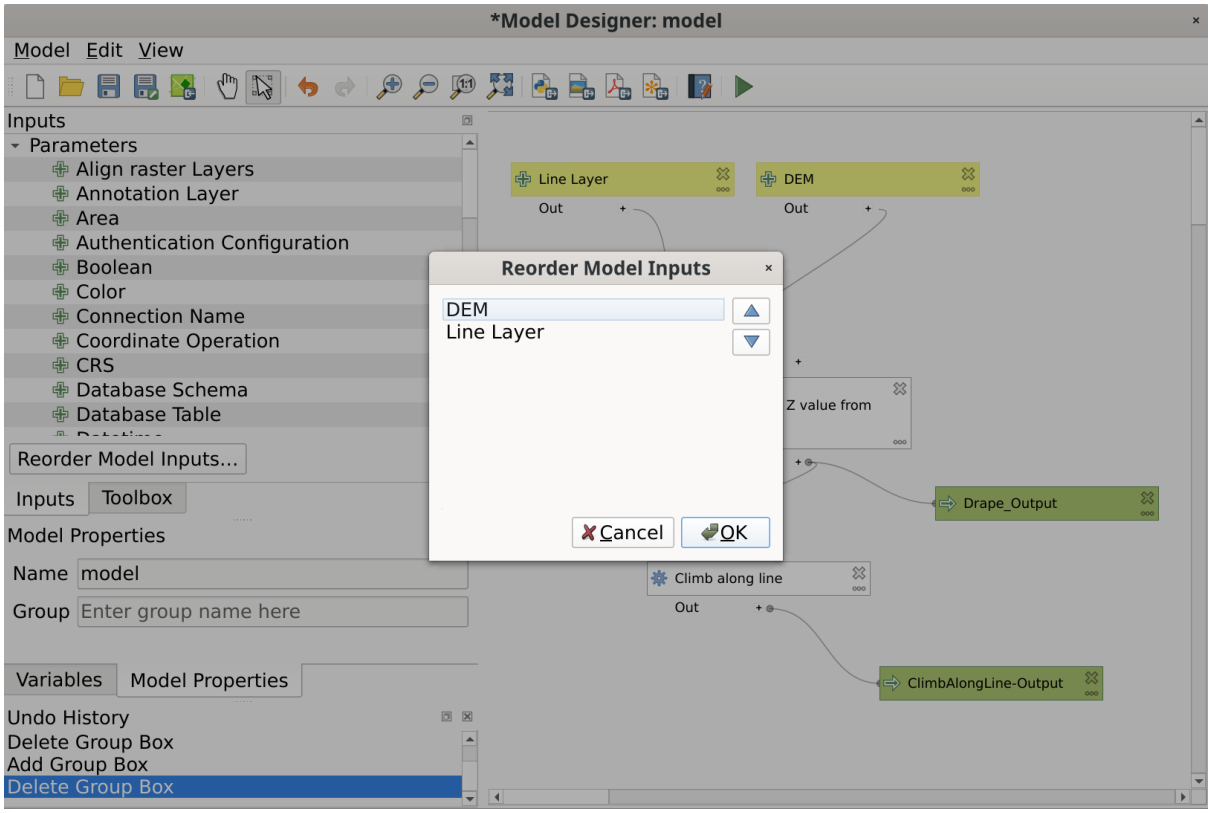


Figura23.26: Reorder Model Inputs

There is also the possibility to set a specific order which the outputs from the model must use when loading the results into a project. This gives the model creator a means of ensuring that layers are logically ordered on the canvas when running a model, such as placing a vector layer output over a raster layer output, or a point layer over a polygon layer. The model creator can also set an optional "Group name" for the outputs for automatically grouping outputs within the layer tree using a new group name or by adding them to an existing group. In the `Model` menu you will find the `Reorder Output Layers...` entry and by clicking on it a new dialog pops up allowing you to change the order of the output layers:

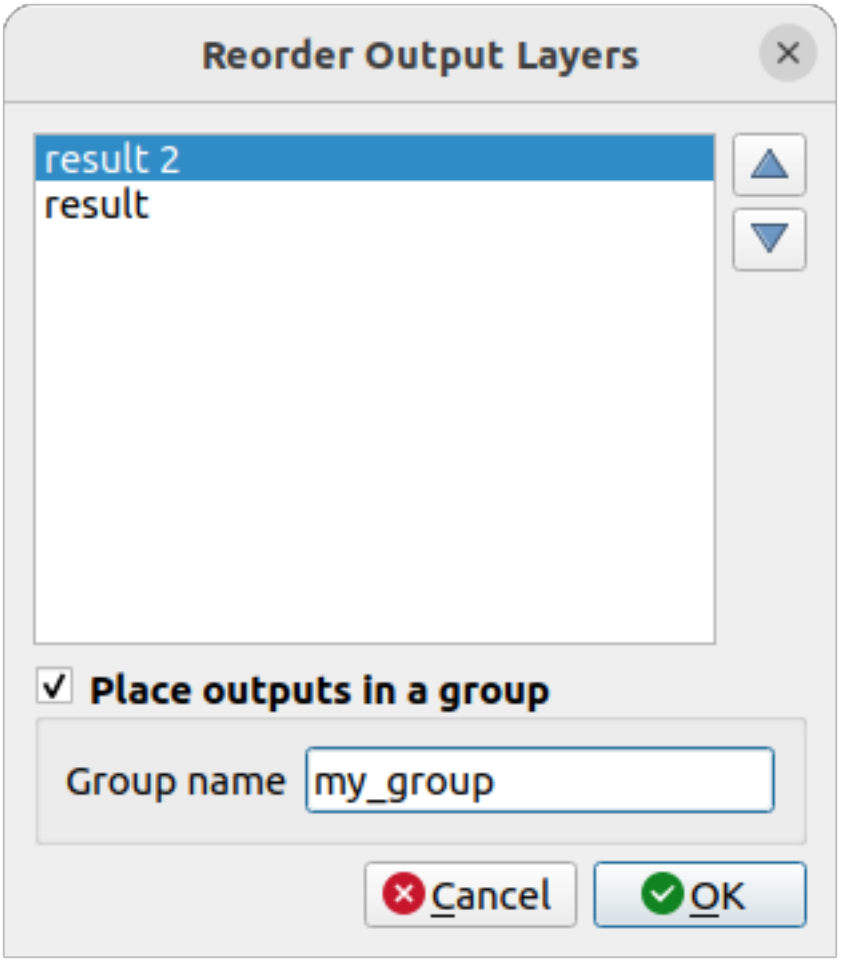


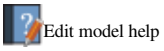
Figura23.27: Reorder Output Layers

Comments can also be added to inputs or algorithms present in the modeler. This can be done by going in the *Comment* tab of the item or with a right-click. In the same tab a color can be set manual for individual model comments. Comments are visible only in the modeler canvas and not in the final algorithm dialog; they can be hidden by deactivating *View* ► *Show Comments*.

Your model can be run in various ways:

- You can run the whole model by clicking on Run Model… from the toolbar, right-clicking on the model in the *Browser panel* or *Model* ► *Run Model…*: when using the editor to execute a model, any non-default values will be saved in the inputs. This means that executing the model at a later time from the editor will have the dialog prefilled with those values on any subsequent run.
- You can select elements of the model and run only that subset of the model: press the *Run Selected Steps…* option from the *Models* ► menu or from the contextual menu of a selected algorithm. The initial state will be taken from any previous executions of the model through the editor, so results from previous steps in the model are available for the selected steps. This makes it possible to fix parts of a large model, without having to constantly run the entire model to test. Especially useful when earlier steps in the model are time consuming.
- You can run a subset of the model, starting from a specific algorithm: right-click the algorithm and select *Run from Here…*. Likewise, values are taken from previous executions.
- In order to use the algorithm from the Processing Toolbox, it has to be saved and the modeler dialog closed, to allow the toolbox to refresh its contents.

Documentando seu modelo

You need to document your model, and this can be done from the modeler itself. Click on the 

button, and a dialog like the one shown next will appear.

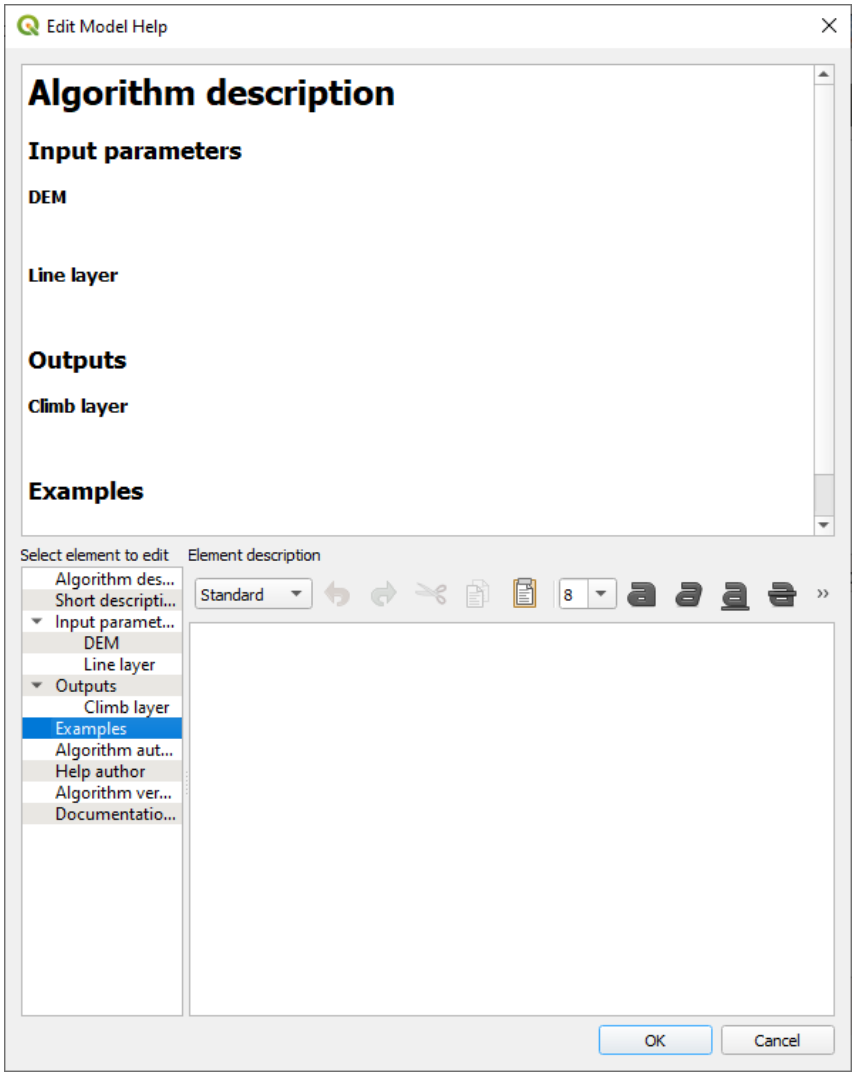


Figura23.28: Editing Help

On the right-hand side, you will see a simple HTML page, created using the description of the input parameters and outputs of the algorithm, along with some additional items like a general description of the model or its author. Also, there is an Example section where you can input your own custom examples to help explain the usage of the model. The first time you open the help editor, all these descriptions are empty, but you can edit them using the elements on the left-hand side of the dialog. Select an element on the upper part and then write its description in the text box below.

Model help is saved as part of the model itself.

23.5.3 Salvando e carregando os modelos

Salvando modelos

Use the Save model button to save the current model and the Open Model button to open a previously saved model. Models are saved with the `.model3` extension. If the model has already been saved from the modeler window, you will not be prompted for a filename. Since there is already a file associated with the model, that file will be used for subsequent saves.

Antes de salvar um modelo, você deve inserir um nome e um grupo para ele nas caixas de texto na parte superior da janela.

Models saved in the `models` folder (the default folder when you are prompted for a filename to save the model) will appear in the toolbox in the corresponding branch. When the toolbox is invoked, it searches the `models` folder for files with the `.model3` extension and loads the models they contain. Since a model is itself an algorithm, it can be added to the toolbox just like any other algorithm.

Models can also be saved within the project file using the Save model in project button. Models saved using this method won't be written as `.model3` files on the disk but will be embedded in the project file.

Project models are available in the *Project models* menu of the toolbox and in the *Project* ► *Models* menu item.

The models folder can be set from the Processing configuration dialog, under the *Modeler* group.

Models loaded from the `models` folder appear not only in the toolbox, but also in the algorithms tree in the *Algorithms* tab of the modeler window. That means that you can incorporate a model as a part of a bigger model, just like other algorithms.

Models will show up in the *Browser* panel and can be run from there.

Exportando um modelo como um script Python

As we will see in a later chapter, Processing algorithms can be called from the QGIS Python console, and new Processing algorithms can be created using Python. A quick way to create such a Python script is to create a model and then export it as a Python file.

To do so, click on the Export as Script Algorithm… in the modeler canvas or right click on the name of the model in the Processing Toolbox and choose Export Model as Python Algorithm….

Exportando um modelo como imagem, PDF ou SVG

A model can also be exported as an image, SVG or PDF (for illustration purposes) by clicking Export as image, Export as PDF or Export as SVG.

23.5.4 Editando um modelo

Você pode editar o modelo que está criando atualmente, redefinindo o fluxo de trabalho e os relacionamentos entre os algoritmos e as entradas que definem o modelo.

You can also edit existing model from the *Browser panel* choosing *Edit Model…* option.

If you right-click on an algorithm in the canvas, you will see a context menu like the one shown next:

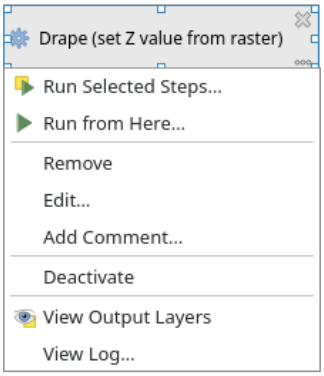


Figura23.29: Modeler Right Click

- Options for *running part of the model* have been exposed earlier.
- Selecting the *Remove* option will cause the selected algorithm to be removed. An algorithm can be removed only if there are no other algorithms depending on it. That is, if no output from the algorithm is used as input in a different one.
- Selecting the *Edit…* option will show the parameter dialog so you can change the inputs and parameter values. Not all input elements available in the model will appear as available inputs. Layers or values generated at a more advanced step in the workflow defined by the model will not be available if they cause circular dependencies.

  Select the new values and click on the *OK* button as usual. The connections between the model elements will change in the modeler canvas accordingly.
- The *Add comment…* allows you to add a comment to the algorithm to better describe the behavior.

Nota

Right-clicking an input parameter will also allow you to *Remove*, *Edit…* and *Add comment…*

- A model can be run partially by deactivating some of its algorithms. To do it, select the *Deactivate* option in the context menu that appears when right-clicking on an algorithm element. The selected algorithm, and all the ones in the model that depend on it will be displayed in grey and will not be executed as part of the model.

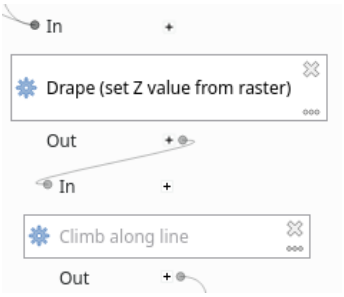


Figura23.30: Model with a deactivated algorithm

  When right-clicking on an algorithm that is not active, you will see an *Activate* menu option that you can use to reactivate it.
- When editing a model through the designer (and after having run that model), right-clicking any child step in the model allows to select *View Output Layers*. This will add the output layers from that step as new layers in the current QGIS project.

  This action is available for all child algorithms in the model, even if the model is not configured to use the outputs from those children as model outputs. This is designed as a helpful debugging action. If a model

fails (or gives unexpected results), you can then trace through the model and view the outputs for suspected problematic steps. It avoids the need to add temporary outputs to a model and re-run to test.

Additionally, this action is always available after running the model, even if the model itself failed (e.g., because of a misconfigured step later in the model).

- The *View Log* action helps you see the log for each child algorithm after you've closed down the algorithm dialog.

23.6 A interface de processamento em lote

23.6.1 Introducción

All algorithms (including models) can be executed as a batch process. That is, they can be executed using not just a single set of inputs, but several of them, executing the algorithm as many times as needed. This is useful when processing large amounts of data, since it is not necessary to launch the algorithm many times from the toolbox.

Para executar um algoritmo como um processamento em lote, clique com o direito do mouse na caixa de ferramentas e selecione a opção *Executar como processamento em lote* no balão de opções que irá aparecer.

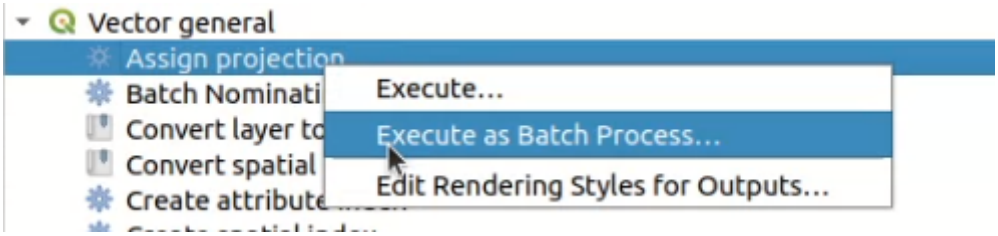


Figura23.31: Batch Processing from right-click

If you have the execution dialog of the algorithm open, you can also start the batch processing interface from there, clicking on the *Run as batch process...* button.

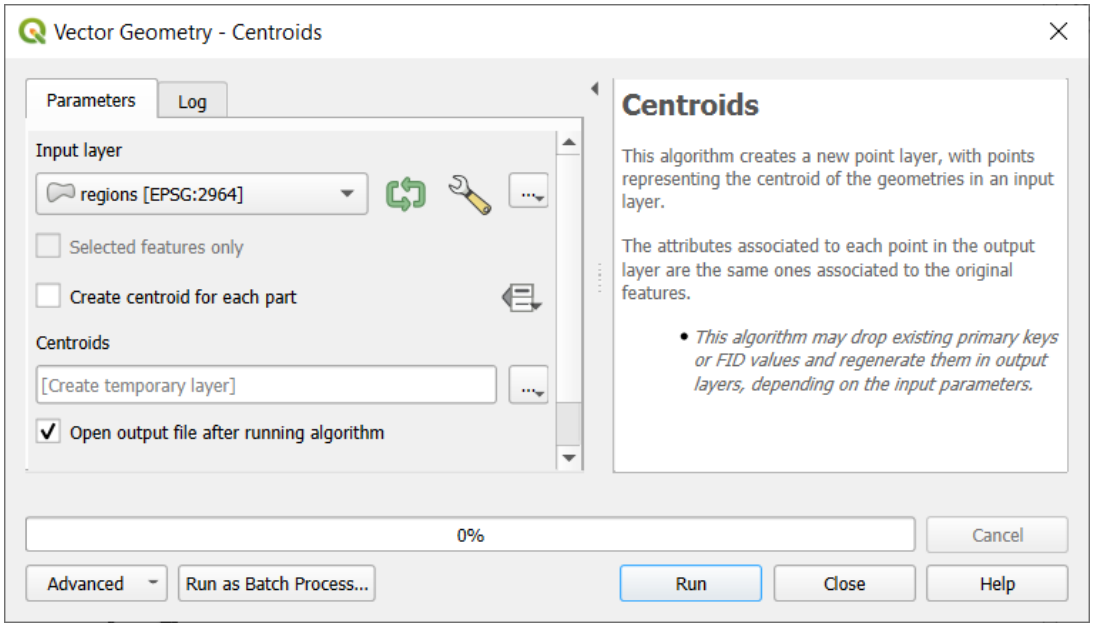


Figura23.32: Batch Processing From Algorithm Dialog

23.6.2 Os parâmetros da tabela

Executing a batch process is similar to performing a single execution of an algorithm. Parameter values have to be defined, but in this case we need not just a single value for each parameter, but a set of them instead, one for each time the algorithm has to be executed. Values are introduced using a table like the one shown next, where each row is an iteration and columns are the parameters of the algorithm.

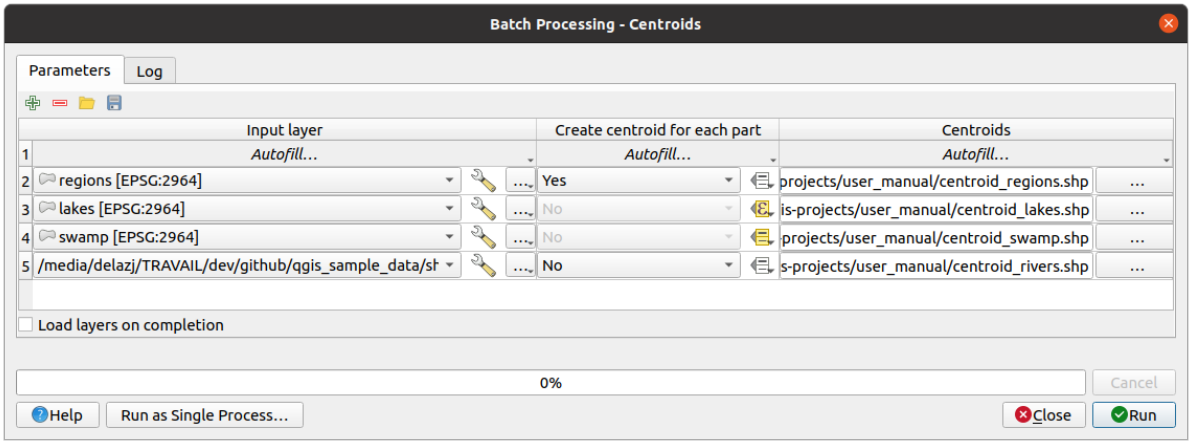


Figura23.33: Batch Processing

A partir da barra de ferramentas superior, você pode:

- Toggle advanced mode: Available only when the algorithm has parameters that are marked as advanced, this button allows to show or hide such parameters in the batch dialog.
- |adicionarSimbologia| Adicionar linha: adiciona uma nova entrada de processamento para configuração
- Remover linha(s): remover as linhas selecionadas da tabela. A seleção das linhas é feita clicando no número à esquerda e permite :ref:` combinação de teclado <interacting_features_table>` para seleção múltipla.
- Abrir um arquivo de configuração de processamento em lote
- Salvar a configuração de processamento em lote para um `JSON`, que pode ser executado posteriormente.

Por padrão, a tabela contém apenas duas linhas:

- The first row displays in each cell an *Autofill...* ► drop-down menu with *options* to quickly fill the cells below. Available options depend on the parameter type.
- A segunda linha (assim como cada uma subsequente) representa uma única execução do algoritmo, e cada célula contém o valor de um dos parâmetros. É semelhante à caixa de diálogo de parâmetros que você vê ao executar um algoritmo da caixa de ferramentas, mas com uma disposição diferente.

Na parte inferior da tabela, você pode definir a |caixa de seleção| *Carregar camadas após termino*

Uma vez que o tamanho da tabela for configurado, terá de ser preenchido com os valores desejados.

23.6.3 Preenchendo os parâmetros da tabela

For most parameters, setting the value is trivial. The appropriate widget, same as in the *single process dialog*, is provided, allowing to just type the value, or select it from a list of possible values, depending on the parameter type. This also includes data-define widget, when compatible.

Para automatizar a definição do processamento de lote e evitar o preenchimento da tabela célula por célula, você pode querer pressionar o menu *Autopreenchimento…* de um parâmetro e selecionar qualquer uma das seguintes opções para substituir os valores na coluna:

- guilabel:*Preencher abaixo* levará a entrada para o primeiro processo e será inserido em todos os outros processos.
- *Calculate by Expression…* will allow you to create a new QGIS expression to use to update all existing values within that column. Existing parameter values (including those from other columns) are available for use inside the expression via *variables*. E.g. setting the number of segments based on the buffer distance of each layer:

```
CASE WHEN @DISTANCE > 20 THEN 12 ELSE 8 END
```

- *Add Values by Expression…* will add new rows using the values from an expression which returns an array (as opposed to *Calculate by Expression…*, which works only on existing rows). The intended use case is to allow populating the batch dialog using complex numeric series. For example adding rows for a batch buffer using the expression `generate_series(100, 1000, 50)` for distance parameter results in new rows with values 100, 150, 200, …. 1000.
- Ao definir um parâmetro de arquivo ou camada, mais opções são fornecidas:
  - *Add Files by Pattern…*: adds new rows to the table for files matching a *File pattern* in a folder to *Look in*. E.g. `*.shp` will add to the list all the `SHP` files in the folder. Check *Search recursively* to also browse sub-folders.
  - *Select Files…* individually on disk
  - *Add All Files from a Directory…*
  - *Select from Open Layers…* in the active project

O parâmetro de dados de saída expõe os mesmos recursos que ao executar o algoritmo como um único processo. Dependendo do algoritmo, a saída pode ser:

- ignorado, se a célula for deixada vazia
- saved as a temporary layer: fill the cell with your chosen output name, select *Create Temporary Layer* from the … drop-down, and remember to tick the *Load layers on completion* checkbox. For temporary layers, the value you provide will be used as the layer name. You can use the *Autofill* options to construct that name.
- saved as a plain file (`.SHP`, `.GPKG`, `.XML`, `.PDF`, `.JPG`,…): choose *Select File/Folder…* from the … drop-down. For plain files, you can set the path using the *Autofill* options exposed beforehand.

  E.g. use *Calculate by Expression…* to set output file names to complex expressions like:

```
'/home/me/stuff/buffer_' || left(@INPUT, 30) || '_' || @DISTANCE || '.shp'
```

  You can also type the file path directly or use the file chooser dialog that appears when clicking on the accompanying *Select File/Folder…* button. Once you select the file, a new dialog is shown to allow for auto-completion of other cells in the same column (same parameter).

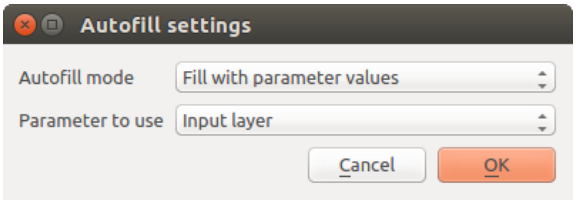


Figura23.34: Salvar Processamento em Lote

Se o valor padrão (*Não preencher automaticamente*) for selecionado, ele apenas colocará o nome de arquivo selecionado na célula selecionada da tabela de parâmetros. Se alguma das outras opções for selecionada, todas as células **abaixo** a selecionada será automaticamente preenchida com base em um critério definido:

- *Preencher com números*: acrescenta um número incrementalmente ao nome do arquivo
- guilabel:*Preencher com valores de parâmetro*: você pode selecionar um parâmetro cujo valor na mesma linha é anexado ao nome do arquivo. Isto é particularmente útil para nomear os objetos de dados de saída de acordo com os de entrada.

- salvo como uma camada em um contêiner de banco de dados:

```
# Indicate a layer within a GeoPackage file
ogr:dbname='C:/Path/To/Geopackage.gpkg' table="New_Table" (geom)

# Use the "Calculate By Expression" to output to different layers in a␣
↪GeoPackage
'ogr:dbname=\'' || @project_folder || '/Buffers.gpkg\' table="' || @INPUT || '_
↪' || @DISTANCE || '" (geom)'
```

23.6.4 Executando o processo em lote

To execute the batch process once you have introduced all the necessary values, just click on *Run*. The *Log* panel is activated and displays details and steps of the execution process. Progress of the global batch task will be shown in the progress bar in the lower part of the dialog.

23.7 Usando os algoritmos do processamento a partir do Terminal Python.

O terminal permite que usuários avançados aumente sua produtividade e execute operações complexas que não podem ser utilizados através dos outros elementos GUI da caixa de processamento. Modelos envolvendo vários algoritmos podem ser definidos usando a interface de linha de comando e operações adicionais como loops e sentenças condicionais podem ser adicionados para criar fluxos de trabalho mais flexíveis e poderosos.

There is not a processing console in QGIS, but all processing commands are available instead from the QGIS built-in *Python console*. That means that you can incorporate those commands into your console work and connect processing algorithms to all the other features (including methods from the QGIS API) available from there.

O código que pode executar a partir do Terminal Python, mesmo quando não chama nenhum método específico do processamento, pode ser convertido num novo algoritmo que pode mais tarde chamar da caixa de ferramentas, o modelador gráfico ou qualquer outro componente, como faz para outro algoritmo. De fato, alguns algoritmos que encontra na caixa de ferramentas são scripts simples.

Nesta seção, veremos como usar algoritmos de processamento do console QGIS Python e também como escrever algoritmos usando Python.

23.7.1 Chamando os algoritmos a partir do Terminal Python

A primeira coisa que tem de fazer é importar as funções do processamento com a seguinte linha:

```
>>> from qgis import processing
```

Now, there is basically just one (interesting) thing you can do with that from the console: execute an algorithm. That is done using the `run()` method, which takes the name of the algorithm to execute as its first parameter, and then a variable number of additional parameters depending on the requirements of the algorithm. So the first thing you need to know is the name of the algorithm to execute. That is not the name you see in the toolbox, but rather a unique command–line name. To find the right name for your algorithm, you can use the `processingRegistry`. Type the following line in your console:

```
>>> for alg in QgsApplication.processingRegistry().algorithms():
        print(alg.id(), "->", alg.displayName())
```

Você verá algo assim (com alguns traços extras adicionados para melhorar a legibilidade).

```
3d:tessellate --------------> Tessellate
gdal:aspect ----------------> Aspect
gdal:assignprojection ------> Assign projection
gdal:buffervectors ---------> Buffer vectors
gdal:buildvirtualraster ----> Build Virtual Raster
gdal:cliprasterbyextent ----> Clip raster by extent
gdal:cliprasterbymasklayer -> Clip raster by mask layer
gdal:clipvectorbyextent ----> Clip vector by extent
gdal:clipvectorbypolygon ---> Clip vector by mask layer
gdal:colorrelief -----------> Color relief
gdal:contour ---------------> Contour
gdal:convertformat ---------> Convert format
gdal:dissolve --------------> Dissolve
...
```

Essa é uma lista de todos os IDs de algoritmo disponíveis, classificados por nome de provedor e nome de algoritmo, juntamente com seus nomes correspondentes.

Once you know the command-line name of the algorithm, the next thing to do is to determine the right syntax to execute it. That means knowing which parameters are needed when calling the `run()` method.

There is a method to describe an algorithm in detail, which can be used to get a list of the parameters that an algorithm requires and the outputs that it will generate. To get this information, you can use the `algorithmHelp(id_of_the_algorithm)` method. Use the ID of the algorithm, not the full descriptive name.

Calling the method with `native:buffer` as parameter (`qgis:buffer` is an alias for `native:buffer` and will also work), you get the following description:

```
>>> processing.algorithmHelp("native:buffer")
Buffer (native:buffer)

This algorithm computes a buffer area for all the features in an
input layer, using a fixed or dynamic distance.

The segments parameter controls the number of line segments to
use to approximate a quarter circle when creating rounded
offsets.

The end cap style parameter controls how line endings are handled
in the buffer.

The join style parameter specifies whether round, miter or
beveled joins should be used when offsetting corners in a line.
```

(continua na próxima página)

(continuação da página anterior)

```
The miter limit parameter is only applicable for miter join
styles, and controls the maximum distance from the offset curve
to use when creating a mitered join.


----------------
Input parameters
----------------

INPUT: Input layer

   Parameter type: QgsProcessingParameterFeatureSource

   Accepted data types:
           - str: layer ID
           - str: layer name
           - str: layer source
           - QgsProcessingFeatureSourceDefinition
           - QgsProperty
           - QgsVectorLayer

DISTANCE: Distance

   Parameter type: QgsProcessingParameterDistance

   Accepted data types:
           - int
           - float
           - QgsProperty

SEGMENTS: Segments

   Parameter type: QgsProcessingParameterNumber

   Accepted data types:
           - int
           - float
           - QgsProperty

END_CAP_STYLE: End cap style

   Parameter type: QgsProcessingParameterEnum

   Available values:
           - 0: Round
           - 1: Flat
           - 2: Square

   Accepted data types:
           - int
           - str: as string representation of int, e.g. '1'
           - QgsProperty

JOIN_STYLE: Join style

   Parameter type: QgsProcessingParameterEnum

   Available values:
           - 0: Round
           - 1: Miter
           - 2: Bevel
```

(continua na próxima página)

(continuação da página anterior)

```
   Accepted data types:
            - int
            - str: as string representation of int, e.g. '1'
            - QgsProperty

MITER_LIMIT: Miter limit

   Parameter type: QgsProcessingParameterNumber

   Accepted data types:
            - int
            - float
            - QgsProperty

DISSOLVE: Dissolve result

   Parameter type: QgsProcessingParameterBoolean

   Accepted data types:
            - bool
            - int
            - str
            - QgsProperty

OUTPUT: Buffered

   Parameter type: QgsProcessingParameterFeatureSink

   Accepted data types:
            - str: destination vector file, e.g. 'd:/test.shp'
            - str: 'memory:' to store result in temporary memory layer
            - str: using vector provider ID prefix and destination URI,
                   e.g. 'postgres:...' to store result in PostgreSQL table
            - QgsProcessingOutputLayerDefinition
            - QgsProperty

----------------
Outputs
----------------

OUTPUT:  <QgsProcessingOutputVectorLayer>
   Buffered
```

Now you have everything you need to run any algorithm. As we have already mentioned, algorithms can be run using: `run()`. Its syntax is as follows:

```
>>> processing.run(name_of_the_algorithm, parameters)
```

Where parameters is a dictionary of parameters that depend on the algorithm you want to run, and is exactly the list that the `algorithmHelp()` method gives you.

```
 >>> processing.run("native:buffer", {'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
```

Se um parâmetro for opcional e você não quiser usá-lo, não o inclua no dicionário.

Se um parâmetro não for especificado, o valor padrão será usado.

Dependendo do tipo de parâmetro, os valores são introduzidos diferentemente. A próxima explicação é uma rápida revisão de como introduzir valores para cada tipo de parâmetro de entrada:

- Raster Layer, Vector Layer or Table. Simply use a string with the name that identifies the data object to use (the name it has in the QGIS Table of Contents) or a filename (if the corresponding layer is not opened, it will be opened but not added to the map canvas). If you have an instance of a QGIS object representing the layer, you can also pass it as parameter.
- Enumeration. If an algorithm has an enumeration parameter, the value of that parameter should be entered using an integer value. To know the available options, you can use the `algorithmHelp()` command, as above. For instance, the `native:buffer` algorithm has an enumeration called JOIN_STYLE:

```
JOIN_STYLE: Join style

   Parameter type: QgsProcessingParameterEnum

   Available values:
           - 0: Round
           - 1: Miter
           - 2: Bevel

   Accepted data types:
           - int
           - str: as string representation of int, e.g. '1'
           - QgsProperty
```

  Neste caso, o parâmetro tem três opções. Observe que a ordem é baseada em zero.

- Boleano. Use `Verdadeiro` ou `Falso`.
- Múltiplas entradas. O valor é uma cadeia de texto com descritores de entrada separados por ponto e vírgula (`;`). Como no caso das camadas ou tabelas únicas, cada descritor de entrada pode ser o nome do objeto ou o caminho do arquivo.
- Campo da Tabela de XXX. Use uma cadeia de texto com o nome do campo a usar. O parâmetro é caso sensitivo.
- Tabela Fixa. Digite a lista de todos os valores da tabela separadas por vírgulas (`,`) e feche entre aspas (`"`). Os valores começam na linha superior e vai da esquerda para a direita. Pode também usar uma matriz 2D de valores que representam a tabela.
- SRC. Introduza o número do código EPSG do SRC desejado.
- Extensão. Deve usar uma cadeia de texto com `xmin`, `xmax`, `ymin` e `ymax` valores separados por vírgulas (`,`).

Os parâmetros booleanos, de arquivo, cadeia de texto e numéricos não necessitam de explicações adicionais.

Parâmetros de entrada como strings, booleanos ou valores numéricos têm valores padrão. O valor padrão é usado se a entrada de parâmetro correspondente estiver ausente.

For output data objects, type the file path to be used to save it, just as it is done from the toolbox. If the output object is not specified, the result is saved to a temporary file (or skipped if it is an optional output). The extension of the file determines the file format. If you enter a file extension not supported by the algorithm, the default file format for that output type will be used, and its corresponding extension appended to the given file path.

Unlike when an algorithm is executed from the toolbox, outputs are not added to the map canvas if you execute that same algorithm from the Python console using the `run()` method. That method returns a dictionary with one or more output names (the ones shown in the algorithm description) as keys and the file paths of those outputs as values:

```
>>> myresult = processing.run("native:buffer", {'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
```

(continua na próxima página)

(continuação da página anterior)

```
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
>>> myresult['OUTPUT']
/data/buffers.shp
```

You can then load the output in the project as any common layer:

```
>>> buffered_layer = myresult['OUTPUT']
>>> QgsProject.instance().addMapLayer(buffered_layer)
```

To immediately load the processing outputs in the project, you can use the `runAndLoadResults()` method instead of `run()`.

```
>>> processing.runAndLoadResults("native:buffer", {parameters:values})
```

If you want to open an algorithm dialog from the console you can use the `createAlgorithmDialog` method. The only mandatory parameter is the algorithm name, but you can also define the dictionary of parameters so that the dialog will be filled automatically:

```
>>> my_dialog = processing.createAlgorithmDialog("native:buffer", {
                'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
>>> my_dialog.show()
```

The `execAlgorithmDialog` method opens the dialog immediately:

```
>>> processing.execAlgorithmDialog("native:buffer", {
                'INPUT': '/data/lines.shp',
                'DISTANCE': 100.0,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10,
                'OUTPUT': '/data/buffers.shp'})
```

23.7.2 Criando scripts e correndo-os a partir da caixa de ferramentas

You can create your own algorithms by writing Python code. Processing scripts extend `QgsProcessingAlgorithm`, so you need to add some extra lines of code to implement mandatory functions. You can find *Create new script* (clean sheet) and *Create New Script from Template* (template that includes code for mandatory functions of `QgsProcessingAlgorithm`) under the *Scripts* dropdown menu on the top of the Processing toolbox. The Processing Script Editor will open, and that's where you should type your code. Saving the script from there in the `scripts` folder (the default folder when you open the save file dialog) with a `.py` extension should create the corresponding algorithm.

O nome do algoritmo (o que você verá na caixa de ferramentas) é definido dentro do código.

Let's have a look at the following code, which defines a Processing algorithm that performs a buffer operation with a user defined buffer distance on a vector layer that is specified by the user, after first smoothing the layer.

```
from qgis.core import (QgsProcessingAlgorithm,
       QgsProcessingParameterNumber,
       QgsProcessingParameterFeatureSource,
       QgsProcessingParameterFeatureSink)

from qgis import processing

class algTest(QgsProcessingAlgorithm):
    INPUT_BUFFERDIST = 'BUFFERDIST'
    OUTPUT_BUFFER = 'OUTPUT_BUFFER'
    INPUT_VECTOR = 'INPUT_VECTOR'

    def __init__(self):
        super().__init__()

    def name(self):
        return "algTest"

    def displayName(self):
        return "algTest script"

    def createInstance(self):
        return type(self)()

    def initAlgorithm(self, config=None):
        self.addParameter(QgsProcessingParameterFeatureSource(
            self.INPUT_VECTOR, "Input vector"))
        self.addParameter(QgsProcessingParameterNumber(
            self.INPUT_BUFFERDIST, "Buffer distance",
            QgsProcessingParameterNumber.Double,
            100.0))
        self.addParameter(QgsProcessingParameterFeatureSink(
            self.OUTPUT_BUFFER, "Output buffer"))

    def processAlgorithm(self, parameters, context, feedback):
        #DO SOMETHING
        algresult = processing.run("native:smoothgeometry",
            {'INPUT': parameters[self.INPUT_VECTOR],
             'ITERATIONS':2,
             'OFFSET':0.25,
             'MAX_ANGLE':180,
             'OUTPUT': 'memory:'},
            context=context, feedback=feedback, is_child_algorithm=True)
        smoothed = algresult['OUTPUT']
        algresult = processing.run('native:buffer',
            {'INPUT': smoothed,
            'DISTANCE': parameters[self.INPUT_BUFFERDIST],
            'SEGMENTS': 5,
            'END_CAP_STYLE': 0,
            'JOIN_STYLE': 0,
            'MITER_LIMIT': 10,
            'DISSOLVE': True,
            'OUTPUT': parameters[self.OUTPUT_BUFFER]},
            context=context, feedback=feedback, is_child_algorithm=True)
        buffered = algresult['OUTPUT']
        return {self.OUTPUT_BUFFER: buffered}
```

After doing the necessary imports, the following `QgsProcessingAlgorithm` functions are specified:

- `name()`: The id of the algorithm (lowercase).
- `displayName()`: A human readable name for the algorithm.
- `createInstance()`: Create a new instance of the algorithm class.

- `initAlgorithm()`: Configure the parameterDefinitions and outputDefinitions.

  Here you describe the parameters and output of the algorithm. In this case, a feature source for the input, a feature sink for the result and a number for the buffer distance.

- `processAlgorithm()`: Do the work.

  Here we first run the `smoothgeometry` algorithm to smooth the geometry, and then we run the `buffer` algorithm on the smoothed output. To be able to run algorithms from within another algorithm we have to set the `is_child_algorithm` argument to `True`. You can see how input and output parameters are used as parameters to the `smoothgeometry` and `buffer` algorithms.

There are a number of different parameter types available for input and output. You can find the full list at *Tipos de entrada e saída para Algoritmos de Processamento*.

O primeiro parâmetro para os construtores é o nome do parâmetro e o segundo é a descrição do parâmetro (para a interface do usuário). O restante dos parâmetros do construtor são específicos do tipo de parâmetro.

The input can be turned into QGIS classes using the `parameterAs` functions of `QgsProcessingAlgorithm`. For instance to get the number provided for the buffer distance as a double:

```
self.parameterAsDouble(parameters, self.INPUT_BUFFERDIST, context)).
```

The `processAlgorithm` function should return a dictionary containing values for every output defined by the algorithm. This allows access to these outputs from other algorithms, including other algorithms contained within the same model.

Well behaved algorithms should define and return as many outputs as makes sense. Non-feature outputs, such as numbers and strings, are very useful when running your algorithm as part of a larger model, as these values can be used as input parameters for subsequent algorithms within the model. Consider adding numeric outputs for things like the number of features processed, the number of invalid features encountered, the number of features output, etc. The more outputs you return, the more useful your algorithm becomes!

Feedback

The `feedback` object passed to `processAlgorithm()` should be used for user feedback / interaction. You can use the `setProgress()` function of the `feedback` object to update the progress bar (0 to 100) to inform the user about the progress of the algorithm. This is very useful if your algorithm takes a long time to complete.

The `feedback` object provides an `isCanceled()` method that should be monitored to enable cancelation of the algorithm by the user. The `pushInfo()` method of `feedback` can be used to send information to the user, and `reportError()` is handy for pushing non-fatal errors to users.

Algorithms should avoid using other forms of providing feedback to users, such as print statements or logging to `QgsMessageLog`, and should always use the feedback object instead. This allows verbose logging for the algorithm, and is also thread-safe (which is important, given that algorithms are typically run in a background thread).

Handling errors

If your algorithm encounters an error which prevents it from executing, such as invalid input values or some other condition from which it cannot or should not recover, then you should raise a `QgsProcessingException`. E.g.:

```
if feature['value'] < 20:
  raise QgsProcessingException('Invalid input value {}, must be >= 20'.
↪format(feature['value']))
```

Try to avoid raising `QgsProcessingException` for non-fatal errors (e.g. when a feature has a null geometry), and instead just report these errors via `feedback.reportError()` and skip the feature. This helps make your algorithm "model-friendly", as it avoids halting the execution of an entire algorithm when a non-fatal error is encountered.

Documentando os seus scripts

As in the case of models, you can create additional documentation for your scripts, to explain what they do and how to use them.

`QgsProcessingAlgorithm` provides the `helpString()`, `shortHelpString()` and `helpUrl()` functions for that purpose. Specify / override these to provide more help to the user.

`shortDescription()` is used in the tooltip when hovering over the algorithm in the toolbox.

23.7.3 Pré- e pós-execução de encaixe da script

Scripts can also be used as pre- and post-execution hooks that are run before and after an algorithm is run, respectively. This can be used to automate tasks that should be performed whenever an algorithm is executed.

A sintaxe é identica à sintaxe explicada em cima, mas uma variável global `alg` está disponível, representando o algoritmo que foi (ou está prestes a ser) executado.

In the *General* group of the processing options dialog, you will find two entries named *Pre-execution script* and *Post-execution script* where the filenames of the scripts to be run in each case can be entered.

23.8 Usando o processamento da linha de comando

QGIS vem com uma ferramenta chamada "QGIS executor de processamento" permitindo que você execute algoritmos de processamento e modelos (construídos dentro do software ou providos por plugins), direto da linha de comando, sem precisar iniciar o aplicativo no desktop.

Por uma ferramenta de linha de comando, execute "qgis_process" e você deverá receber como saída:

```
QGIS Processing Executor - 3.35.0-Master 'Master' (3.35.0-Master)
Usage: C:\OSGeo4W\apps\qgis-dev\bin\qgis_process.exe [--help] [--version] [--json]␣
↪[--verbose] [--no-python] [--skip-loading-plugins] [command] [algorithm id, path␣
↪to model file, or path to Python script] [parameters]

Options:

  --help or -h            Output the help
  --version or -v         Output all versions related to QGIS Process
  --json                  Output results as JSON objects
  --verbose               Output verbose logs
  --no-python             Disable Python support (results in faster startup)
  --skip-loading-plugins  Avoid loading enabled plugins (results in faster startup)

Available commands:

  plugins            list available and active plugins
  plugins enable     enables an installed plugin. The plugin name must be␣
↪specified, e.g. "plugins enable cartography_tools"
  plugins disable    disables an installed plugin. The plugin name must be␣
↪specified, e.g. "plugins disable cartography_tools"
  list               list all available processing algorithms
  help               show help for an algorithm. The algorithm id or a path to a␣
↪model file must be specified.
  run                runs an algorithm. The algorithm id or a path to a model file␣
↪and parameter values must be specified.
                     Parameter values are specified after -- with PARAMETER=VALUE␣
↪syntax.
                     Ordered list values for a parameter can be created by␣
```

(continua na próxima página)

(continuação da página anterior)

```
↪specifying the parameter multiple times,
                    e.g. --LAYERS=layer1.shp --LAYERS=layer2.shp
                    Alternatively, a '-' character in place of the parameters␣
↪argument indicates that the parameters should be read from STDIN as a JSON␣
↪object.
                    The JSON should be structured as a map containing at least␣
↪the "inputs" key specifying a map of input parameter values.
                    This implies the --json option for output as a JSON object.
                    If required, the ellipsoid to use for distance and area␣
↪calculations can be specified via the "--ELLIPSOID=name" argument.
                    If required, an existing QGIS project to use during the␣
↪algorithm execution can be specified via the "--PROJECT_PATH=path" argument.
                    When passing parameters as a JSON object from STDIN, these␣
↪extra arguments can be provided as an "ellipsoid" and a "project_path" key␣
↪respectively.
```

Nota

Only installed plugins that advertise `hasProcessingProvider=yes` in their `metadata.txt` file are recognized and can be activated or loaded by qgis_process tool.

Dica

Before calling qgis_process on a system without window manager (e.g. a headless server), you should set:

```
export QT_QPA_PLATFORM=offscreen
```

O comando `lista` pode ser usado para obter uma lista de todos os provedores e algoritmos disponíveis.

```
qgis_process list
```

O comando `ajuda` pode ser usado para obter mais informações sobre comandos ou algoritmos.

```
qgis_process help qgis:regularpoints
```

O comando `executar` pode ser usado para executar um algoritmo ou modelo. Especifique o nome do algoritmo ou um caminho para um modelo como primeiro parâmetro.

```
qgis_process run native:buffer -- INPUT=source.shp DISTANCE=2 OUTPUT=buffered.shp
```

Onde um parâmetro aceita uma lista de valores, defina a mesma variável várias vezes.

```
qgis_process run native:mergevectorlayers -- LAYERS=input1.shp LAYERS=input2.shp␣
↪OUTPUT=merged.shp
```

While running an algorithm a text-based feedback bar is shown, and the operation can be cancelled via `CTRL+C`.

The `run` command also supports further parameters.

- `--json` irá formatar a saída stdout dentro de uma estrutura chamada JSON.
- –ellipsoid`` preparará o ellipsoid para o do tipo específico.
- `--distance_units` usará as unidades especificas de distância.
- `--area_units` usará as unidades especificas de área.
- `--project_path` irá carregar o projeto específico que executa o algoritmo

Complex input parameters, i.e. parameter types which are themselves specified as a dictionary type object for algorithms, are supported by qgis_process. To indicate that parameters will be specified via stdin, the qgis_process command must follow the format (with a trailing – in place of the usual arguments list).

```
qgis_process run algorithmId –
```

The JSON object must contain an "inputs" key, which is a map of the input parameter values. E.g.

```
echo "{'inputs': {'INPUT': 'my_shape.shp', 'DISTANCE': 5}}" | qgis_process run␣
↪native:buffer –
```

Additionally, extra settings like the distance units, area units, ellipsoid and project path can be included in this JSON object:

```
{
 'ellipsoid': 'EPSG:7019',
 'distance_units': 'feet',
 'area_units': 'ha',
 'project_path': 'C:/temp/my_project.qgs'
 'inputs': {'DISTANCE': 5, 'SEGMENTS': 8 ... }
}
```

Specifying input parameters via stdin implies automatically the `JSON` output format for results.

23.9 Escrevendo novos algoritmos de Processamento como scripts Python

Existem duas opções para escrever algoritmos de Processamento usando Python.

- *Extending* `QgsProcessingAlgorithm`
- *Using the @alg decorator*

Within QGIS, you can use *Create new script* in the *Scripts* menu at the top of the *Processing Toolbox* to open the *Processing Script Editor* where you can write your code. To simplify the task, you can start with a script template by using *Create new script from template* from the same menu. This opens a template that extends `QgsProcessingAlgorithm`.

If you save the script in the `scripts` folder (the default location) with a `.py` extension, the algorithm will become available in the *Processing Toolbox*.

23.9.1 Extending QgsProcessingAlgorithm

O seguinte código

1. recebe uma camada vetorial como entrada
2. conta o número de feições
3. faz uma operação de buffer
4. cria uma camada raster a partir do resultado da operação do buffer
5. retorna a camada de buffer, a camada raster e o número de feições

```
1 from qgis.PyQt.QtCore import QCoreApplication
2 from qgis.core import (QgsProcessing,
3                        QgsProcessingAlgorithm,
4                        QgsProcessingException,
```

(continua na próxima página)

(continuação da página anterior)

```
                       QgsProcessingOutputNumber,
                       QgsProcessingParameterDistance,
                       QgsProcessingParameterFeatureSource,
                       QgsProcessingParameterVectorDestination,
                       QgsProcessingParameterRasterDestination)
from qgis import processing


class ExampleProcessingAlgorithm(QgsProcessingAlgorithm):
    """
    This is an example algorithm that takes a vector layer,
    creates some new layers and returns some results.
    """

    def tr(self, string):
        """
        Returns a translatable string with the self.tr() function.
        """
        return QCoreApplication.translate('Processing', string)

    def createInstance(self):
        # Must return a new copy of your algorithm.
        return ExampleProcessingAlgorithm()

    def name(self):
        """
        Returns the unique algorithm name.
        """
        return 'bufferrasterextend'

    def displayName(self):
        """
        Returns the translated algorithm name.
        """
        return self.tr('Buffer and export to raster (extend)')

    def group(self):
        """
        Returns the name of the group this algorithm belongs to.
        """
        return self.tr('Example scripts')

    def groupId(self):
        """
        Returns the unique ID of the group this algorithm belongs
        to.
        """
        return 'examplescripts'

    def shortHelpString(self):
        """
        Returns a localised short help string for the algorithm.
        """
        return self.tr('Example algorithm short description')

    def initAlgorithm(self, config=None):
        """
        Here we define the inputs and outputs of the algorithm.
        """
        # 'INPUT' is the recommended name for the main input
        # parameter.
```

(continua na próxima página)

(continuação da página anterior)

```
        self.addParameter(
            QgsProcessingParameterFeatureSource(
                'INPUT',
                self.tr('Input vector layer'),
                types=[QgsProcessing.TypeVectorAnyGeometry]
            )
        )
        self.addParameter(
            QgsProcessingParameterVectorDestination(
                'BUFFER_OUTPUT',
                self.tr('Buffer output'),
            )
        )
        # 'OUTPUT' is the recommended name for the main output
        # parameter.
        self.addParameter(
            QgsProcessingParameterRasterDestination(
                'OUTPUT',
                self.tr('Raster output')
            )
        )
        self.addParameter(
            QgsProcessingParameterDistance(
                'BUFFERDIST',
                self.tr('BUFFERDIST'),
                defaultValue = 1.0,
                # Make distance units match the INPUT layer units:
                parentParameterName='INPUT'
            )
        )
        self.addParameter(
            QgsProcessingParameterDistance(
                'CELLSIZE',
                self.tr('CELLSIZE'),
                defaultValue = 10.0,
                parentParameterName='INPUT'
            )
        )
        self.addOutput(
            QgsProcessingOutputNumber(
                'NUMBEROFFEATURES',
                self.tr('Number of features processed')
            )
        )

    def processAlgorithm(self, parameters, context, feedback):
        """
        Here is where the processing itself takes place.
        """
        # First, we get the count of features from the INPUT layer.
        # This layer is defined as a QgsProcessingParameterFeatureSource
        # parameter, so it is retrieved by calling
        # self.parameterAsSource.
        input_featuresource = self.parameterAsSource(parameters,
                                                     'INPUT',
                                                     context)
        numfeatures = input_featuresource.featureCount()

        # Retrieve the buffer distance and raster cell size numeric
        # values. Since these are numeric values, they are retrieved
        # using self.parameterAsDouble.
```

(continua na próxima página)

(continuação da página anterior)

```
        bufferdist = self.parameterAsDouble(parameters, 'BUFFERDIST',
                                            context)
        rastercellsize = self.parameterAsDouble(parameters, 'CELLSIZE',
                                                context)
        if feedback.isCanceled():
            return {}
        buffer_result = processing.run(
            'native:buffer',
            {
                # Here we pass on the original parameter values of INPUT
                # and BUFFER_OUTPUT to the buffer algorithm.
                'INPUT': parameters['INPUT'],
                'OUTPUT': parameters['BUFFER_OUTPUT'],
                'DISTANCE': bufferdist,
                'SEGMENTS': 10,
                'DISSOLVE': True,
                'END_CAP_STYLE': 0,
                'JOIN_STYLE': 0,
                'MITER_LIMIT': 10
            },
            # Because the buffer algorithm is being run as a step in
            # another larger algorithm, the is_child_algorithm option
            # should be set to True
            is_child_algorithm=True,
            #
            # It's important to pass on the context and feedback objects to
            # child algorithms, so that they can properly give feedback to
            # users and handle cancelation requests.
            context=context,
            feedback=feedback)

        # Check for cancelation
        if feedback.isCanceled():
            return {}

        # Run the separate rasterization algorithm using the buffer result
        # as an input.
        rasterized_result = processing.run(
            'qgis:rasterize',
            {
                # Here we pass the 'OUTPUT' value from the buffer's result
                # dictionary off to the rasterize child algorithm.
                'LAYER': buffer_result['OUTPUT'],
                'EXTENT': buffer_result['OUTPUT'],
                'MAP_UNITS_PER_PIXEL': rastercellsize,
                # Use the original parameter value.
                'OUTPUT': parameters['OUTPUT']
            },
            is_child_algorithm=True,
            context=context,
            feedback=feedback)

        if feedback.isCanceled():
            return {}

        # Return the results
        return {'OUTPUT': rasterized_result['OUTPUT'],
                'BUFFER_OUTPUT': buffer_result['OUTPUT'],
                'NUMBEROFFEATURES': numfeatures}
```

Funções padrão do algoritmo de processamento:

- **createInstance (mandatory)**
  Deve retornar uma nova cópia do seu algoritmo. Se você alterar o nome da classe, certifique-se de atualizar também o valor retornado aqui para corresponder!
- **nome (obrigatório)**
  Retorna o nome exclusivo do algoritmo, usado para identificar o algoritmo.
- **displayName (mandatory)**
  Retorna o nome do algoritmo traduzido.
- **group**
  Retorna o nome do grupo ao qual este algoritmo pertence.
- **groupId**
  Retorna o ID único do grupo ao qual este algoritmo pertence.
- **shortHelpString**
  Returns a localised short help string for the algorithm.
- **initAlgorithm (mandatory)**
  Aqui definimos as entradas e saídas do algoritmo.

  `INPUT` and `OUTPUT` are recommended names for the main input and main output parameters, respectively.

  If a parameter depends on another parameter, `parentParameterName` is used to specify this relationship (could be the field / band of a layer or the distance units of a layer).
- **processAlgorithm (mandatory)**
  É onde ocorre o processamento.

  Parameters are retrieved using special purpose functions, for instance `parameterAsSource` and `parameterAsDouble`.

  `processing.run` can be used to run other processing algorithms from a processing algorithm. The first parameter is the name of the algorithm, the second is a dictionary of the parameters to the algorithm. `is_child_algorithm` is normally set to `True` when running an algorithm from within another algorithm. `context` and `feedback` inform the algorithm about the environment to run in and the channel for communicating with the user (catching cancel request, reporting progress, providing textual feedback). When using the (parent) algorithm's parameters as parameters to "child" algorithms, the original parameter values should be used (e.g. `parameters['OUTPUT']`).

  It is good practice to check the feedback object for cancelation as much as is sensibly possible! Doing so allows for responsive cancelation, instead of forcing users to wait for unwanted processing to occur.

  The algorithm should return values for all the output parameters it has defined as a dictionary. In this case, that's the buffer and rasterized output layers, and the count of features processed. The dictionary keys must match the original parameter/output names.

23.9.2 The @alg decorator

Using the @alg decorator, you can create your own algorithms by writing the Python code and adding a few extra lines to supply additional information needed to make it a proper Processing algorithm. This simplifies the creation of algorithms and the specification of inputs and outputs.

One important limitation with the decorator approach is that algorithms created in this way will always be added to a user's Processing Scripts provider – it is not possible to add these algorithms to a custom provider, e.g. for use in plugins.

The following code uses the @alg decorator to

1. usar uma camada vetorial como entrada
2. contar o número de feições
3. fazer uma operação de buffer

4. criar uma camada raster a partir do resultado da operação do buffer
5. retorna a camada de buffer, a camada raster e o número de feições

```
from qgis import processing
from qgis.processing import alg
from qgis.core import QgsProject

@alg(name='bufferrasteralg', label='Buffer and export to raster (alg)',
     group='examplescripts', group_label='Example scripts')
# 'INPUT' is the recommended name for the main input parameter
@alg.input(type=alg.SOURCE, name='INPUT', label='Input vector layer')
# 'OUTPUT' is the recommended name for the main output parameter
@alg.input(type=alg.RASTER_LAYER_DEST, name='OUTPUT',
           label='Raster output')
@alg.input(type=alg.VECTOR_LAYER_DEST, name='BUFFER_OUTPUT',
           label='Buffer output')
@alg.input(type=alg.DISTANCE, name='BUFFERDIST', label='BUFFER DISTANCE',
           default=1.0)
@alg.input(type=alg.DISTANCE, name='CELLSIZE', label='RASTER CELL SIZE',
           default=10.0)
@alg.output(type=alg.NUMBER, name='NUMBEROFFEATURES',
            label='Number of features processed')

def bufferrasteralg(instance, parameters, context, feedback, inputs):
    """
    Description of the algorithm.
    (If there is no comment here, you will get an error)
    """
    input_featuresource = instance.parameterAsSource(parameters,
                                                     'INPUT', context)
    numfeatures = input_featuresource.featureCount()
    bufferdist = instance.parameterAsDouble(parameters, 'BUFFERDIST',
                                            context)
    rastercellsize = instance.parameterAsDouble(parameters, 'CELLSIZE',
                                                context)
    if feedback.isCanceled():
        return {}
    buffer_result = processing.run('native:buffer',
                                   {'INPUT': parameters['INPUT'],
                                    'OUTPUT': parameters['BUFFER_OUTPUT'],
                                    'DISTANCE': bufferdist,
                                    'SEGMENTS': 10,
                                    'DISSOLVE': True,
                                    'END_CAP_STYLE': 0,
                                    'JOIN_STYLE': 0,
                                    'MITER_LIMIT': 10
                                    },
                                   is_child_algorithm=True,
                                   context=context,
                                   feedback=feedback)
    if feedback.isCanceled():
        return {}
    rasterized_result = processing.run('qgis:rasterize',
                                   {'LAYER': buffer_result['OUTPUT'],
                                    'EXTENT': buffer_result['OUTPUT'],
                                    'MAP_UNITS_PER_PIXEL': rastercellsize,
                                    'OUTPUT': parameters['OUTPUT']
                                   },
                                   is_child_algorithm=True, context=context,
                                   feedback=feedback)
    if feedback.isCanceled():
        return {}
```

(continua na próxima página)

(continuação da página anterior)

```
60      return {'OUTPUT': rasterized_result['OUTPUT'],
61              'BUFFER_OUTPUT': buffer_result['OUTPUT'],
62              'NUMBEROFFEATURES': numfeatures}
```

As you can see, it involves two algorithms ('native:buffer' and 'qgis:rasterize'). The last one ('qgis:rasterize') creates a raster layer from the buffer layer that was generated by the first one ('native:buffer').

The part of the code where this processing takes place is not difficult to understand if you have read the previous chapter. The first lines, however, need some additional explanation. They provide the information that is needed to turn your code into an algorithm that can be run from any of the GUI components, like the toolbox or the model designer.

These lines are all calls to the `@alg` decorator functions that help simplify the coding of the algorithm.

- The @alg decorator is used to define the name and location of the algorithm in the Toolbox.
- The @alg.input decorator is used to define the inputs of the algorithm.
- The @alg.output decorator is used to define the outputs of the algorithm.

For existing parameters and their correspondance, read *Tipos de entrada e saída para Algoritmos de Processamento*.

23.9.3 Handing algorithm output

Quando você declara uma saída representando uma camada (raster ou vetor), o algoritmo tentará adicioná-la ao QGIS assim que terminar.

- Raster layer output: QgsProcessingParameterRasterDestination / alg.RASTER_LAYER_DEST.
- Vector layer output: QgsProcessingParameterVectorDestination / alg.VECTOR_LAYER_DEST.

So even if the `processing.run()` method does not add the layers it creates to the user's current project, the two output layers (buffer and raster buffer) will be loaded, since they are saved to the destinations entered by the user (or to temporary destinations if the user does not specify destinations).

If a layer is created as output of an algorithm, it should be declared as such. Otherwise, you will not be able to properly use the algorithm in the modeler, since what is declared will not match what the algorithm really creates.

You can return strings, numbers and more by specifying them in the result dictionary (as demonstrated for "NUMBEROFFEATURES"), but they should always be explicitly defined as outputs from your algorithm. We encourage algorithms to output as many useful values as possible, since these can be valuable for use in later algorithms when your algorithm is used as part of a model.

23.9.4 Comunicação com o usuário

If your algorithm takes a long time to process, it is a good idea to inform the user about the progress. You can use `feedback` (`QgsProcessingFeedback`) for this.

The progress text and progressbar can be updated using two methods: `setProgressText(text)` and `setProgress(percent)`.

You can provide more information by using `pushCommandInfo(text)`, `pushDebugInfo(text)`, `pushInfo(text)` and `reportError(text)`.

If your script has a problem, the correct way of handling it is to raise a `QgsProcessingException`. You can pass a message as an argument to the constructor of the exception. Processing will take care of handling it and communicating with the user, depending on where the algorithm is being executed from (toolbox, modeler, Python console, ...)

23.9.5 Documentando os seus scripts

You can document your scripts by overloading the `helpString()` and `helpUrl()` methods of `QgsProcessingAlgorithm`.

23.9.6 Bandeiras

You can override the `flags()` method of `QgsProcessingAlgorithm` to tell QGIS more about your algorithm. You can for instance tell QGIS that the script shall be hidden from the modeler, that it can be canceled, that it is not thread safe, and more.

Dica

By default, Processing runs algorithms in a separate thread in order to keep QGIS responsive while the processing task runs. If your algorithm is regularly crashing, you are probably using API calls which are not safe to do in a background thread. Try returning the QgsProcessingAlgorithm.FlagNoThreading flag from your algorithm's flags() method to force Processing to run your algorithm in the main thread instead.

23.9.7 Melhores práticas para algoritmos de script escrito

Here's a quick summary of ideas to consider when creating your script algorithms and, especially, if you want to share them with other QGIS users. Following these simple rules will ensure consistency across the different Processing elements such as the toolbox, the modeler or the batch processing interface.

- Não coloque resultados das camadas. Vamos trabalhar o Processamento com seus resultados e carregar suas camadas se necessárias.
- Sempre declare as saídas que seu algoritmo cria.
- Do not show message boxes or use any GUI element from the script. If you want to communicate with the user, use the methods of the feedback object (`QgsProcessingFeedback`) or throw a `QgsProcessingException`.

There are already many processing algorithms available in QGIS. You can find code on the QGIS repo.

23.10 Tipos de entrada e saída para Algoritmos de Processamento

Here is the list of input and output types that are supported in Processing with their corresponding alg decorator constants, their name in the modeler designer if exposed.

23.10.1 Tipos de entrada

Input name	Classes	Alg constant	Descrição
Annotation layer	`QgsProcessingParameterAnnotationLayer`	`alg.ANNOTATION_LAYE`	Uma camada de texto
Area	`QgsProcessingParameterArea`		A numeric parameter representing an area measure

continua na próxima página

Tabela 23.1 – continuação da página anterior

Input name	Classes	Alg constant	Descrição
Configuração de Autenticação	QgsProcessingParameterAuthConfig	alg.AUTH_CFG	Permite que os usuários selecionem entre as configurações de autenticação disponíveis ou criem novas configurações de autenticação
Banda raster	QgsProcessingParameterBand	alg.BAND	Uma banda de uma camada raster
Booleano	QgsProcessingParameterBoolean	alg.BOOL	Um valor booleano
Connection Name	QgsProcessingParameterProviderConnection		A selection from available registered connections for a particular data provider
Cor	QgsProcessingParameterColor	alg.COLOR	Uma cor
Coordinate Operation	QgsProcessingParameterCoordinateOperation	alg.COORDINATE_OPER	Uma operação de coordenadas (para transformações SRC)
SRC	QgsProcessingParameterCrs	alg.CRS	Um Sistema de Referência de Coordenadas
Database Schema	QgsProcessingParameterDatabaseSchema	alg.DATABASE_SCHEMA	A database schema
Database Table	QgsProcessingParameterDatabaseTable	alg.DATABASE_TABLE	A database table
Data	ProcessingDateTimeParameterDate	alg.DATE	A pure date
Datetime	ProcessingDateTimeParameterDateTime	alg.DATETIME	A datetime
Distance	QgsProcessingParameterDistance	alg.DISTANCE	Um parâmetro numérico duplo para valores de distância
DXF Layers	QgsProcessingParameterDxfLayers		A list of vector layers to export as DXF file
Enum	QgsProcessingParameterEnum	alg.ENUM	Uma enumeração, permitindo a seleção de um conjunto de valores predefinidos
Expressão	QgsProcessingParameterExpression	alg.EXPRESSION	Uma expressão
Extensão	QgsProcessingParameterExtent	alg.EXTENT	Uma extensão espacial definida por xmin, xmax, ymin, ymax
	QgsProcessingParameterField	alg.FIELD	Um campo na tabela de atributos de uma camada vetorial
Field Aggregates	QgsProcessingParameterAggregate		A parameter for “aggregate” configurations, which consist of a definition of desired output fields, types, and aggregate used to populate them
Fields Mapper	QgsProcessingParameterFieldMapping		A parameter for “field mapping” configurations, which consist of a definition of desired output fields, types, and expressions used to populate them
File/Folder	QgsProcessingParameterFile	alg.FILE	Um nome de arquivo de um arquivo existente

continua na próxima página

Tabela 23.1 – continuação da página anterior

Input name	Classes	Alg constant	Descrição
	QgsProcessingParameterFileDestination	alg.FILE_DEST	Um nome de arquivo para um arquivo de saída recém-criado
	QgsProcessingParameterFolderDestination	alg.FOLDER_DEST	Uma pasta (pasta de destino)
Geometry	QgsProcessingParameterGeometry	alg.GEOMETRY	Uma geometria
	ProcessingNumberParInteger	alg.INT	Um número inteiro
Layout de impressão	QgsProcessingParameterLayout	alg.LAYOUT	Um layout
Print Layout Item	QgsProcessingParameterLayoutItem	alg.LAYOUT_ITEM	Um item de layout
Map Layer	QgsProcessingParameterMapLayer	alg.MAPLAYER	Uma camada do mapa
Map Theme	QgsProcessingParameterMapTheme	alg.MAP_THEME	A project map theme
Matriz	QgsProcessingParameterMatrix	alg.MATRIX	Uma matriz
Mesh Dataset Groups	QgsProcessing-ParameterMesh-DatasetGroups		Dataset groups of mesh layer
Mesh Dataset Time	QgsProcessing-ParameterMesh-DatasetTime		Dataset index from a mesh layer by time setting
Mesh Layer	QgsProcessingParameterMeshLayer	alg.MESH_LAYER	Uma camada de malha
Multiple Input	QgsProcessingParameterMultipleLayers	alg.MULTILAYER	A set of layers
Número	ProcessingNumberParDouble	alg.NUMBER	Um valor numérico
Ponto	QgsProcessingParameterPoint	alg.POINT	Um ponto
Point Cloud Attribute	QgsProcessingParameterPointCloudAttribute		An attribute from a point cloud layer
	QgsProcessing-ParameterPoint-CloudDestination	alg.POINTCLOUD_LAYE	A point cloud layer destination parameter, for specifying the destination path for a point cloud layer created by the algorithm
Point Cloud Layer	QgsProcessing-ParameterPoint-CloudLayer	alg.POINTCLOUD_LAYE	Uma camada de nuvens de pontos
	QgsProcessingParameterProvider-Connection	alg.PROVIDER_CONNEC	An available connection for a database provider
Intervalo	QgsProcessingParameterRange	alg.RANGE	Um intervalo de números
Camada Raster	QgsProcessing-ParameterRaster-Layer	alg.RASTER_LAYER	Uma camada raster

continua na próxima página

Tabela 23.1 – continuação da página anterior

Input name	Classes	Alg constant	Descrição
	QgsProcessingParameterRasterDestination	alg.RASTER_LAYER_DE	A raster layer destination parameter, for specifying the destination path for a raster layer created by the algorithm
Escala	QgsProcessingParameterScale	alg.SCALE	Uma escala do mapa
	QgsProcessingParameterFeatureSink	alg.SINK	A feature sink
String	QgsProcessingParameterString	alg.STRING	A text string
TIN Creation Layers	QgsProcessingParameterTinInputLayers		Selection of multiple layers to create a TIN with vertices and/or break lines
Hora	ProcessingDateTimeParameterTime	alg.TIME	A pure time
Vector Features	QgsProcessingParameterFeatureSource	alg.SOURCE	A feature source
Camada Vetorial	QgsProcessingParameterVectorLayer	alg.VECTOR_LAYER	Uma camada vetorial
	QgsProcessingParameterVectorDestination	alg.VECTOR_LAYER_DE	A vector layer destination parameter, for specifying the destination path for a vector layer created by the algorithm
Vector Tile Writer Layers	QgsProcessingParameterVectorTileDestination		A vector tile layer destination parameter, for specifying the destination path for a vector tile layer created by the algorithm
Volume	QgsProcessingParameterVolume		A numeric parameter representing a volume measure

23.10.2 Tipos de saída

Classes	Alg constant	Descrição
QgsProcessingOutputBoolean	alg.BOOL	Um valor booleano
QgsProcessingOutputNumber	alg.DISTANCE	Um parâmetro numérico duplo para valores de distância
QgsProcessingOutputFile	alg.FILE	Um nome de arquivo de um arquivo existente
QgsProcessingOutputFolder	alg.FOLDER	Uma pasta
QgsProcessingOutputHtml	alg.HTML	HTML
QgsProcessingOutputNumber	alg.INT	Um número inteiro
QgsProcessingOutputLayerDefinition	alg.LAYERDEF	Uma definição de camada
QgsProcessingOutputMapLayer	alg.MAPLAYER	Uma camada do mapa
QgsProcessingOutputMultipleLayers	alg.MULTILAYER	A set of layers
QgsProcessingOutputNumber	alg.NUMBER	Um valor numérico
QgsProcessingOutputPointCloudLayer	alg.POINTCLOUD_LAYER	Uma camada de nuvens de pontos
QgsProcessingOutputRasterLayer	alg.RASTER_LAYER	Uma camada raster

continua na próxima página

Tabela 23.2 – continuação da página anterior

Classes	Alg constant	Descrição
QgsProcessingOutputString	alg.STRING	A text string
QgsProcessingOutputVectorLayer	alg.VECTOR_LAYER	Uma camada vetorial
QgsProcessingOutputVectorTile-Layer		A vector tile layer

23.11 Configurando as aplicações externas

The Processing framework can be extended using additional applications. Algorithms that rely on external applications are managed by their own algorithm providers. Additional providers can be found as separate plugins, and installed using the QGIS Plugin Manager.

This section will show you how to configure the Processing framework to include these additional applications, and it will explain some particular features of the algorithms based on them. Once you have correctly configured the system, you will be able to execute external algorithms from any component like the toolbox or the model designer, just like you do with any other algorithm.

By default, algorithms that rely on an external application not shipped with QGIS are not enabled. You can enable them in the Processing settings dialog if they are installed on your system.

23.11.1 Uma nota para usuários Windows

If you are not an advanced user and you are running QGIS on Windows, you might not be interested in reading the rest of this chapter. Make sure you install QGIS in your system using the standalone installer. That will automatically install GRASS in your system and configure it so it can be run from QGIS. All the algorithms from this provider will be ready to be run without needing any further configuration. If installing with the OSGeo4W application, make sure that you also select GRASS for installation.

23.11.2 Uma nota para os formatos dos arquivos

When using external software, opening a file in QGIS does not mean that it can be opened and processed in that other software. In most cases, other software can read what you have opened in QGIS, but in some cases, that might not be true. When using databases or uncommon file formats, whether for raster or vector layers, problems might arise. If that happens, try to use well-known file formats that you are sure are understood by both programs, and check the console output (in the log panel) to find out what is going wrong.

You might for instance get trouble and not be able to complete your work if you call an external algorithm with a GRASS raster layers as input. For this reason, such layers will not appear as available to algorithms.

You should, however, not have problems with vector layers, since QGIS automatically converts from the original file format to one accepted by the external application before passing the layer to it. This adds extra processing time, which might be significant for large layers, so do not be surprised if it takes more time to process a layer from a DB connection than a layer from a Shapefile format dataset of similar size.

Providers not using external applications can process any layer that you can open in QGIS, since they open it for analysis through QGIS.

All raster and vector output formats produced by QGIS can be used as input layers. Some providers do not support certain formats, but all can export to common formats that can later be transformed by QGIS automatically. As for input layers, if a conversion is needed, that might increase the processing time.

23.11.3 Uma nota para as seleções da camada vetorial

External applications may also be made aware of the selections that exist in vector layers within QGIS. However, that requires rewriting all input vector layers, just as if they were originally in a format not supported by the external application. Only when no selection exists, or the *Use only selected features* option is not enabled in the processing general configuration, can a layer be directly passed to an external application.

Em outros casos, é necessário exportar apenas as feições selecionadas, o que causa tempos de execução mais longos.

23.11.4 Using third-party Providers

R scripts and libraries

To enable R in Processing you need to install the **Processing R Provider** plugin and configure R for QGIS.

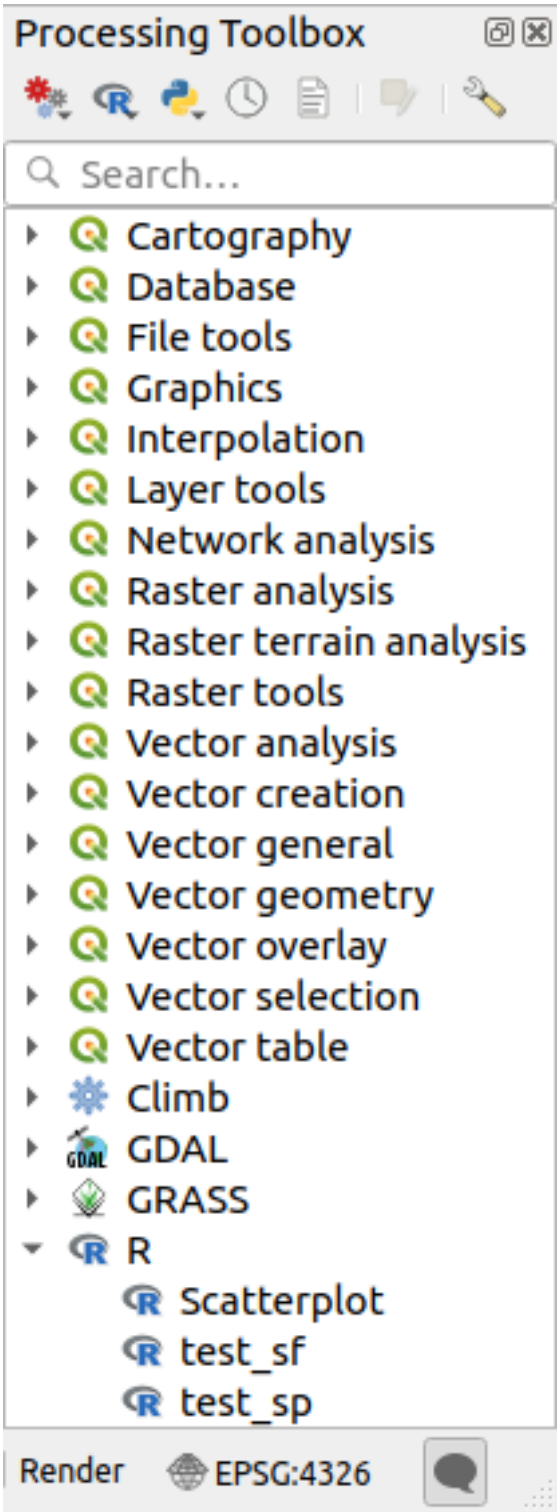
Configuration is done in *Provider* ► *R* in the *Processing* tab of *Settings* ► *Options*.

Depending on your operating system, you may have to use *R folder* to specify where your R binaries are located.

> **Nota**
>
> On **Windows** the R executable file is normally in a folder (`R-<version>`) under `C:\Program Files\R\`. Specify the folder and **NOT** the binary!
>
> On **Linux** you just have to make sure that the R folder is in the PATH environment variable. If `R` in a terminal window starts R, then you are ready to go.

After installing the **Processing R Provider** plugin, you will find some example scripts in the *Processing Toolbox*:

- *Scatterplot* runs an R function that produces a scatter plot from two numerical fields of the provided vector layer.
- *test_sf* does some operations that depend on the `sf` package and can be used to check if the R package `sf` is installed. If the package is not installed, R will try to install it (and all the packages it depends on) for you, using the *Package repository* specified in *Provider* ► *R* in the Processing options. The default is https://cran.r-project.org/. Installing may take some time…
- *test_sp* can be used to check if the R package `sp` is installed. If the package is not installed, R will try to install it for you.



Se você configurou o R corretamente para o QGIS, você poderá executar esses scripts.

Adicionando scripts R da coleção QGIS

R integration in QGIS is different from that of some other third party providers like SAGA in that there is not a predefined set of algorithms you can run (except for some example script that come with the *Processing R Provider* plugin).

A set of example R scripts is available in the QGIS Repository. Perform the following steps to load and enable them using the *QGIS Resource Sharing* plugin.

1. Add the *QGIS Resource Sharing* plugin (you may have to enable *Show also experimental plugins* in the Plugin Manager *Settings*)
2. Open it (*Plugins* ► *Resource Sharing* ► *Resource Sharing*)
3. Choose the *Settings* tab
4. Click *Reload repositories*
5. Choose the *All* tab
6. Select *QGIS R script collection* in the list and click on the *Install* button
7. The collection should now be listed in the *Installed* tab
8. Feche o plug-in
9. Open the *Processing Toolbox*, and if everything is OK, the example scripts will be present under R, in various groups (only some of the groups are expanded in the screenshot below).

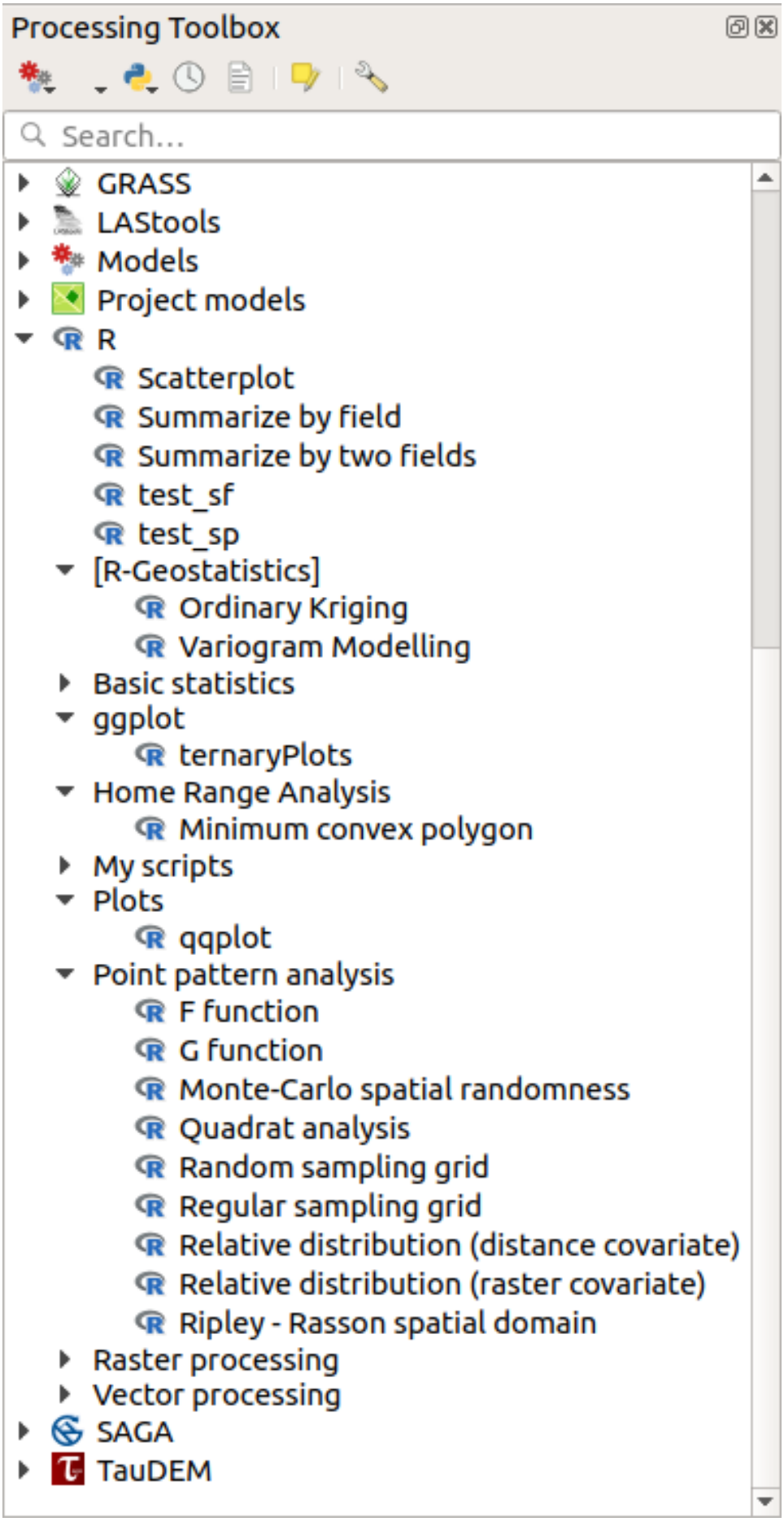
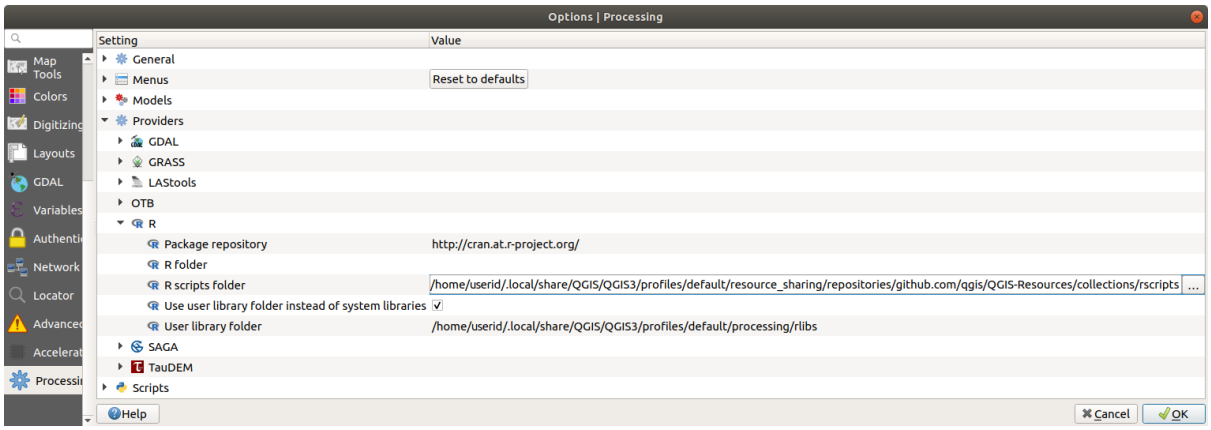


Figura23.35: The *Processing Toolbox* with some R scripts shown

The scripts at the top are the example scripts from the *Processing R Provider* plugin.

10. If, for some reason, the scripts are not available in the *Processing Toolbox*, you can try to:
    1. Open the Processing settings (*Settings* ► *Options* ► *Processing* tab)
    2. Go to *Providers* ► *R* ► *R scripts folder*
       - No Ubuntu, defina o caminho para (ou, melhor, inclua no caminho):

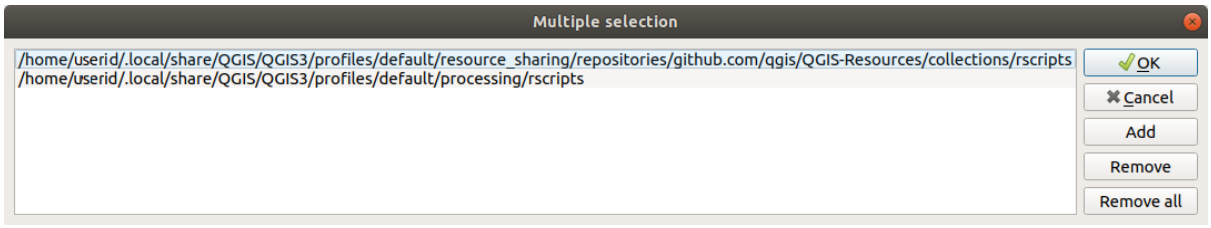
         /home/<user>/.local/share/QGIS/QGIS3/profiles/default/resource_sharing/repositories/github.com/qgis/QGIS-Resources/collections/rscripts



       - On Windows, set the path to (or, better, include in the path):

         C:\Users\<user>\AppData\Roaming\QGIS\QGIS3\profiles\default\resource_sharing\repositories\github.com\qgis\QGIS-Resources\collections\rscripts

    To edit, double-click. You can then choose to just paste / type the path, or you can navigate to the directory by using the … button and press the *Add* button in the dialog that opens. It is possible to provide several directories here. They will be separated by a semicolon (";").



If you would like to get all the R scrips from the QGIS 2 on-line collection, you can select *QGIS R script collection (from QGIS 2)* instead of *QGIS R script collection*. You will probably find that scripts that depend on vector data input or output will not work.

Criando scripts R

You can write scripts and call R commands, as you would do from R. This section shows you the syntax for using R commands in QGIS, and how to use QGIS objects (layers, tables) in them.

To add an algorithm that calls an R function (or a more complex R script that you have developed and you would like to have available from QGIS), you have to create a script file that performs the R commands.

R script files have the extension `.rsx`, and creating them is pretty easy if you just have a basic knowledge of R syntax and R scripting. They should be stored in the R scripts folder. You can specify the folder (*R scripts folder*) in the *R* settings group in Processing settings dialog).

Let's have a look at a very simple script file, which calls the R method `spsample` to create a random grid within the boundary of the polygons in a given polygon layer. This method belongs to the `maptools` package. Since almost all the algorithms that you might like to incorporate into QGIS will use or generate spatial data, knowledge of spatial packages like `maptools` and `sp`/`sf`, is very useful.

```
##Random points within layer extent=name
##Point pattern analysis=group
##Vector_layer=vector
##Number_of_points=number 10
##Output=output vector
library(sp)
spatpoly = as(Vector_layer, "Spatial")
pts=spsample(spatpoly,Number_of_points,type="random")
spdf=SpatialPointsDataFrame(pts, as.data.frame(pts))
Output=st_as_sf(spdf)
```

The first lines, which start with a double Python comment sign (`##`), define the display name and group of the script, and tell QGIS about its inputs and outputs.

Nota

To find out more about how to write your own R scripts, have a look at the R Intro section in the training manual and consult the *QGIS R Syntax* section.

Quando você declara um parâmetro de entrada, o QGIS usa essa informação para duas coisas: criar a interface do usuário para perguntar ao usuário o valor desse parâmetro e criar uma variável R correspondente que pode ser usada como entrada da função R.

In the above example, we have declared an input of type `vector`, named `Vector_layer`. When executing the algorithm, QGIS will open the layer selected by the user and store it in a variable named `Vector_layer`. So, the name of a parameter is the name of the variable that you use in R for accessing the value of that parameter (you should therefore avoid using reserved R words as parameter names).

Spatial parameters such as vector and raster layers are read using the `st_read()` (or `readOGR`) and `brick()` (or `readGDAL`) commands (you do not have to worry about adding those commands to your description file – QGIS will do it), and they are stored as `sf` (or `Spatial*DataFrame`) objects.

Os campos da tabela são armazenados como cadeiais contendo o nome do campo selecionado.

Vector files can be read using the `readOGR()` command instead of `st_read()` by specifying `##load_vector_using_rgdal`. This will produce a `Spatial*DataFrame` object instead of an `sf` object.

Raster files can be read using the `readGDAL()` command instead of `brick()` by specifying `##load_raster_using_rgdal`.

If you are an advanced user and do not want QGIS to create the object for the layer, you can use `##pass_filenames` to indicate that you prefer a string with the filename. In this case, it is up to you to open the file before performing any operation on the data it contains.

With the above information, it is possible to understand the first lines of the R script (the first line not starting with a Python comment character).

```
library(sp)
spatpoly = as(Vector_layer, "Spatial")
pts=spsample(polyg,numpoints,type="random")
```

The `spsample` function is provided by the *sp* library, so the first thing we do is to load that library. The variable `Vector_layer` contains an `sf` object. Since we are going to use a function (`spsample`) from the *sp* library, we must convert the `sf` object to a `SpatialPolygonsDataFrame` object using the `as` function.

Then we call the `spsample` function with this object and the `numpoints` input parameter (which specifies the number of points to generate).

Since we have declared a vector output named `Output`, we have to create a variable named `Output` containing an `sf` object.

We do this in two steps. First we create a `SpatialPolygonsDataFrame` object from the result of the function, using the *SpatialPointsDataFrame* function, and then we convert that object to an `sf` object using the `st_as_sf` function (of the *sf* library).

You can use whatever names you like for your intermediate variables. Just make sure that the variable storing your final result has the defined name (in this case `Output`), and that it contains a suitable value (an `sf` object for vector layer output).

In this case, the result obtained from the `spsample` method had to be converted explicitly into an `sf` object via a `SpatialPointsDataFrame` object, since it is itself an object of class `ppp`, which can not be returned to QGIS.

If your algorithm generates raster layers, the way they are saved will depend on whether or not you have used the `##dontuserasterpackage` option. If you have used it, layers are saved using the `writeGDAL()` method. If not, the `writeRaster()` method from the `raster` package will be used.

If you have used the `##pass_filenames` option, outputs are generated using the `raster` package (with `writeRaster()`).

If your algorithm does not generate a layer, but a text result in the console instead, you have to indicate that you want the console to be shown once the execution is finished. To do so, just start the command lines that produce the results you want to print with the > ('greater than') sign. Only output from lines prefixed with > are shown. For instance, here is the description file of an algorithm that performs a normality test on a given field (column) of the attributes of a vector layer:

```
##layer=vector
##field=field layer
##nortest=group
library(nortest)
>lillie.test(layer[[field]])
```

The output of the last line is printed, but the output of the first is not (and neither are the outputs from other command lines added automatically by QGIS).

If your algorithm creates any kind of graphics (using the `plot()` method), add the following line (`output_plots_to_html` used to be `showplots`):

```
##output_plots_to_html
```

Isso fará com que o QGIS redirecione todas as saídas gráficas do R para um arquivo temporário, que será aberto assim que a execução do R terminar.

Both graphics and console results will be available through the processing results manager.

For more information, please check the R scripts in the official QGIS collection (you download and install them using the *QGIS Resource Sharing* plugin, as explained in *Adicionando scripts R da coleção QGIS*). Most of them are rather simple and will greatly help you understand how to create your own scripts.

Nota

The `sf`, `rgdal` and `raster` libraries are loaded by default, so you do not have to add the corresponding `library()` commands. However, other libraries that you might need have to be explicitly loaded by typing: `library(ggplot2)` (to load the `ggplot2` library). If the package is not already installed on your machine, Processing will try to download and install it. In this way the package will also become available in R Standalone. **Be aware** that if the package has to be downloaded, the script may take a long time to run the first time.

R libraries installed when running sf_test

O script R `sp_test` tenta carregar os pacotes R `sp` e `raster`.

The R script *sf_test* tries to load `sf` and `raster`. If these two packages are not installed, R may try to load and install them (and all the libraries that they depend on).

The following R libraries end up in `~/.local/share/QGIS/QGIS3/profiles/default/processing/rscripts` after `sf_test` has been run from the Processing Toolbox on Ubuntu with version 2.0 of the *Processing R Provider* plugin and a fresh install of *R* 3.4.4 (*apt* package `r-base-core` only):

```
abind, askpass, assertthat, backports, base64enc, BH, bit, bit64, blob, brew,␣
↪callr, classInt, cli, colorspace, covr, crayon, crosstalk, curl, DBI, deldir,
desc, dichromat, digest, dplyr, e1071, ellipsis, evaluate, fansi, farver, fastmap,␣
↪gdtools, ggplot2, glue, goftest, gridExtra, gtable, highr, hms,
htmltools, htmlwidgets, httpuv, httr, jsonlite, knitr, labeling, later, lazyeval,␣
↪leafem, leaflet, leaflet.providers, leafpop, leafsync, lifecycle, lwgeom,
magrittr, maps, mapview, markdown, memoise, microbenchmark, mime, munsell, odbc,␣
↪openssl, pillar, pkgbuild, pkgconfig, pkgload, plogr, plyr, png, polyclip,
praise, prettyunits, processx, promises, ps, purrr, R6, raster, RColorBrewer, Rcpp,
↪ reshape2, rex, rgeos, rlang, rmarkdown, RPostgres, RPostgreSQL,
rprojroot, RSQLite, rstudioapi, satellite, scales, sf, shiny, sourcetools, sp,␣
↪spatstat, spatstat.data, spatstat.utils, stars, stringi, stringr, svglite,
sys, systemfonts, tensor, testthat, tibble, tidyselect, tinytex, units, utf8, uuid,
↪ vctrs, viridis, viridisLite, webshot, withr, xfun, XML, xtable
```

GRASS

Configuring GRASS is very easy. First, the path to the GRASS folder has to be defined, but only if you are running Windows.

By default, the Processing framework tries to configure its GRASS connector to use the GRASS distribution that ships along with QGIS. This should work without problems for most systems, but if you experience problems, you might have to configure the GRASS connector manually. Also, if you want to use a different GRASS installation, you can change the setting to point to the folder where the other version is installed. GRASS 7 is needed for algorithms to work correctly.

If you are running Linux, you just have to make sure that GRASS is correctly installed, and that it can be run without problem from a terminal window.

GRASS algorithms use a region for calculations. This region can be defined manually or automatically, taking the minimum extent that covers all the input layers used to execute the algorithm each time.

LAStools

To use LAStools in QGIS, you need to download and install LAStools on your computer and install the LAStools plugin (available from the official repository) in QGIS.

On Linux platforms, you will need Wine to be able to run some of the tools.

LAStools is activated and configured in the Processing options (*Settings* ► *Options*, *Processing* tab, *Providers* ► *LAStools*), where you can specify the location of LAStools (*LAStools folder*) and Wine (*Wine folder*). On Ubuntu, the default Wine folder is `/usr/bin`.

OTB Applications

OTB applications are fully supported within the QGIS Processing framework.

OTB (Orfeo ToolBox) is an image processing library for remote sensing data. It also provides applications that provide image processing functionalities. The list of applications and their documentation are available in OTB CookBook

Nota

Note that OTB is not distributed with QGIS and needs to be installed separately. Binary packages for OTB can be found on the download page.

To configure QGIS processing to find the OTB library:

1. Open the processing settings: *Settings* ► *Options* ► *Processing*
2. You can see `OTB` under *Providers* menu:
    1. Expand the *OTB* entry
    2. Set the *OTB folder*. This is the location of your OTB installation.
    3. Set the *OTB application folder*. This is the location of your OTB applications (`<PATH_TO_OTB_INSTALLATION>/lib/otb/applications`)
    4. Click *OK* to save the settings and close the dialog.

If settings are correct, OTB algorithms will be available in the *Processing Toolbox*.

Documentation of OTB settings available in QGIS Processing

- **OTB folder**: This is the directory where OTB is available.
- **OTB application folder**: This is the location(s) of OTB applications.

  São permitidos múltiplos caminhos.
- **Logger level** (optional): Level of logger to use by OTB applications.

  The level of logging controls the amount of detail printed during algorithm execution. Possible values for logger level are `INFO`, `WARNING`, `CRITICAL`, `DEBUG`. This value is `INFO` by default. This is an advanced user configuration.
- **Maximum RAM to use** (optional): by default, OTB applications use all available system RAM.

  You can, however, instruct OTB to use a specific amount of RAM (in MB) using this option. A value of 256 is ignored by the OTB processing provider. This is an advanced user configuration.
- **Geoid file** (optional): Path to the geoid file.

  This option sets the value of the elev.dem.geoid and elev.geoid parameters in OTB applications. Setting this value globally enables users to share it across multiple processing algorithms. Empty by default.
- **SRTM tiles folder** (optional): Directory where SRTM tiles are available.

  SRTM data can be stored locally to avoid downloading of files during processing. This option sets the value of elev.dem.path and elev.dem parameters in OTB applications. Setting this value globally enables users to share it across multiple processing algorithms. Empty by default.

Compatibility and Troubleshoot

Starting from OTB 6.6.1, new releases of OTB are made compatible with at least the latest QGIS version available at that time.

Se você tiver problemas com aplicações OTB no Processamento QGIS, por favor, abra um problema no OTB bug tracker, utilizando a etiqueta `qgis`.

Additional information about OTB and QGIS can be found in OTB Cookbook.

CAPÍTULO 24

Fornece algoritmos e processamento

Algoritmos de processamento e os seus parâmetros (tal como apresentados no interface do usuário) são documentados aqui.

24.1 Provedor de algorítmo QGIS

QGIS algorithm provider implements various analysis and geoprocessing operations using mostly only QGIS API. So almost all algorithms from this provider will work "out of the box" without any additional configuration.

Este provedor incorpora alguns algoritmos de plugins e também adiciona seus próprios algoritmos.

24.1.1 3D Tiles

Convert B3DM to GLTF

Converts files from the legacy `.B3DM` format to `.GLTF` or `.GLB`.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Input B3DM	`INPUT`	[3D Tile]	Input file to convert.
Arquivo de saída	`OUTPUT`	[3D Tile] Padrão: `[Salvar em arquivo temporário]`	Specify the output 3D tile file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo de saída	`OUTPUT`	[3D Tile]	Output layer in `.GLTF` format.

Código Python

Algorithm ID: `native:b3dmtogltf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Convert GLTF to vector features

Converts `.GLTF` or `.GLB` file contents to standard vector layer formats.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Input GLTF	`INPUT`	[3D Tile]	Input file to convert.
Output polygons Opcional	`OUTPUT_POLYGONS`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer to store polygon features. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Output lines Opcional	`OUTPUT_LINES`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output vector layer to store line features. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Output polygons	`OUTPUT_POLYGONS`	[vetor: polígono]	A vector layer storing the polygon features in input layer.
Output lines	`OUTPUT_LINES`	[vetor: linha]	A vector layer storing the line features in input layer.

Código Python

Algorithm ID: `native:gltftovector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.2 Cartografia

Alinhar pontos às feições

Calcula a rotação necessária para alinhar feições de ponto com a feição mais próxima de outra camada de referência. Um novo campo é adicionado à camada de saída que é preenchido com o ângulo (em graus, no sentido horário) para a feição de referência mais próxima.

Opcionalmente, a simbologia da camada de saída pode ser definida para usar automaticamente o campo de rotação calculado para girar os símbolos do marcador. Se desejado, uma distância máxima a ser usada ao alinhar pontos pode ser definida, para evitar o alinhamento de pontos isolados a feições distantes.

Dica

This algorithm is designed for use cases like aligning building point symbols to follow the nearest road direction.

permite features in-place modification 1 de pontos da característica.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Feições de ponto para calcular a rotação para
Camada de referência	`REFERENCE_LAYER`	[vector: geometry]	Camada para encontrar a feição mais próxima para cálculo de rotação
Distância máxima a considerar Opcional	`MAX_DISTANCE`	[numeric: double] Padrão: Não definido	Se nenhuma feição de referência for encontrada dentro dessa distância, nenhuma rotação será atribuída à feição de ponto.

continua na próxima página

Tabela 24.5 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Nome do campo de ângulo	`FIELD_NAME`	[string] Padrão: 'rotação'	Campo no qual armazenar o valor de rotação.
Aplicar simbologia automaticamente	`APPLY_SYMBOLOGY`	[boleano] Padrão: Verdadeiro	Rotaciona o marcador de símbolo das feições usando o valor de ângulo do campo
Camada alinhada	`OUTPUT`	[vetor: ponto] Padrão: `[Salvar em arquivo temporário]`	Specify the rotated output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada alinhada	`OUTPUT`	[vetor: ponto]	The point layer appended with a rotation field. If loaded to QGIS, it is applied by default the input layer symbology, with a data-defined rotation of its marker symbol.

Código Python

Algorithm ID: `native:angletonearest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Combinar bases de dados de estilo

Combines multiple QGIS style databases into a single style database. If items of the same type with the same name exist in different source databases these will be renamed to have unique names in the output combined database.

Ver também

Criar banco de dados de estilo do projeto

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Bancos de dados de entrada	`INPUT`	[arquivo] [lista]	Files containing QGIS style items
Objetos para combinar	`OBJECTS`	[enumeração] [lista]	Types of style items in the input databases you would like to put in the new database. These can be: • 0 — *Símbolos* • 1 — *Rampas de cores* • 2 — *Formatos de texto* • 3 — *Configurações de rótulo*
Output style database	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Output `.XML` file combining the selected style items. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Color ramp count	`COLORRAMPS`	[numeric: integer]	Number of color ramps
Label settings count	`LABELSETTINGS`	[numeric: integer]	Número de configurações de rótulos
Output style database	`OUTPUT`	[arquivo]	Output `.XML` file combining the selected style items
Contagem de símbolos	`SYMBOLS`	[numeric: integer]	Número de símbolos
Text format count	`TEXTFORMATS`	[numeric: integer]	Número de formatos de texto

Código Python

Algorithm ID: `native:combinestyles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar renderizador categorizado a partir de estilos

Sets a vector layer's renderer to a categorized renderer using matching symbols from a style database. If no style file is specified, symbols from the user's current *symbol library* are used instead.

A specified expression or field is used to create categories for the renderer. Each category is individually matched to the symbols which exist within the specified QGIS XML style database. Whenever a matching symbol name is found, the category's symbol will be set to this matched symbol.

Se desejado, as saídas também podem ser tabelas contendo listas das categorias que não puderam ser combinadas com símbolos, e símbolos que não foram combinadas com categorias.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial para aplicar um estilo categorizado a
Categorizar usando expressão	`FIELD`	[expressão]	Campo ou expressão para categorizar as feições
Estilo de banco de dados (deixar em branco para usar símbolos salvos)	`STYLE`	[arquivo]	File (`.XML`) containing the symbols to apply to the input layer categories. The file can be obtained from the Style Manager *Share symbols* tool. If no file is specified, QGIS local symbols library is used.
Usar correspondência sensível a maiúsculas e minúsculas para nomes de símbolos*	`CASE_SENSITIVE`	[boleano] Padrão: Falso	If True (checked), applies a case sensitive comparison between the categories and symbols names
Ignorar caracteres não alfanuméricos enquanto combinam	`TOLERANT`	[boleano] Padrão: Falso	Se Verdadeiro (marcado), os caracteres não alfanuméricos nos nomes das categorias e símbolos serão ignorados, permitindo maior tolerância durante a partida.
Categorias não correspondentes Opcional	`NON_MATCHING_C`	[vector: table] Default: `[Pular saída]`	Output table for categories which do not match any symbol in the database. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

continua na próxima página

Tabela 24.9 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Nomes de símbolos não correspondentes Opcional	NON_MATCHING_S	[vector: table] Default: [Pular saída]	Output table for symbols from the provided style database which do not match any category. *One of*: • Ignorar Saída • Criar Camada Temporária (TEMPORARY_OUTPUT) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Categorias não correspondentes	NON_MATCHING_C	[vector: table]	Lista as categorias que não podem corresponder a nenhum símbolo no banco de dados de estilos fornecido
Nomes de símbolos não correspondentes	NON_MATCHING_S	[vector: table]	Lista os símbolos do banco de dados de estilos fornecido que não podem corresponder a nenhuma categoria
Camada categorizada	OUTPUT	[o mesmo que entrada]	A camada de vetor de entrada com o estilo categorizado aplicado. Nenhuma nova camada é gerada.

Código Python

Algorithm ID: native:categorizeusingstyle

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar banco de dados de estilo do projeto

Extrai todos os objetos de estilo (símbolos, rampas de cores, formatos de texto e configurações de rótulos) de um projeto QGIS.

The extracted symbols are saved to a QGIS style database (XML format), which can be managed and imported via the *Style Manager* dialog.

Ver também

Combinar bases de dados de estilo

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Insira o projeto (deixe em branco para usar o atual) Opcional	`INPUT`	[arquivo]	Um arquivo de projeto QGIS para extrair os itens de estilo
Objetos para extrair	`OBJECTS`	[enumeração] [lista]	Tipos de padrões de itens no projeto de entrada que você gostaria de colocar no novo banco de dados. Estes podem ser: • 0 — *Símbolos* • 1 — *Rampas de cores* • 2 — *Formatos de texto* • 3 — *Configurações de rótulo*
Output style database	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Specify the output `.XML` file for the selected style items. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Color ramp count	`COLORRAMPS`	[numeric: integer]	Number of color ramps
Label settings count	`LABELSETTINGS`	[numeric: integer]	Número de configurações de rótulos
Output style database	`OUTPUT`	[arquivo]	Output `.XML` file for the selected style items
Contagem de símbolos	`SYMBOLS`	[numeric: integer]	Número de símbolos
Text format count	`TEXTFORMATS`	[numeric: integer]	Número de formatos de texto

Código Python

Algorithm ID: `native:stylefromproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exportar o layout do atlas como imagem

Exporta o atlas de um layout de impressão como arquivos de imagem (por exemplo, imagens PNG ou JPEG).

If a coverage layer is set, the selected layout's atlas settings exposed in this algorithm will be overwritten. In this case, an empty filter or sort by expression will turn those settings off.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Layout atlas	`LAYOUT`	[layout]	Layout para exportar
Camada de cobertura Opcional	`COVERAGE_LAYER`	[vector: geometry]	Camada a ser usada para gerar o atlas
Expressão do filtro	`FILTER_EXPRESSION`	[expressão]	Expressão a ser usada para filtrar as feições do atlas
Expressão ordenada Opcional	`SORTBY_EXPRESS`	[expressão]	Expression to use to sort the atlas features
Ordem de classificação reversa	`SORTBY_REVERSE`	[boleano] Padrão: Falso	Determines if sorting should be inverted. Used when a sort expression is provided.
Expressão do nome do arquivo de saída	`FILENAME_EXPRESSIO`	[expressão] Default: 'output_'||@atlas_featurer	Expressão para uso na geração de nomes de arquivos
Pasta de saída	`FOLDER`	[pasta]	Pasta de destino onde as imagens serão geradas

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Map layers to assign to unlocked map item(s) Opcional	`LAYERS`	[enumeração] [camada]	Layers to display in the map item(s) whose contents are not locked
Formato de imagem	`EXTENSION`	[enumeração] Padrão: png	File format of the generated output(s). The list of available formats varies depending on OS and installed drivers.
DPI Opcional	`DPI` Padrão: Não definido	[numeric: double]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Generate world file	`GEOREFERENCE`	[boleano] Padrão: Verdadeiro	Determina se um arquivo global deve ser gerado
Export RDF metadata	`INCLUDE_METADATA`	[boleano] Padrão: Verdadeiro	Determines if RDF metadata (title, author, …) should be generated
Enable antialiasing	`ANTIALIAS`	[boleano] Padrão: Verdadeiro	Determines if antialiasing should be enabled

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo de imagem	`OUTPUT`	[arquivo]	Image files generated by the atlas layout

Código Python

Algorithm ID: `native:atlaslayouttoimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Export atlas layout as PDF (multiple files)

Exports the atlas of a print layout to multiple PDF files.

If a coverage layer is set, the selected layout's atlas settings exposed in this algorithm will be overwritten. In this case, an empty filter or sort by expression will turn those settings off.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Layout atlas	`LAYOUT`	[layout]	Layout para exportar
Camada de cobertura Opcional	`COVERAGE_LAYER`	[vector: geometry]	Camada a ser usada para gerar o atlas
Expressão do filtro	`FILTER_EXPRESSION`	[expressão]	Expressão a ser usada para filtrar as feições do atlas
Expressão ordenada Opcional	`SORTBY_EXPRESS`	[expressão]	Expression to use to sort the atlas features
Ordem de classificação reversa	`SORTBY_REVERSE`	[boleano] Padrão: Falso	Determines if sorting should be inverted. Used when a sort expression is provided.
Output filename Opcional	`OUTPUT_FILENAME`	[expressão]	Name pattern of the PDF output files.
Pasta de saída	`OUTPUT_FOLDER`	[pasta]	Destination folder for the output PDF files.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Map layers to assign to unlocked map item(s) Opcional	`LAYERS`	[enumeração] [camada]	Layers to display in the map item(s) whose contents are not locked
DPI Opcional	`DPI` Padrão: Não definido	[numeric: double]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Sempre exportar como vetores	`FORCE_VECTOR`	[boleano] Padrão: Falso	Determina se os dados vetoriais devem ser deixados como vetores
Always export as raster	`FORCE_RASTER`	[boleano] Padrão: Falso	Forces all the items in the map to be rasterized. This parameter takes precedence over the `FORCE_VECTOR` parameter.
Anexar informações de georeferência	`GEOREFERENCE`	[boleano] Padrão: Verdadeiro	Determina se um arquivo global deve ser gerado
Export RDF metadata	`INCLUDE_METADATA`	[boleano] Padrão: Verdadeiro	Determines if RDF metadata (title, author, …) should be generated
Disable tiled raster layer exports	`DISABLE_TILED`	[boleano] Padrão: Falso	Determines if raster should be tiled
Simplificar as geometrias para reduzir o tamanho do arquivo de saída	`SIMPLIFY`	[boleano] Padrão: Verdadeiro	Determina se as geometrias devem ser simplificadas para reduzir o tamanho do arquivo de saída
Text export	`TEXT_FORMAT`	[enumeração] Padrão: 0	Determines if text should be exported as path or text objects. *Possible options* are: • 0 - Sempre exportar texto como caminhos (recomendado) • 1 - Sempre exportar textos como objetos de texto • 2 - Prefer exporting texts as text objects
Image compression	`IMAGE_COMPRESSIO`	[enumeração] Padrão: 0	Determines compression level of the image and how suitable the file could be for printing outputs or post-production in external applications. Possible options are: • 0 - Lossy (JPEG) • 1 - Lossless

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo PDF	`OUTPUT`	[arquivo]	PDF file corresponding to the exported atlas layout

Código Python

Algorithm ID: `native:atlaslayouttomultiplepdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Export atlas layout as PDF (single file)

Exports the atlas of a print layout as a single PDF file.

If a coverage layer is set, the selected layout's atlas settings exposed in this algorithm will be overwritten. In this case, an empty filter or sort by expression will turn those settings off.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Layout atlas	`LAYOUT`	[layout]	Layout para exportar
Camada de cobertura Opcional	`COVERAGE_LAYER`	[vector: geometry]	Camada a ser usada para gerar o atlas
Expressão do filtro	`FILTER_EXPRESSION`	[expressão]	Expressão a ser usada para filtrar as feições do atlas
Expressão ordenada Opcional	`SORTBY_EXPRESS`	[expressão]	Expression to use to sort the atlas features
Ordem de classificação reversa	`SORTBY_REVERSE`	[boleano] Padrão: Falso	Determines if sorting should be inverted. Used when a sort expression is provided.
Arquivo PDF	`OUTPUT`	[arquivo] Padrão: [Salvar em arquivo temporário]	Name (including path) of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Map layers to assign to unlocked map item(s) Opcional	`LAYERS`	[enumeração] [camada]	Layers to display in the map item(s) whose contents are not locked
DPI Opcional	`DPI` Padrão: Não definido	[numeric: double]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Sempre exportar como vetores	`FORCE_VECTOR`	[boleano] Padrão: Falso	Determina se os dados vetoriais devem ser deixados como vetores
Always export as raster	`FORCE_RASTER`	[boleano] Padrão: Falso	Forces all the items in the map to be rasterized. This parameter takes precedence over the `FORCE_VECTOR` parameter.
Anexar informações de georeferência	`GEOREFERENCE`	[boleano] Padrão: Verdadeiro	Determina se um arquivo global deve ser gerado
Export RDF metadata	`INCLUDE_METADATA`	[boleano] Padrão: Verdadeiro	Determines if RDF metadata (title, author, …) should be generated
Disable tiled raster layer exports	`DISABLE_TILED`	[boleano] Padrão: Falso	Determines if raster should be tiled
Simplificar as geometrias para reduzir o tamanho do arquivo de saída	`SIMPLIFY`	[boleano] Padrão: Verdadeiro	Determina se as geometrias devem ser simplificadas para reduzir o tamanho do arquivo de saída
Text export	`TEXT_FORMAT`	[enumeração] Padrão: 0	Determines if text should be exported as path or text objects. *Possible options* are: • 0 - Sempre exportar texto como caminhos (recomendado) • 1 - Sempre exportar textos como objetos de texto • 2 - Prefer exporting texts as text objects
Image compression	`IMAGE_COMPRESSIO`	[enumeração] Padrão: 0	Determines compression level of the image and how suitable the file could be for printing outputs or post-production in external applications. Possible options are: • 0 - Lossy (JPEG) • 1 - Lossless

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo PDF	`OUTPUT`	[arquivo]	PDF file corresponding to the exported atlas layout

Código Python

Algorithm ID: `native:atlaslayouttopdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exportar layout de impressão como imagem

Exporta um layout de impressão como um arquivo de imagem (por exemplo, imagens PNG ou JPEG)

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Print layout	`LAYOUT`	[layout]	Layout para exportar
Arquivo de imagem	`OUTPUT`	[arquivo] Padrão: [Salvar em arquivo temporário]	Name (including path) of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Map layers to assign to unlocked map item(s) Opcional	`LAYERS`	[enumeração] [camada]	Layers to display in the map item(s) whose contents are not locked
DPI Opcional	`DPI` Padrão: Não definido	[numeric: double]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Generate world file	`GEOREFERENCE`	[boleano] Padrão: Verdadeiro	Determina se um arquivo global deve ser gerado

continua na próxima página

Tabela 24.23 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Export RDF metadata	`INCLUDE_METADATA`	[boleano] Padrão: Verdadeiro	Determines if RDF metadata (title, author, …) should be generated
Enable antialiasing	`ANTIALIAS`	[boleano] Padrão: Verdadeiro	Determines if antialiasing should be enabled

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo de imagem	`OUTPUT`	[arquivo]	Image file corresponding to the exported print layout

Código Python

Algorithm ID: `native:printlayouttoimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exportar layout de impressão em PDF

Exports a print layout as a PDF file.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Print Layout	`LAYOUT`	[layout]	Layout para exportar
Arquivo PDF	`OUTPUT`	[arquivo] Padrão: [Salvar em arquivo temporário]	Name (including path) of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Map layers to assign to unlocked map item(s) Opcional	LAYERS	[enumeração] [camada]	Layers to display in the map item(s) whose contents are not locked
DPI Opcional	DPI Padrão: Não definido	[numeric: double]	DPI of the output file(s). If not set, the value in the print layout settings will be used.
Sempre exportar como vetores	FORCE_VECTOR	[boleano] Padrão: Falso	Determina se os dados vetoriais devem ser deixados como vetores
Always export as raster	FORCE_RASTER	[boleano] Padrão: Falso	Forces all the items in the map to be rasterized. This parameter takes precedence over the FORCE_VECTOR parameter.
Anexar informações de georeferência	GEOREFERENCE	[boleano] Padrão: Verdadeiro	Determina se um arquivo global deve ser gerado
Export RDF metadata	INCLUDE_METADATA	[boleano] Padrão: Verdadeiro	Determines if RDF metadata (title, author, …) should be generated
Disable tiled raster layer exports	DISABLE_TILED	[boleano] Padrão: Falso	Determines if raster should be tiled
Simplificar as geometrias para reduzir o tamanho do arquivo de saída	SIMPLIFY	[boleano] Padrão: Verdadeiro	Determina se as geometrias devem ser simplificadas para reduzir o tamanho do arquivo de saída
Text export	TEXT_FORMAT	[enumeração] Padrão: 0	Determines if text should be exported as path or text objects. *Possible options* are: • 0 - Sempre exportar texto como caminhos (recomendado) • 1 - Sempre exportar textos como objetos de texto • 2 - Prefer exporting texts as text objects
Image compression	IMAGE_COMPRESSIO	[enumeração] Padrão: 0	Determines compression level of the image and how suitable the file could be for printing outputs or post-production in external applications. Possible options are: • 0 - Lossy (JPEG) • 1 - Lossless
Exportar camadas como arquivos PDF separados	SEPARATE_LAYERS	[boleano] Padrão: Falso	If True, then a separate PDF file will be created per layer per map item in the layout. Additionally, separate PDF files may be created for other complex layout items, resulting in a set of PDF files which contain logical atomic components of the layout.

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo PDF	`OUTPUT`	[arquivo]	PDF file(s) corresponding to the exported print layout

Código Python

Algorithm ID: `native:printlayouttopdf`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extract labels

Extracts label information from a rendered map at a given extent and scale.

If a map theme is provided, the rendered map will match the visibility and symbology of that theme. If left blank, all visible layers from the project will be used. Extracted label information include: position (served as point geometries), the associated layer name and feature ID, label text, rotation (in degree, clockwise), multiline alignment, and font details.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Map extent	`EXTENT`	[extensão]	Extent of the map to extract the labels from Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Escala do mapa	`SCALE`	[escala]	Extracted labels will be rendered using their properties set at this scale.
Map theme Opcional	`MAP_THEME`	[maptheme]	A map theme displaying the layers to extract the labels from. If unset, labels of the currently visible layers are extracted.
Include unplaced labels	`INCLUDE_UNPLACED`	[boleano] Padrão: Verdadeiro	Specify whether all overlapping labels should be extracted, including the conflicting (thus unplaced) ones.
Extracted labels	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the extent(s). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Map resolution (in DPI) Opcional	`DPI` Default: 96.0	[numeric: double]	

Saídas

Etiqueta	Nome	Tipo	Descrição
Extracted labels	`OUTPUT`	[vetor: ponto]	Point vector layer representing the fetched labels. Each feature has attributes identifying its source (layer, feature ID) and the assigned labeling properties (text, font, size, rotation, …). A default style with labeling and null symbol is also applied to the layer. **Aviso** Because some of the generated fields have name with more than 10 characters, using the ESRI shapefile format (`.SHP`) to store the output may lead to unexpected rendering while loading the layer in QGIS.

Código Python

Algorithm ID: `native:extractlabels`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Print layout map extent to layer

Creates a polygon layer containing the extent of a print layout map item (or items), with attributes specifying the map size (in layout units, i.e. the *reference map* units), scale and rotation.

If the map item parameter is specified, then only the matching map extent will be exported. If it is not specified, all map extents from the layout will be exported.

Opcionalmente, um SRC de saída específico pode ser especificado. Se não for especificado, o item original do mapa SRC será usado.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Print layout	`LAYOUT`	[enumeração]	Um layout de impressão no projeto atual

continua na próxima página

Tabela 24.31 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Map item Opcional	`MAP`	[enumeração] Padrão: *Todos os itens do mapa*	O(s) item(ns) do mapa cujas informações você deseja extrair. Se nenhum for fornecido, todos os itens do mapa serão processados.
Extensão	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the extent(s). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Substituir SRC Opcional	`CRS`	[src] Padrão: *O layout SRC*	Selecione o SRC para a camada na qual as informações serão relatadas.

Saídas

Etiqueta	Nome	Tipo	Descrição
Altura do mapa	`HEIGHT`	[numeric: double]	
Extensão	`OUTPUT`	[vetor: polígono]	Camada vetorial de polígono de saída contendo extensões de todos os item(ns) do mapa de layout de entrada
Rotação do mapa	`ROTATION`	[numeric: double]	
Escala do mapa	`SCALE`	[numeric: double]	
Largura do mapa	`WIDTH`	[numeric: double]	

Código Python

Algorithm ID: `native:printlayoutmapextenttolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Definir estilo de camada

Aplica um estilo fornecido a uma camada. O estilo deve ser definido em um arquivo `QML`.

Nenhuma nova saída é criada: o estilo é imediatamente atribuído à camada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[camada]	Camada de entrada à qual você deseja aplicar o estilo
Arquivo de estilo	`STYLE`	[arquivo]	Caminho para o arquivo `.qml` do estilo

Saídas

Etiqueta	Nome	Tipo	Descrição
	`OUTPUT`	[o mesmo que entrada]	A camada de entrada com o novo estilo atribuído. Nenhuma nova camada é criada.

Código Python

Algorithm ID: `native:setlayerstyle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Coloração topológica

Atribui um índice de cores às feições do polígono de forma que nenhum polígono adjacente compartilhe o mesmo índice de cores, minimizando o número de cores necessárias.

O algoritmo permite a escolha do método a ser usado ao atribuir cores.

Um número mínimo de cores pode ser especificado, se desejado. O índice de cores é salvo em um novo atributo chamado **cor_id**.

O exemplo a seguir mostra o algoritmo com quatro cores diferentes escolhidas; como você pode ver, cada classe de cor tem a mesma quantidade de feições.

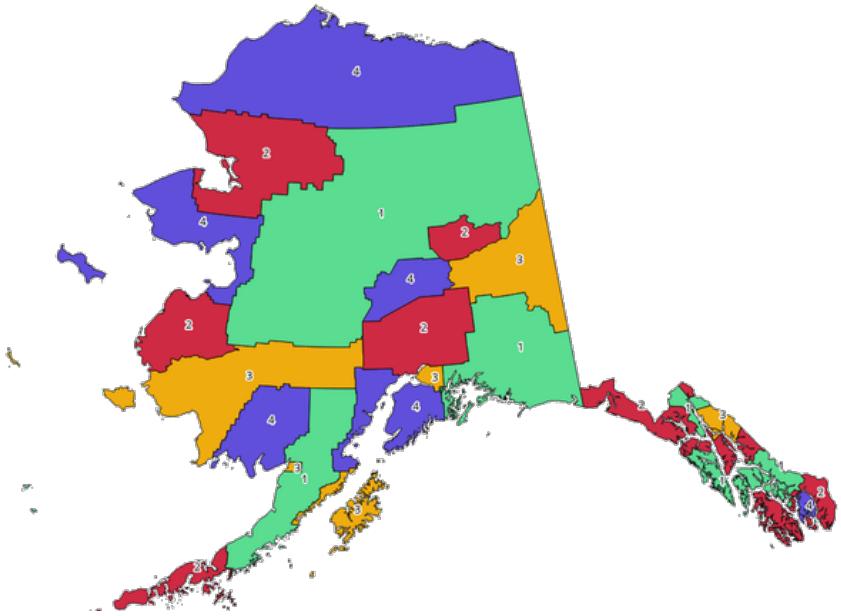


Figura24.1: Exemplo de cores topológicas

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	A camada de polígono de entrada
Número mínimo de cores	`MIN_COLORS`	[numeric: integer] Padrão: 4	O número mínimo de cores a serem atribuídas. Mínimo 1, máximo 1000.
Distância mínima entre feições	`MIN_DISTANCE`	[numeric: double] Padrão: 0.0	Evite que feições próximas (mas não tocantes) recebam cores iguais. Mínimo 0,0.
Atribuição de cores de equilíbrio	`BALANCE`	[enumeração] Padrão: 0	As opções são: • 0 — Por contagem de feiçãp Tenta atribuir cores para que a contagem de recursos atribuídos a cada índice de cores individual seja equilibrada. • 1 — Por área designada Atribui cores para que a área total de feições atribuídas a cada cor seja equilibrada. Este modo pode ser útil para ajudar a evitar grandes feições, resultando em uma das cores aparecendo mais dominante em um mapa colorido. • 2 — Por distância entre as cores Atribui cores para maximizar a distância entre as feições da mesma cor. Este modo ajuda a criar uma distribuição de cores mais uniforme em um mapa.

continua na próxima página

Tabela 24.36 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Colorido	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Colorido	`OUTPUT`	[vetor: polígono]	Polygon vector layer with an added `color_id` column

Código Python

Algorithm ID: `qgis:topologicalcoloring`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Transferência de anotações da camada principal

Transfers all *annotations* from the main annotation layer in a project to a new annotation layer. Items placement can then be adjusted within the layer stack.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Novo nome de camada	`LAYER_NAME`	[string] Padrão: ‘Anotações’	Nome da camada de anotações a ser criada

Saídas

Etiqueta	Nome	Tipo	Descrição
Novo nome de camada	`OUTPUT`	[camada]	Uma camada com itens da camada principal de anotação

Código Python

Algorithm ID: `native:transferannotationsfrommain`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.3 Check Geometry

Dangle-end lines

Detects dangle-end lines in line geometries and reports them as errors. A dangle-end line is a line feature that terminates at a vertex connected to only one segment, resulting in an endpoint without a proper connection.

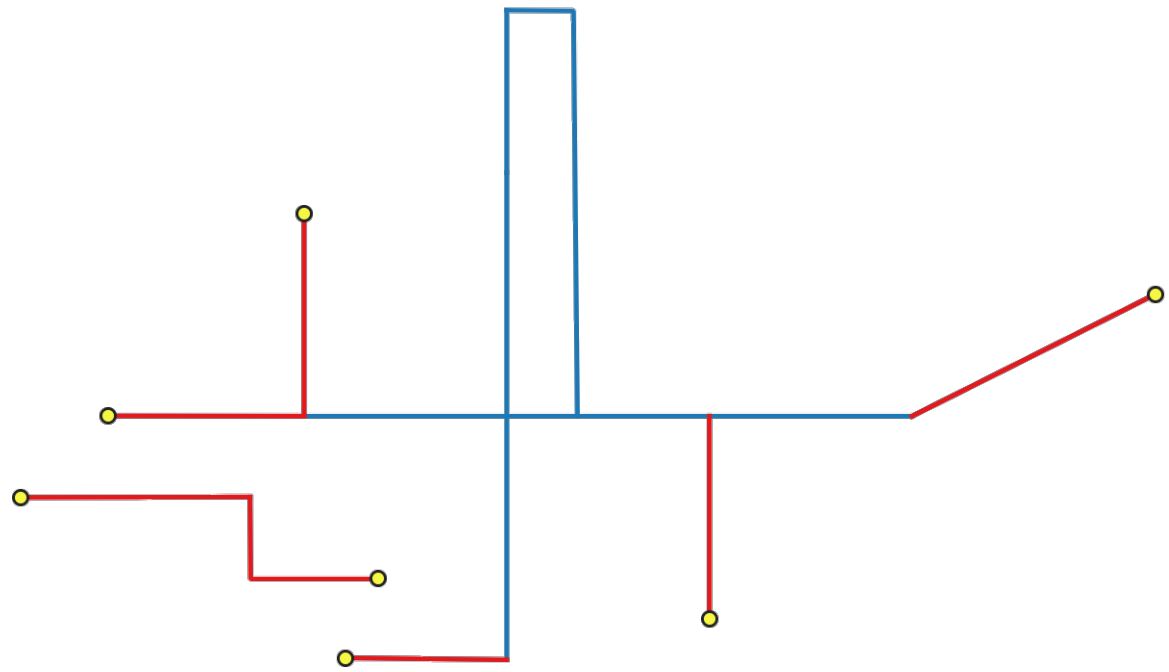


Figura24.2: Identifying dangle-end features (in red) and vertices (yellow).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Dangle-end errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Dangle-end features Optional	`OUTPUT`	[vector: line] Default: `[Skip output]`	Line layer containing the identified dangle-end line features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Dangle-end errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the dangle-end. • `gc_ringidx`: the index of the feature's geometry ring containing the dangle-end. • `gc_vertidx`: the vertex index of the dangle-end in the feature's geometry ring. • `gc_errorx`: the x coordinate of the dangle-end. • `gc_errory`: the y coordinate of the dangle-end. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature with the dangle-end.
Dangle-end features	`OUTPUT`	[vector: line]	Output line layer containing the input dangle-end features. If no dangle-end features are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrydangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Degenerate polygons

Checks the polygons with less than 3 points, which are degenerate polygons. Degenerate polygons are errors.

Nota

This algorithm detects invalid polygon geometries (degenerate polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Degenerate polygons errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the centroid of the degenerate polygons. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Degenerate polygons features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Specification of the output layer containing the degenerate polygons. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Degenerate polygons errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the geometry part of the degenerate polygon. • gc_ringidx: the index of the ring of the degenerate polygon. • gc_vertidx • gc_errorx: the x coordinate of the centroid of the degenerate polygon • gc_errory: the y coordinate of the centroid of the degenerate polygon • gc_error • UNIQUE_ID field: the unique ID of the input feature that is degenerate.
Degenerate polygons features	OUTPUT	[vector: polygon]	Output polygon layer with features containing the degenerate polygons. If no degenerate polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrydegeneratepolygon

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Duplicated geometries

Checks for duplicated geometries in a vector layer, and reports them as errors.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification
Duplicate geometries errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Duplicate geometries Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer containing the geometries that were found to be duplicated. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Duplicate geometries errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the duplicate geometry. • gc_errory: the y coordinate of the centroid of the duplicate geometry. • gc_error: the indices of the duplicate geometry (all except the highest index). • UNIQUE_ID field: the unique ID of the input feature that is a duplicate.
Duplicate geometries	OUTPUT	[vector: same as input]	Output layer containing the geometries that were found to be duplicated. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryduplicate

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Duplicated vertices

Checks for duplicated vertices in line or polygon geometries, and reports them as errors.

Ver também

Delete duplicated vertices, *Remover vértices duplicados*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification
Duplicated vertices errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Duplicated vertices features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer containing the features with duplicated vertices. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Duplicated vertices errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part containing the duplicate vertex. • gc_ringidx: the index of the feature's geometry ring containing the duplicate vertex. • gc_vertidx: the index of the duplicate vertex in the feature's geometry ring. • gc_errorx: the x coordinate of the duplicate vertex. • gc_errory: the y coordinate of the duplicate vertex. • gc_error • UNIQUE_ID field: the unique ID of the input feature that has duplicate vertices.
Duplicated vertices features	OUTPUT	[vector: same as input]	Output layer containing the features with duplicated vertices. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryduplicatenodes

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Features inside polygon

Checks the input geometries contained in the polygons from the polygon layers list. A polygon layer can be checked against itself. Any contained features are reported as errors.

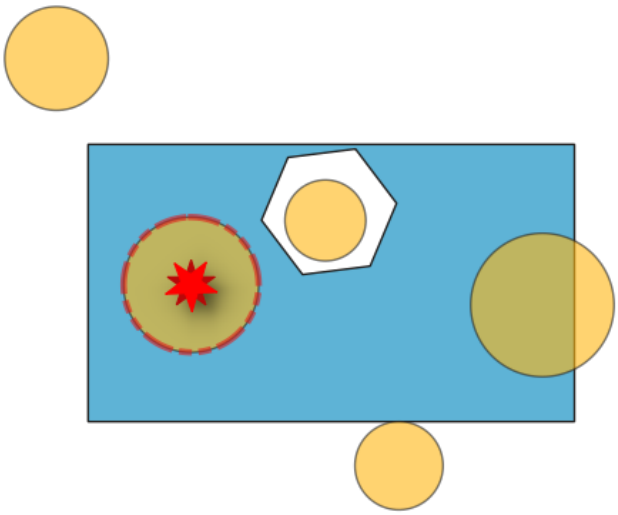


Figura24.3: Reporting errors on features inside a polygon.

Ver também

Delete features

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Polygon layers	`POLYGONS`	[vector: polygon][list]	List of polygon layers to check against.
Errors from contained features	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Point layer representing the contained features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continua na próxima página

Tabela 24.52 – continuação da página anterior

Label	Name	Type	Description
Contained features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer containing features within the checked polygons. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Errors from contained features	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the contained feature. • `gc_errory`: the y coordinate of the centroid of the contained feature. • `gc_error`: the layer name and feature ID of a polygon that contains the feature. • `UNIQUE_ID` field: the unique ID of the input feature that is contained within another.
Contained features	`OUTPUT`	[vector: same as input]	Output layer containing features within the checked polygons. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrycontained`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Holes

Detects holes in polygon geometries and reports them as errors.

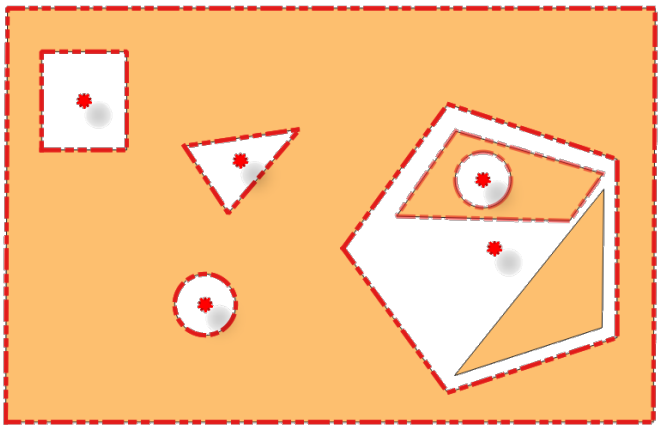


Figura24.4: Reporting errors for holes on a polygon feature.

Ver também

Fill holes

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification

continua na próxima página

Tabela 24.55 – continuação da página anterior

Label	Name	Type	Description
Holes errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Polygons with holes Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Polygon layer with the features containing holes. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Holes errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the hole. • `gc_ringidx`: the index of the feature's geometry ring containing the hole. • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the hole. • `gc_errory`: the y coordinate of the centroid of the hole. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that has a hole.
Polygon with holes	`OUTPUT`	[vector: polygon]	Output polygon layer with features containing holes. If no holes are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometryhole`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Lines intersecting each other

Checks intersections between line geometries within a layer. Intersections between two different lines are errors.

Ver também

Interseções de linha, *Interseção*, *Self-intersections*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Intersection errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the intersections location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Intersecting feature Optional	`OUTPUT`	[vector: line] Default: `[Skip output]`	Specification of the output layer for features intersecting each other. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Intersection errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part where the intersection occurs. • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the intersection point. • `gc_errory`: the y coordinate of the intersection point. • `UNIQUE_ID` field: the unique ID of an intersecting input feature. • `gc_error`: the layer name and index of the other intersecting feature.
Intersecting feature	`OUTPUT`	[vector: line]	Output layer containing, for each identified intersection, the feature (with same `UNIQUE_ID` value) it belongs to. If no intersections are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrylineintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Lines intersecting other layer

Checks if the input line layer features intersect with the check layer features. An input feature that intersects with a check layer feature is an error.

Ver também

Interseções de linha, *Interseção*, *Self-intersections*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Check layer	`CHECK_LAYER`	[vector: line, polygon]	Layer with the geometries to check against.
Line intersecting other layer errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output point layer containing the locations of intersection errors. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Line intersecting other layer features Optional	`OUTPUT`	[line] Default: `[Skip output]`	Line layer with the input features that intersect features in the check layer. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Line intersecting other layer errors	`ERRORS`	[vector: point]	Output point layer representing the error location and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part where the intersection occurs. • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the intersection point. • `gc_errory`: the y coordinate of the intersection point. • `UNIQUE_ID` field: the unique ID of the intersecting input feature. • `gc_error`: the index of the check layer where the intersection occurs.
Line intersecting other layer features	`OUTPUT`	[vector: line]	Output layer containing, for each identified intersection, the input feature it belongs to. If no intersections are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrylinelayerintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Missing vertices along borders

Checks for missing vertices along polygon borders. To be topologically correct, a vertex at the junction of two polygons must be present on both polygons. Missing vertices are errors.

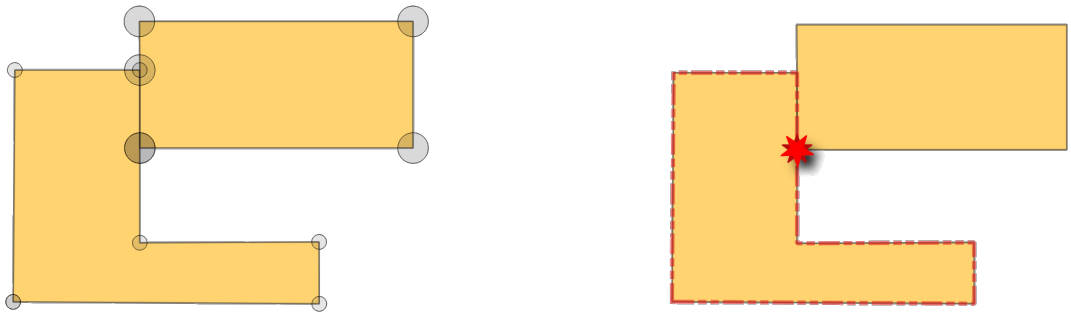


Figura24.5: Reporting errors for missing vertices on polygon features.

Ver também

Add missing vertices along borders

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Missing vertices errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Missing vertices features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Polygon layer with the features whose vertices are missing. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Missing vertices errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the missing vertex. • `gc_errory`: the y coordinate of the missing vertex. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that has a missing vertex.
Missing vertices features	`OUTPUT`	[vector: polygon]	Output polygon layer with features whose vertices are missing. There will be as many (duplicate) features as there are missing vertices in each geometry. If no vertices are missing, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrymissingvertex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Overlaps

Calculates overlapping areas in polygon geometries, and reports areas smaller than a given minimum as errors.

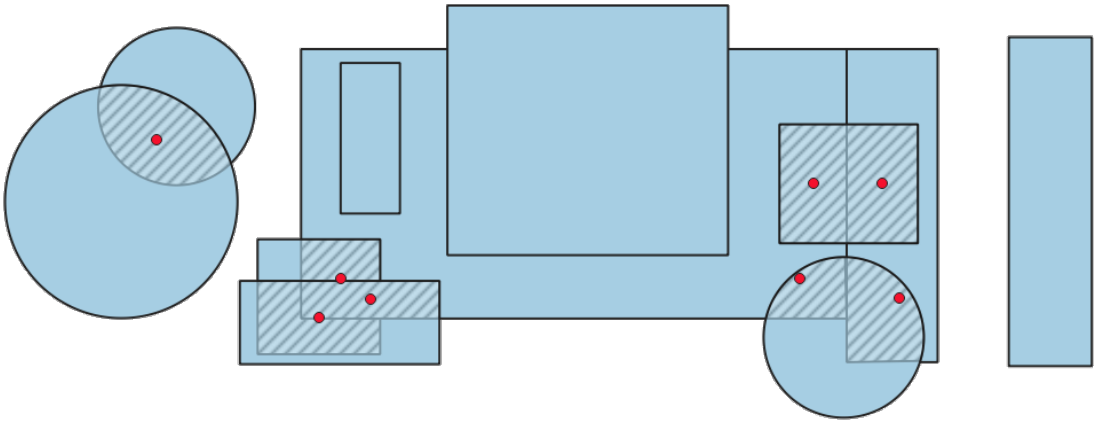


Figura24.6: Hashed polygons indicate overlapping areas smaller than the specified minimum.

Ver também

Delete overlaps, *Validate coverage*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Minimum overlap area	`MIN_OVERLAP_AR`	[numeric: double]	Minimum area of the overlap to be reported as an error. If set to 0, all overlaps are reported.
Overlap errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid points of the overlapping areas. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continua na próxima página

Tabela 24.67 – continuação da página anterior

Label	Name	Type	Description
Overlap features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Specification of the output layer for features containing overlaps. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Overlap errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_errorx`: the x coordinate of the centroid of the overlapping area. • `gc_errory`: the y coordinate of the centroid of the overlapping area. • `gc_error`: the area of the overlapping geometry. • `UNIQUE_ID` field: the unique ID of an overlapped input feature. • `gc_overlap_feature_{unique_id}`: the `UNIQUE_ID` field value for the other overlapping feature.
Overlap features	`OUTPUT`	[vector: polygon]	Output layer containing the overlapping areas. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometryoverlap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Points outside lines

Checks if the points in the input layer are covered by a line in the selected line layers. A point not covered by a line is an error.

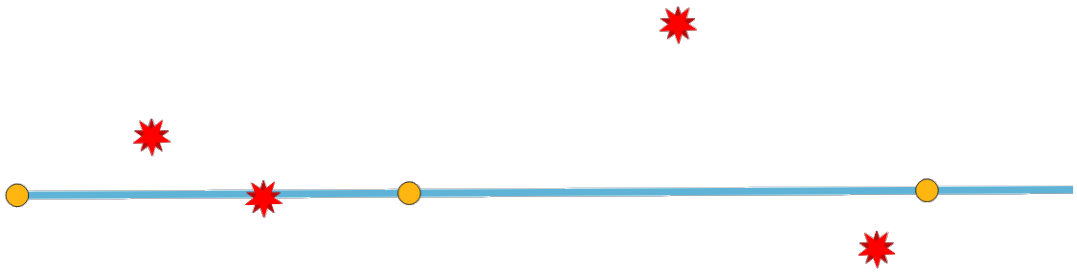


Figura24.7: Reporting errors on points not covered by a line.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: point]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Line layers	`LINES`	[vector: line][list]	Layer(s) with the lines to check against.
Points not covered by a line	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing points not covered by a line. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Points not covered by a line	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of point outside the lines. • `gc_errory`: the y coordinate of point outside the lines. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that is outside the lines.

Python code

Algorithm ID: `native:checkgeometrypointcoveredbyline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Points outside polygons

Checks if points from the input layer are in polygons from the selected polygon layers. Points that are not fully inside polygons are errors.

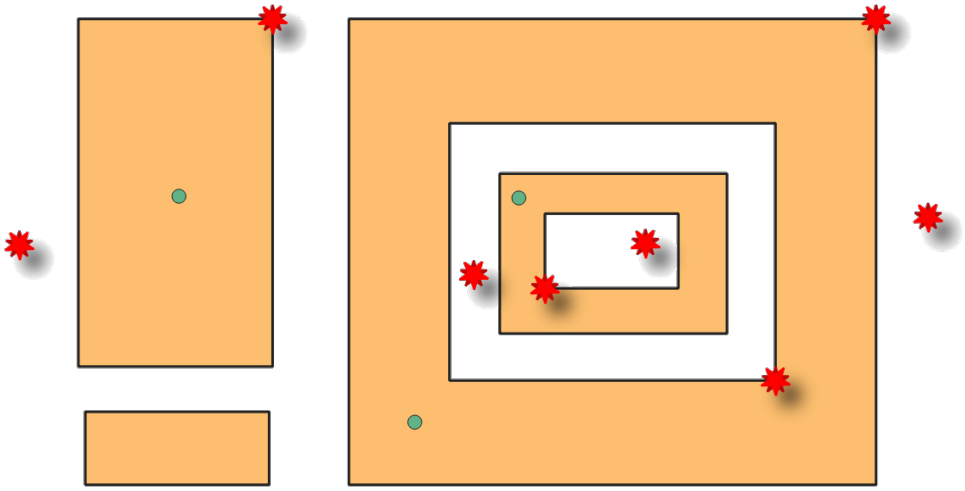


Figura24.8: Reporting errors on points outside polygons.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: point]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Polygon layers	`POLYGONS`	[vector: polygon][list]	Layer(s) with the polygons to check against.
Points outside polygons errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing points outside the polygons. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Points outside polygons errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of point outside the polygons. • `gc_errory`: the y coordinate of point outside the polygons. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that is outside the polygons.

Python code

Algorithm ID: `native:checkgeometrypointinpolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Self-contacts

Checks if the geometry has self contact points (in line or polygon), i.e., a vertex that touches more than two segments of the same ring. Self contacts are errors.

Nota

This algorithm detects invalid polygon geometries (self-contacts). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

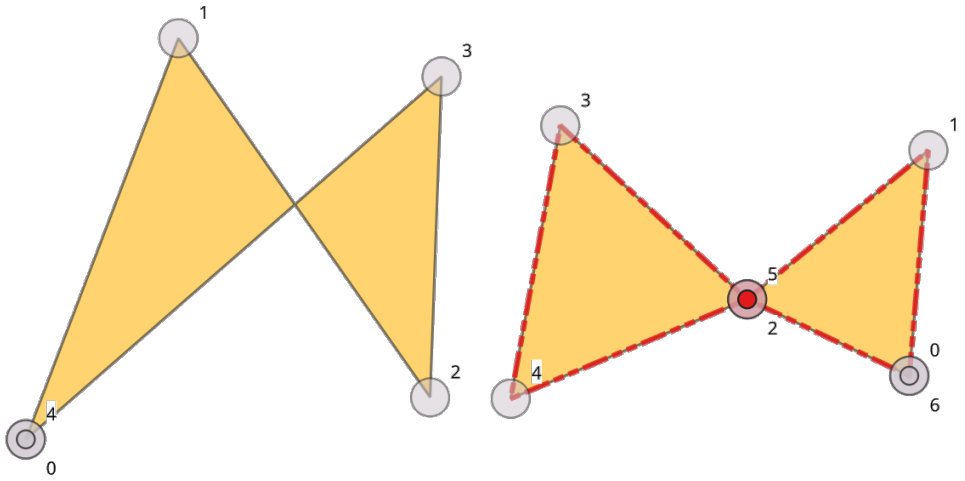


Figura24.9: Self-intersection vs self-contact.

Ver também

Self-intersections

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.

continua na próxima página

Tabela 24.76 – continuação da página anterior

Label	Name	Type	Description
Self contact error points	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Self contact features Optional	`OUTPUT`	[same as input] Default: `[Skip output]`	Line or polygon layer containing self-contact features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Self contact error points	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx: the index of the feature's geometry part where the self-contact occurs. • gc_ringidx: the index of the ring where the self-contact occurs. • gc_vertidx: the index of the vertex where the self-contact occurs. • gc_errorx: the x coordinate of the self-contact. • gc_errory: the y coordinate of the self-contact. • gc_error • UNIQUE_ID field: the unique ID of the self-contacting input feature.
Self contact features	OUTPUT	[vector: same as input]	Output polygon or line layer with features containing the self-contact features. If no self-contact features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryselfcontact

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Self-intersections

Detects self-intersections in line or polygon geometries, and reports them as errors. Self-intersections occur when the segments of a geometry cross over each other without having a common vertex.

Nota

This algorithm detects invalid polygon geometries (self-intersections). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to Do not Filter (Better Performance).

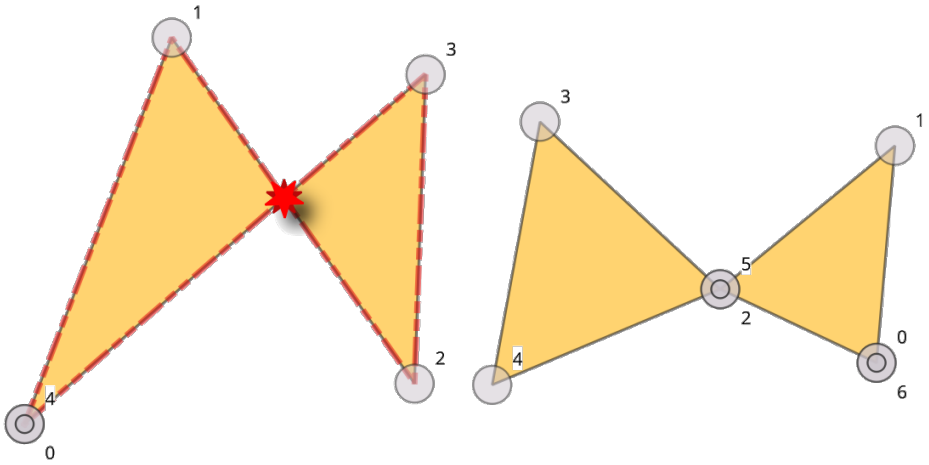


Figura24.10: Self-intersection vs self-contact.

Ver também

Split self-intersecting geometries, *Self-contacts*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification
Self-intersecting errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Self-intersecting features Optional	`OUTPUT`	[same as input] Default: `[Skip output]`	Line or polygon layer containing self-intersected features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Self-intersecting errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part where the self-intersection occurs. • `gc_ringidx`: the index of the feature's geometry ring where the self-intersection occurs. • `gc_vertidx` • `gc_errorx`: the x coordinate of the self-intersection. • `gc_errory`: the y coordinate of the self-intersection. • `gc_error` • `gc_segment_1`: the index of the first segment involved in the intersection. • `gc_segment_2`: the index of the second segment involved in the intersection. • `UNIQUE_ID` field: the unique ID of the self-intersecting input feature.
Self-intersecting features	`OUTPUT`	[vector: line, polygon]	Output polygon or line layer with features containing the self-intersecting features. If no self-intersecting features are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometryselfintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Sliver polygons

Detects sliver polygons in a polygon layer, i.e., polygons with a thinness greater than a specified value.

The thinness is the ratio between the area of the minimum square containing the polygon and the area of the polygon itself (a square has a thinness value of 1). The thinness value is between 1 and +infinity. A maximum area can be set for limiting the checks to polygons of a lower area. Polygons having a thinness higher than the maximum thinness are errors. To fix sliver polygons, use the *Fix small polygons* algorithm.

Nota

This algorithm detects invalid polygon geometries (sliver polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

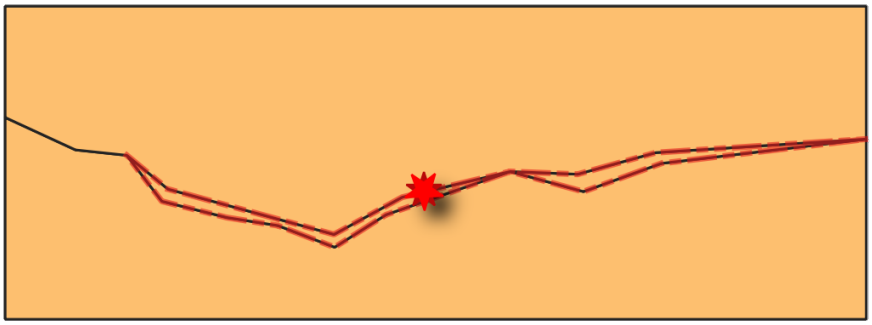


Figura24.11: Reporting errors on features with sliver polygons.

Ver também

Fix small polygons

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Maximum thinness	`MAX_THINNESS`	[numeric: double] Default: 20.0	Maximum thinness value of a polygon to be considered valid. Any polygon with a thinness value above this threshold is reported as error.
Maximum area (map units squared)	`MAX_AREA`	[numeric: double] Default: 0.0	Maximum area of a polygon to be checked (in map units). Any polygon with an area above this threshold is skipped. A value of 0 means all polygons are checked.
Sliver polygon errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the centroid points of sliver polygon features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Sliver polygon features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Polygon layer containing sliver polygon features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Sliver polygon errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the sliver polygon feature. • `gc_errory`: the y coordinate of the centroid of the sliver polygon feature. • `gc_error`: the thinness value of the sliver polygon feature. • `UNIQUE_ID` field: the unique ID of the input feature that is a sliver polygon.
Sliver polygon features	`OUTPUT`	[vector: polygon]	Output layer with features containing the sliver polygons. If no sliver polygon features are found, the output layer will be empty. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrysliverpolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Small angles

Compares the angles within line or polygon geometries to a specified threshold, and reports as error any angle below that value.

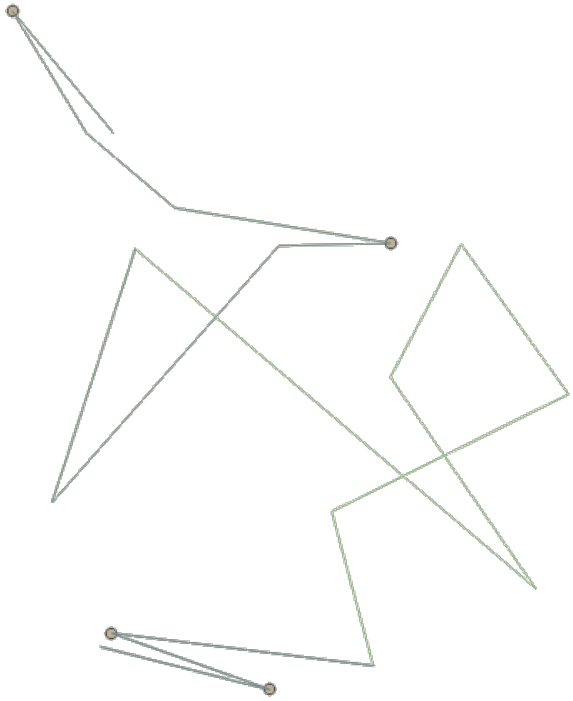


Figura24.12: Reporting errors on a line feature for angles lower than 15°.

Ver também

Delete small angles

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Minimum angle	`MIN_ANGLE`	[numeric: double]	Allowed minimum angle between adjacent segments in a feature's geometry. Any angle below this threshold is reported as error.

continua na próxima página

Tabela 24.85 – continuação da página anterior

Label	Name	Type	Description
Small angle errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the errors location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Small angle errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the small angle. • `gc_ringidx`: the index of the feature's geometry ring containing the small angle. • `gc_vertidx`: the index of the vertex with the small angle in the feature's geometry ring. • `gc_errorx`: the x coordinate of the vertex with the small angle. • `gc_errory`: the y coordinate of the vertex with the small angle. • `gc_error`: the error angle value. • `UNIQUE_ID` field: the unique ID of the input feature with the small angle.

Python code

Algorithm ID: `native:checkgeometryangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Small polygons

Detects polygon features whose area is below a specified value as errors.

Nota

This algorithm detects invalid polygon geometries (small polygons). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

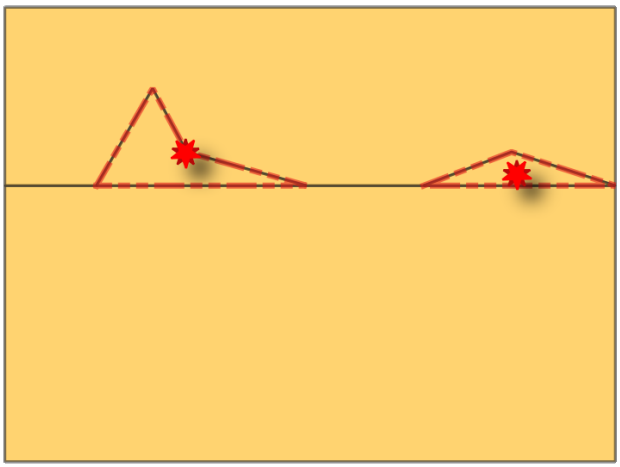


Figura24.13: Reporting errors on features with small polygons (below given area threshold).

Ver também

Fix small polygons

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Area threshold	`AREATHRESHOLD`	[numeric: double] Default: 0	Minimum area of a polygon to be considered valid, in selected units. Any polygon with an area below this threshold is reported as error.
Small polygons errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid points of small polygons. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Small polygons features Optional	`OUTPUT`	[vector: polygon] Default: `[Skip output]`	Specification of the output layer for features containing small polygon features. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Small polygons errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the small polygon. • gc_errory: the y coordinate of the centroid of the small polygon. • gc_error: the area of the small polygon. • UNIQUE_ID field: the unique ID of the input feature that is a small polygon.
Small polygons features	OUTPUT	[vector: polygon]	Output polygon layer with features containing the small polygon features. If no small polygon features are found, the output layer will be empty. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometryarea

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Small gaps

Checks for gaps between polygons in the input layer. Gaps with an area smaller than the gap threshold are reported as errors. If an allowed gaps layer is provided, the algorithm ignores gaps that fall entirely within polygons from this layer. An optional buffer can be applied to the allowed gaps.

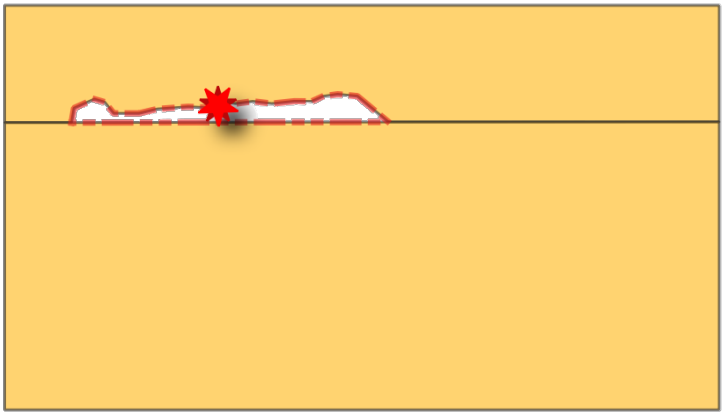


Figura24.14: Reporting errors on polygon features for gaps smaller than the specified threshold.

Ver também

Fill gaps, *Validate coverage*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Gap threshold	`GAP_THRESHOLD`	[numeric: double] Default: 0.0	Maximum area of gaps to be reported as errors, in map units. If set to 0, all the gaps are reported.
Allowed gaps layer Optional	`ALLOWED_GAPS_LAY`	[vector: polygon]	Optional layer specifying polygons whose areas should be ignored during the gap check. Gaps that fall entirely within these polygons will not be reported as errors.
Allowed gaps buffer Optional	`ALLOWED_GAPS_BUF`	[numeric: double] Default: 0.0	Buffer distance to apply to the allowed gaps layer, in selected units. Gaps located within this buffered area are ignored. Allows adding a spatial tolerance around allowed gaps to avoid reporting near-boundary gaps as errors.

continua na próxima página

Tabela 24.91 – continuação da página anterior

Label	Name	Type	Description
Neighbors layer	`NEIGHBORS`	[vector: table] Default: `[Create temporary layer]`	Specification of the output table representing the gap ID and the unique ID of its neighbor features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Gap errors Optional	`ERRORS`	[vector: point] Default: `[Skip output]`	Specification of the output layer containing the centroid points of the gaps. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Gap features	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing the gap geometries. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Neighbors layer	`NEIGHBORS`	[vector: table]	A 1–N relational table, meaning that one gap can be associated with multiple neighboring polygons. The output table contains the following fields: • `gc_errorid`: the ID of the gap. • `UNIQUE_ID` field: the unique ID of an input feature that is a neighbor of the gap.

continua na próxima página

Tabela 24.93 – continuação da página anterior

Label	Name	Type	Description
Gap errors	ERRORS	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • gc_layerid: the ID of the input layer. • gc_layername: the name of the input layer. • gc_partidx • gc_ringidx • gc_vertidx • gc_errorx: the x coordinate of the centroid of the gap. • gc_errory: the y coordinate of the centroid of the gap. • gc_error: the area of the gap. • gc_errorid: the ID of the gap.
Gap features	OUTPUT	[vector: polygon]	Output layer containing the gap geometries. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrygap

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Small segments

Calculates length of individual segments in line or polygon geometries, and reports segments shorter than a minimum length as errors.

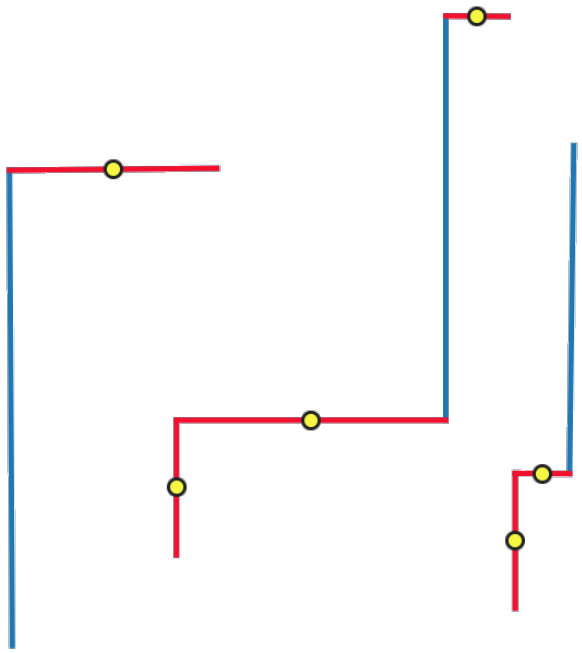


Figura24.15: Reporting errors (in red) on single line segments shorter than the given threshold.

Ver também

Delete features

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to check.
Unique feature identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification.
Minimum segment length	`MIN_SEGMENT_LE`	[numeric: double] Default: 0.0	Minimum length of segments to be considered valid, in map units.
Short segments errors	`ERRORS`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing centroid of short segments. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continua na próxima página

Tabela 24.94 – continuação da página anterior

Label	Name	Type	Description
Short segments features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer for features containing short segments. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Short segments errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the small segment. • `gc_ringidx`: the index of the feature's geometry ring containing the small segment. • `gc_vertidx`: the index of the end vertex of the small segment in the feature's geometry ring. • `gc_errorx`: the x coordinate of the centroid of the small segment. • `gc_errory`: the y coordinate of the centroid of the small segment. • `gc_error`: the error segment length. • `UNIQUE_ID` field: the unique ID of the input feature with the small segment.

continua na próxima página

Tabela 24.96 – continuação da página anterior

Label	Name	Type	Description
Short segments features	OUTPUT	[vector: same as input]	Output layer containing, for each identified short segment, the feature it belongs to. Available fields are the same as in the ERRORS output.

Python code

Algorithm ID: native:checkgeometrysegmentlength

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Strictly multipart

Checks if multipart geometries have more than one part. Multipart geometries with only one part are errors.

Ver também

Convert to strictly multipart

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	INPUT	[vector: geometry]	Layer with the geometries to check.
Unique feature identifier	UNIQUE_ID	[tablefield: any]	Field storing unique values for feature identification.
One-part geometry errors	ERRORS	[vector: point] Default: [Create temporary layer]	Specification of the output layer containing centroid of the multipart geometries that have only one part. *One of*: • Create Temporary Layer (TEMPORARY_OUTPUT) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continua na próxima página

Tabela 24.97 – continuação da página anterior

Label	Name	Type	Description
One-part geometry features Optional	`OUTPUT`	[vector: same as input] Default: `[Skip output]`	Specification of the output layer for features containing multipart geometries that have only one part. *One of*: • Skip Output • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
One-part geometry errors	`ERRORS`	[vector: point]	Output point layer representing the error locations and information. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the multipart geometry that has only one part. • `gc_errory`: the y coordinate of the centroid of the multipart geometry that has only one part. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that is multipart but has only one part.
One-part geometry features	`OUTPUT`	[vector: same as input]	Output layer containing features that are multipart but have only one part. Available fields are the same as in the `ERRORS` output.

Python code

Algorithm ID: `native:checkgeometrymultipart`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

24.1.4 Base de dados

Exportar para PostgreSQL

Exports a vector layer to a PostgreSQL database, creating a new relation. If a relation with the same name exists, it can be removed before the new relation is created. Prior to this a connection between QGIS and the PostgreSQL database has to be created (see eg *Criando uma conexão armazenada*).

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada para importar	`INPUT`	[vetor: qualquer]	Camada vetorial para adicionar ao banco de dados
Banco de dados (nome da conexão)	`DATABASE`	[string]	Name of the database connection (not the database name). Existing connections will be shown in the combobox.
Esquema (nome do esquema) Opcional	`SCHEMA`	[string] Padrão: 'público'	Nome do esquema para armazenar os dados. Pode ser um novo ou já existente.
Tabela para importar (deixe em branco para usar o nome da camada) Opcional	`TABLENAME`	[string] Padrão: ''	Defina um nome de tabela para o arquivo vetorial importado. Se nada for adicionado, será usado o nome da camada.
Campo de chave primária Opcional	`PRIMARY_KEY`	[campo da tabela: qualquer]	Sets the primary key field from an existing field in the vector layer. A column with **unique** values can be used as Primary key for the database.
Coluna de Geometria	`GEOMETRY_COLUMN`	[string] Padrão: 'geom'	Defines the name of the geometry column in the new PostgreSQL table. Geometry information for the features is stored in this column.
Codificando Opcional	`ENCODING`	[string] Padrão: 'UTF-8'	Define a codificação da camada de saída
Sobreescrever	`OVERWRITE`	[boleano] Padrão: Verdadeiro	If the specified table exists, setting this option to `True` will make sure that it is deleted and a new table will be created before the features are added. If this option is `False` and the table exists, the algorithm will throw an exception ("relation already exists").

continua na próxima página

Tabela 24.100 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Criar índice espacial	`CREATEINDEX`	[boleano] Padrão: Verdadeiro	Especifica se deve ou não ser criado um índice espacial
*Converte nomes de campos em minúsculas**	`LOWER-CASE_NAMES`	[boleano] Padrão: Verdadeiro	Converte os nomes dos campos da camada vetorial de entrada em letras minúsculas
Drop length constraint on character fields	`DROP_STRING_LE`	[boleano] Padrão: Falso	Should length constraints on character fields be dropped or not
Create single-part geometries instead of multi-part	`FORCE_SINGLEPA`	[boleano] Padrão: Falso	Caso as características da camada de saída sejam de peça única em vez de peças múltiplas. Por padrão, as informações geométricas existentes são preservadas.

Saídas

O algoritmo não tem saída.

Código Python

Algorithm ID: `native:importintopostgis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exporta para SpatiaLite

Exports a vector layer to a SpatiaLite database. Prior to this a connection between QGIS and the SpatiaLite database has to be created (see eg *Camadas SpatiaLite*).

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada para importar	`INPUT`	[vetor: qualquer]	Camada vetorial para adicionar ao banco de dados
Base de dados de arquivos	`DATABASE`	[vetor: qualquer]	O arquivo de banco de dados SQLite/SpatiaLite para conexão com
Tabela para importar (deixe em branco para usar o nome da camada) Opcional	`TABLENAME`	[string] Padrão: ‘’	Define o nome da tabela para o arquivo vetorial importado. Se nada for especificado, será usado o nome da camada.
Campo de chave primária Opcional	`PRIMARY_KEY`	[campo da tabela: qualquer]	Usar um campo na camada vetorial de entrada como chave primária

continua na próxima página

Tabela 24.101 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Coluna de Geometria	`GEOMETRY_COLUMN`	[string] Padrão: 'geom'	Define o nome da coluna de geometria na nova tabela SpatiaLite. As informações geométricas para as características são armazenadas nesta coluna.
Codificando Opcional	`ENCODING`	[string] Padrão: 'UTF-8'	Define a codificação da camada de saída
Sobreescrever	`OVERWRITE`	[boleano] Padrão: Verdadeiro	Se a tabela especificada existir, definindo esta opção como "Verdadeira", certificar-se de que caso seja apagada, uma nova tabela será criada antes que as características da camada sejam adicionadas. Se esta opção for ``Falsa" e a tabela existir, o algoritmo lançará uma exceção ("tabela já existe").
Criar índice espacial	`CREATEINDEX`	[boleano] Padrão: Verdadeiro	Especifica se deve ou não ser criado um índice espacial
*Converte nomes de campos em minúsculas**	`LOWERCASE_NAMES`	[boleano] Padrão: Verdadeiro	Converte os nomes dos campos da camada vetorial de entrada em letras minúsculas
Drop length constraint on character fields	`DROP_STRING_LE`	[boleano] Padrão: Falso	Should length constraints on character fields be dropped or not
Create single-part geometries instead of multi-part	`FORCE_SINGLEPA`	[boleano] Padrão: Falso	Caso as características da camada de saída sejam de peça única em vez de peças múltiplas. Por padrão, as informações geométricas existentes são preservadas.

Saídas

O algoritmo não tem saída.

Código Python

Algorithm ID: `native:importintospatialite`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Package layers

Adiciona camadas a um GeoPackage.

If the GeoPackage exists and `Overwrite existing GeoPackage` is checked, it will be overwritten (removed and recreated). If the GeoPackage exists and `Overwrite existing GeoPackage` is not checked, the layer will be appended.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`LAYERS`	[vetor: qualquer] [lista]	The (vector) layers to import into the GeoPackage. Raster layers are not supported. If a raster layer is added, a `QgsProcessingException` will be thrown.
Substituir GeoPackage existente	`OVERWRITE`	[boleano] Padrão: Falso	If the specified GeoPackage exists, setting this option to `True` will make sure that it is deleted and a new one will be created before the layers are added. If set to `False`, layers will be appended.
Salvar estilos de camada no GeoPackage	`SAVE_STYLES`	[boleano] Padrão: Verdadeiro	Save the layer styles.
Salvar apenas feições selecionadas	`SELECTED_FEATURE`	[boleano] Padrão: Falso	If a layer has a selection, setting this option to `True` will result in only selected features being saved. For layers without a selection all features will be saved.
Export related layers following relations defined in the project	`EXPORT_RELATED_L`	[boleano] Padrão: Falso	If an input layer has *relations* set in the project, setting this option to `True` will result in exporting also its related layer(s). If the layer has features selected, then only their related features will be exported unless the related layer was also an input layer.
GeoPackage de Destino	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Especifique onde armazenar o arquivo GeoPackage. Um de • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Layers within new package	`OUTPUT_LAYERS`	[cadeia] [lista]	A lista de camadas adicionadas ao GeoPackage.

Código Python

Algorithm ID: `native:package`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

PostgreSQL executa e carrega SQL

Permite que uma consulta de banco de dados SQL seja realizada em um banco de dados PostgreSQL conectado ao QGIS e carrega o resultado. O algoritmo **não** criará uma nova camada: ele é projetado para executar consultas na própria camada.

Exemplo

1. Defina todos os valores de um campo existente para um valor fixado. A consulta SQL de string será:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   In the example above, the values of the field `field_to_update` of the table `your_table` will be all set to `20`.

2. Create a new `area` column and calculate the area of each feature with the `ST_AREA` PostGIS function.

   ```
   -- Create the new column "area" on the table your_table"
   ALTER TABLE your_table ADD COLUMN area double precision;
   -- Update the "area" column and calculate the area of each feature:
   UPDATE your_table SET area=ST_AREA(geom);
   ```

Ver também

PostgreSQL executa SQL, *Executar SQL*, *SpatiaLite execute SQL*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Banco de dados (nome da conexão)	`DATABASE`	[string]	The database connection (not the database name). Existing connections will be shown in the combobox.
Consulta SQL	`SQL`	[string]	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.
Nome do campo de ID único	`ID_FIELD`	[string] Padrão: 'id'	Define o campo chave primário (uma coluna na tabela de resultados)
Nome do campo de geometria Opcional	`GEOMETRY_FIELD`	[string] Padrão: 'geom'	Nome da coluna de geometria (uma coluna na tabela de resultados)

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada SQL	`OUTPUT`	[vetor: qualquer]	A camada vetorial resultante a ser carregada no QGIS.

Código Python

Algorithm ID: `qgis:postgisexecuteandloadsql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

PostgreSQL executa SQL

Allows a SQL database query to be performed on a PostgreSQL database connected to QGIS. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

Exemplo

1. Defina todos os valores de um campo existente para um valor fixado. A consulta SQL de string será:

   ```
   UPDATE your_table SET field_to_update=20;
   ```

   In the example above, the values of the field `field_to_update` of the table `your_table` will be all set to `20`.

2. Create a new `area` column and calculate the area of each feature with the `ST_AREA` PostGIS function.

   ```
   -- Create the new column "area" on the table your_table"
   ALTER TABLE your_table ADD COLUMN area double precision;
   -- Update the "area" column and calculate the area of each feature:
   UPDATE your_table SET area=ST_AREA(geom);
   ```

Ver também

PostgreSQL executa e carrega SQL, *Executar SQL*, *SpatiaLite execute SQL*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Banco de dados (nome da conexão)	`DATABASE`	[string]	The database connection (not the database name). Existing connections will be shown in the combobox.
Consulta SQL	`SQL`	[string]	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.

Saídas

No output is created. The SQL query is executed in place.

Código Python

Algorithm ID: `native:postgisexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

SpatiaLite execute SQL

Allows a SQL database query to be performed on a SpatiaLite database. The database is determined by an input layer or file. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

Ver também

PostgreSQL executa SQL, *Executar SQL*

For some SQL query examples see *PostgreSQL Query Examples*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Database layer (or file)	`DATABASE`	[arquivo]	O arquivo de banco de dados SQLite/SpatiaLite para conexão com
Consulta SQL	`SQL`	[string] Padrão: ''	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.

Saídas

No output is created. The SQL query is executed in place.

Código Python

Algorithm ID: `native:spatialiteexecutesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

SpatiaLite execute SQL (registered DB)

Allows a SQL database query to be performed on a SpatiaLite database connected to QGIS. The algorithm **won't** create a new layer: it is designed to run queries on the layer itself.

Ver também

PostgreSQL executa SQL, *Executar SQL*

For some SQL query examples see *PostgreSQL Query Examples*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Base de dados	`DATABASE`	[enumeração] Padrão: não definido	Select a SQLite/SpatiaLite database connected to the current session
Consulta SQL	`SQL`	[string] Padrão: ''	Defines the SQL query, for example `'UPDATE my_table SET field=10'`.

Saídas

No output is created. The SQL query is executed in place.

Código Python

Algorithm ID: `native:spatialiteexecutesqlregistered`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.5 Ferramentas de arquivo

Download file via HTTP(S)

Downloads a URL to the file system with an HTTP(S) GET or POST request.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
URL	`URL`	[string]	A URL do arquivo a ser baixado.
Destino do arquivo	`OUTPUT`	[string] Padrão: `[Salvar em arquivo temporário]`	Specification of the file destination. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Método	`METHOD`	[enumeração] Padrão: 0	O método HTTP a ser utilizado para a solicitação. As opções são: • 0 — LER • 1 — CRIAR
Dados Opcional	`DATA`	[string]	Os dados a serem acrescentados no corpo se o pedido for um 'CRIAR'.

Saídas

Rótulo	Nome	Tipo	Descrição
Destino do arquivo	`OUTPUT`	[string]	A localização do arquivo baixado

Código Python

Algorithm ID: `native:filedownloader`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

HTTP(S) POST/GET request

Performs a HTTP(S) POST/GET request and returns the HTTP status code and the reply data. If an error occurs then the error code and the message will be returned. Optionally, the result can be written to a file on the disk. By default the algorithm will warn on errors. Optionally, the algorithm can be set to treat HTTP errors as failures.

Parâmetros

Rótulo	Nome	Tipo	Descrição
URL or file path	`URL`	[string]	The URL or path of the file to open.
Método	`METHOD`	[enumeração] Padrão: 0	O método HTTP a ser utilizado para a solicitação. As opções são: • 0 — LER • 1 — CRIAR
POST Data Opcional	`DATA`	[string]	Os dados a serem acrescentados no corpo se o pedido for um 'CRIAR'.
Authentication Opcional	`AUTH_CONFIG`	[authconfig] Default: No authentication	An authentication configuration to pass
Consider HTTP errors as failures	`FAIL_ON_ERROR`	[boleano] Padrão: Falso	If set, the algorithm will fail on encountering an HTTP error.
Destino do arquivo Opcional	`OUTPUT`	[string] Default: `[Skip Output]`	The result can be written to a file instead of being returned as a string. Specification of the file destination. *One of*: • Ignorar Saída • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Destino do arquivo	`OUTPUT`	[arquivo]	The downloaded file with the returned result
HTTP Status	`STATUS_CODE`	[string]	The HTTP Status
Network error code	`ERROR_CODE`	[string]	The error code when it fails (like as well on 404 HTTP Status etc.)
Network error message	`ERROR_MESSAGE`	[string]	A string containing the error message in case of failure
Result data	`RESULT_DATA`	[string]	A string containing the result data in case of success

Código Python

Algorithm ID: `native:httprequest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Open file or URL

Opens files in their default associated application, or URLs in the user's default web browser.

Parâmetros

Rótulo	Nome	Tipo	Descrição
URL or file path	`URL`	[string]	The URL or path of the file to open.

Saídas

O algoritmo não tem saída.

Código Python

Algorithm ID: `native:openurl`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.6 Fix Geometry

Add missing vertices along borders

Adds the missing vertices along polygons borders, based on an error layer from the *Missing vertices along borders* algorithm.

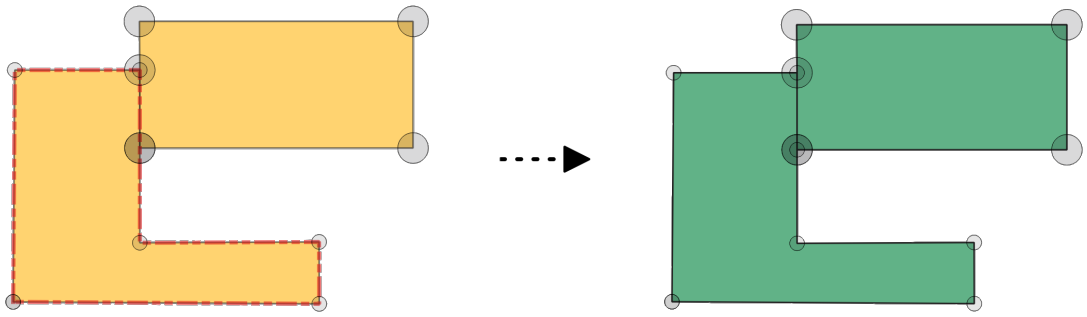


Figura24.16: Fixing missing vertices along polygon borders.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Border vertices fixed layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

continua na próxima página

Tabela 24.115 – continuação da página anterior

Label	Name	Type	Description
Report layer from fixing border vertices	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing border vertices	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the missing vertex. • `gc_errory`: the y coordinate of the missing vertex. • `gc_error` • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature that has a missing vertex.
Border vertices fixed layer	`OUTPUT`	[vector: polygon]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrymissingvertex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Convert to strictly multipart

Converts multipart geometries that consists of only one geometry into singlepart geometries, based on an error layer from the *Strictly multipart* algorithm. Layer geometry type will not be changed and will remain multipart.

Ver também

Promover para multipartes

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Strictly-multipart layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing multipart	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the centroids of the fixed geometries. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing multipart	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the multipart geometry that has only one part. • `gc_errory`: the y coordinate of the centroid of the multipart geometry that has only one part. • `gc_error` • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature that is multipart but has only one part.
Strictly-multipart layer	`OUTPUT`	[vector: same as input]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrymultipart`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Delete duplicated vertices

Deletes duplicated vertices from the input geometries, based on errors reported by the *Duplicated vertices* algorithm.

Ver também

Remover vértices duplicados

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Fixed duplicate vertices layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing duplicate vertices	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing duplicate vertices	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the duplicate vertex. • `gc_ringidx`: the index of the feature's geometry ring containing the duplicate vertex. • `gc_vertidx`: the index of the duplicate vertex in the feature's geometry ring. • `gc_errorx`: the x coordinate of the duplicate vertex. • `gc_errory`: the y coordinate of the duplicate vertex. • `gc_error` • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature that has duplicate vertices.
Fixed duplicate vertices layer	`OUTPUT`	[vector: same as input]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryduplicatenodes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Delete features

Deletes error features based on an error layer from some check geometry algorithms, such as:

- *Features inside polygon*
- *Degenerate polygons*
- *Small segments*
- *Duplicated geometries*

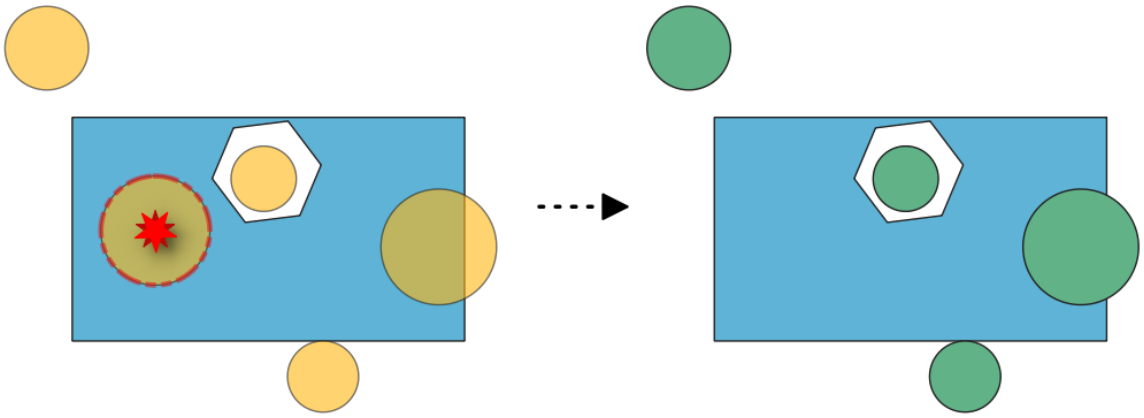


Figura24.17: Fixing errors from the "*Features inside polygon*" algorithm check.

Ver também

Excluir geometrias duplicadas, *Drop geometries*

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: geometry]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.

continua na próxima página

Tabela 24.124 – continuação da página anterior

Label	Name	Type	Description
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Cleaned layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from deleting features	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Outputs

Label	Name	Type	Description
Report layer from deleting features	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the erroneous geometry. • `gc_errory`: the y coordinate of the centroid of the erroneous geometry. • `gc_error`: the index of the feature where the error belongs. • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed. • `UNIQUE_ID` field: the unique ID of the input feature with the error.
Cleaned layer	`OUTPUT`	[vector: same as input]	Output layer with features removed based on detected errors. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrydeletefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Delete overlaps

Deletes overlapping areas based on an error layer from the *Overlaps* algorithm.

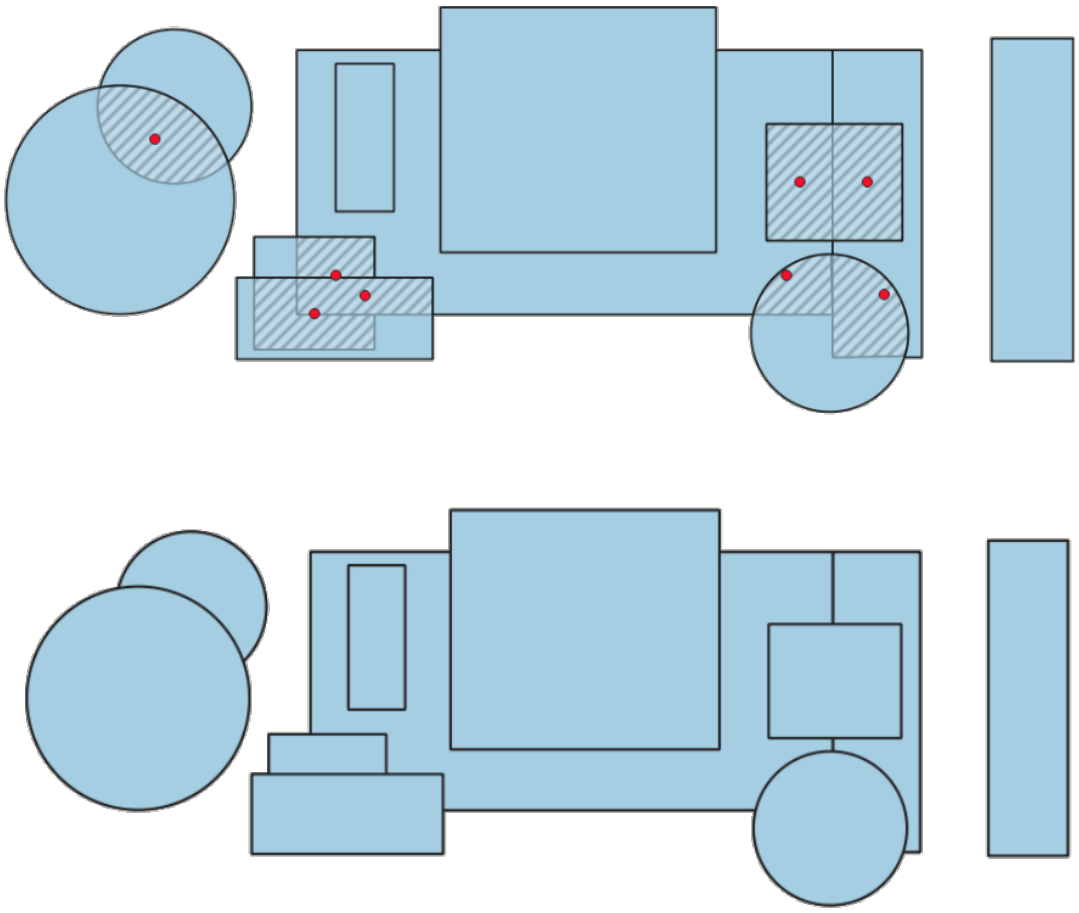


Figura24.18: Deleting overlapping areas.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of overlap feature unique identifier	`OVERLAP_FEATURE_UN`	[tablefield: any]	Field storing unique values for the overlapping feature identification in input layer, as set in the check algorithm.
Field of error value	`ERROR_VALUE_ID`	[tablefield: any]	Field storing the error value for each feature, as set in the check algorithm.
No-overlap layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing overlaps	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing overlaps	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_errorx`: the x coordinate of the centroid of the overlapping area. • `gc_errory`: the y coordinate of the centroid of the overlapping area. • `gc_error`: the area of the overlapping geometry. • `UNIQUE_ID` field: the unique ID of the unique ID of an overlapped input feature. • `gc_overlap_feature_{unique_id}`: the `UNIQUE_ID` field value for the other overlapping feature. • `report`: a text field describing the fix applied or justifying the failure. Possible values are: – Remove overlapping area from neighboring polygon with shortest shared edge – Error is obsolete • `error_fixed`: a boolean field indicating whether the error was fixed.
No-overlap layer	`OUTPUT`	[vector: polygon]	Output layer with input features edited. Overlapping areas reported as errors are removed. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryoverlap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Delete small angles

Deletes vertices based on an error layer from the *Small angles* algorithm. When deletion of a vertex results in a duplicate vertex (when a spike vertex is deleted), the duplicate vertex is deleted to keep a single vertex and preserve topology.

> **Atenção**
>
> This algorithm removes the vertex at the reported small angles, generating new segments that may form a new small angle.

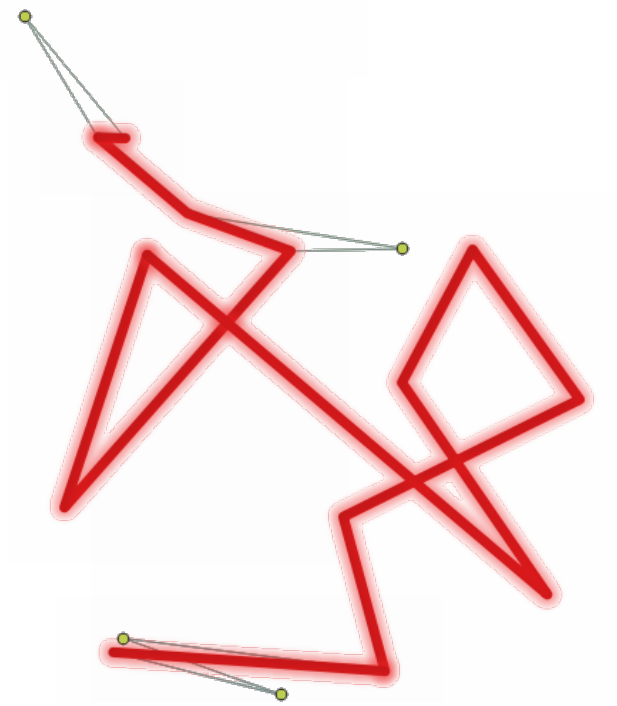


Figura24.19: Fixing reported errors on a line feature for angles lower than 15°.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.

continua na próxima página

Tabela 24.129 – continuação da página anterior

Label	Name	Type	Description
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Small angle fixed layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing small angles	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing small angles	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the small angle. • `gc_ringidx`: the index of the feature's geometry ring containing the small angle. • `gc_vertidx`: the index of the vertex with the small angle in the feature's geometry ring. • `gc_errorx`: the x coordinate of the vertex with the small angle. • `gc_errory`: the y coordinate of the vertex with the small angle. • `gc_error`: the error angle value. • `UNIQUE_ID` field: the unique ID of the input feature with the small angle. • `report`: a text field describing the fix applied or justifying the failure. • `error_fixed`: a boolean field indicating whether the error was fixed.
Small angle fixed layer	`OUTPUT`	[same as input]	Output layer with the geometry fix applied to the input features The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Fill gaps

Fills the gaps based on a gap and neighbors layer from the *Small gaps* algorithm. Three different fixing methods are available.

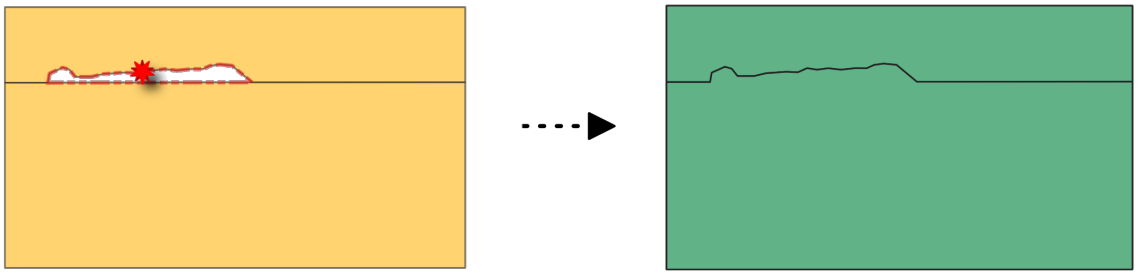


Figura24.20: Fixing gap between polygon features using the `Add to largest neighbouring area` method.

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Neighbors layer	`NEIGHBORS`	[vector: table]	Relational table with the unique ID of the gap and its neighbors in input layer, generated by the check algorithm.
Gaps layer	`GAPS`	[vector: polygon]	Polygon layer with the gaps geometries, generated by the check algorithm.
Method	`METHOD`	[enumeration] Default: 0	Method to apply to the gap fixing. Possible values are: • 0 — `Add to longest shared edge`: adds the gap to the neighboring feature with the longest shared edge. • 1 — `Create new feature`: creates a new feature for each gap. • 2 — `Add to largest neighbouring area`: adds the gap to the neighboring feature with the largest area.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any] Default: gc_errorid	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of error id	`ERROR_ID_IDX`	[tablefield: any]	Field storing the error ID for each gap, as output by the check algorithm.

continua na próxima página

Tabela 24.132 – continuação da página anterior

Label	Name	Type	Description
Gaps-filled layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing gaps	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing gaps	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the gap. • `gc_errory`: the y coordinate of the centroid of the gap. • `gc_error`: the area of the gap. • `gc_errorid`: the ID of the gap. • `report`: a text field mentioning the method used to fix the gap. • `error_fixed`: a boolean field indicating whether the error was fixed.
Gaps-filled layer	`OUTPUT`	[vector: polygon]	Output layer of polygons without gaps. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometrygap`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Fill holes

Deletes holes in polygon geometries based on an error layer from the *Holes* algorithm.

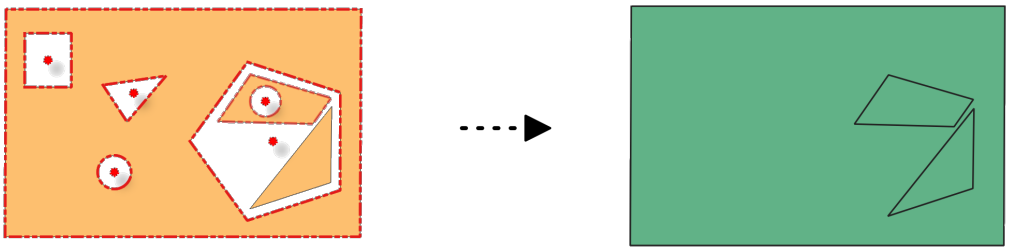


Figura24.21: Deleting holes in a polygon feature.

Ver também

Excluir buracos

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Holes-filled layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing holes	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing holes	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part containing the hole. • `gc_ringidx`: the index of the feature's geometry ring containing the hole. • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the hole. • `gc_errory`: the y coordinate of the centroid of the hole. • `gc_error` • `UNIQUE_ID` field: the unique ID of the input feature that has a hole. • `report`: a text field describing the fix applied or justifying the failure. • `error_fixed`: a boolean field indicating whether the error was fixed.
Holes-filled layer	`OUTPUT`	[vector: polygon]	Output layer of polygons without holes. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryhole`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Fix small polygons

Merges neighboring polygons according to the chosen method, based on an error layer from the *Small polygons* or *Sliver polygons* algorithm.

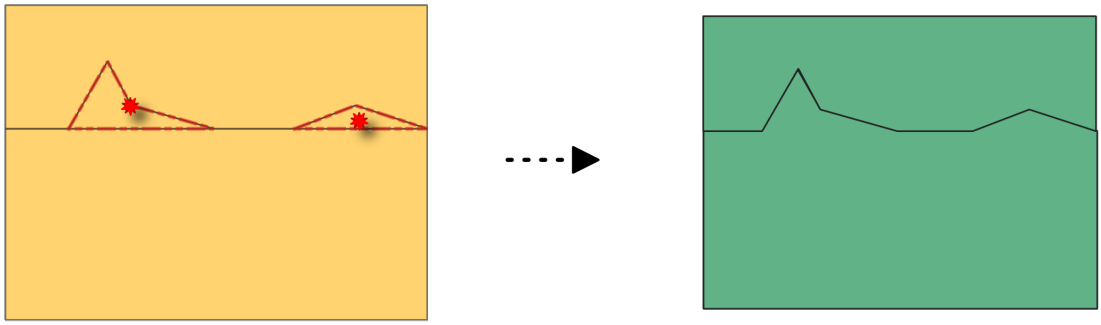


Figura24.22: Before and after fixing small polygons (method: merge with neighboring polygon with largest area).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: polygon]	Layer with the geometries to fix, same as input in the check algorithm.
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm.
Method	`METHOD`	[enumeration] Default: 0	Method to apply to the small polygon geometries: • 0 — `Merge with neighboring polygon with longest shared edge`: merges the small polygon geometry with the neighboring polygon that has the longest shared edge. • 1 — `Merge with neighboring polygon with largest area`: merges the small polygon geometry with the neighboring polygon that has the largest area. • 2 — `Merge with neighboring polygon with identical attribute value, if any, or leave as is`: merges the small polygon geometry with a neighboring polygon that has the same attribute value, if such a polygon exists.

continua na próxima página

Tabela 24.138 – continuação da página anterior

Label	Name	Type	Description
Field to consider when merging polygons with the identical attribute method Optional	`MERGE_ATTRIBUT`	[tablefield: any]	Field in the input layer to consider when merging polygons with the identical attribute method.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm.
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Small polygons merged layer	`OUTPUT`	[vector: polygon] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from merging small polygons	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from merging small polygons	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx` • `gc_ringidx` • `gc_vertidx` • `gc_errorx`: the x coordinate of the centroid of the small polygon. • `gc_errory`: the y coordinate of the centroid of the small polygon. • `gc_error`: the area of the small polygon. • `UNIQUE_ID` field: the unique ID of the input feature that is a small polygon. • `report`: a text field describing the fix applied or justifying the failure. • `error_fixed`: a boolean field indicating whether the error was fixed.
Small polygons merged layer	`OUTPUT`	[vector: polygon]	Output layer of polygons with small polygons merged according to the chosen method. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryarea`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Split self-intersecting geometries

Splits self-intersecting geometries based on an error layer from the *Self-intersections* algorithm.

Nota

This algorithm detects invalid polygon geometries (self-intersections). To ensure these invalid features are not filtered out before processing, open Advanced options next to the *Input layer* and set *Invalid feature filtering* to `Do not Filter (Better Performance)`.

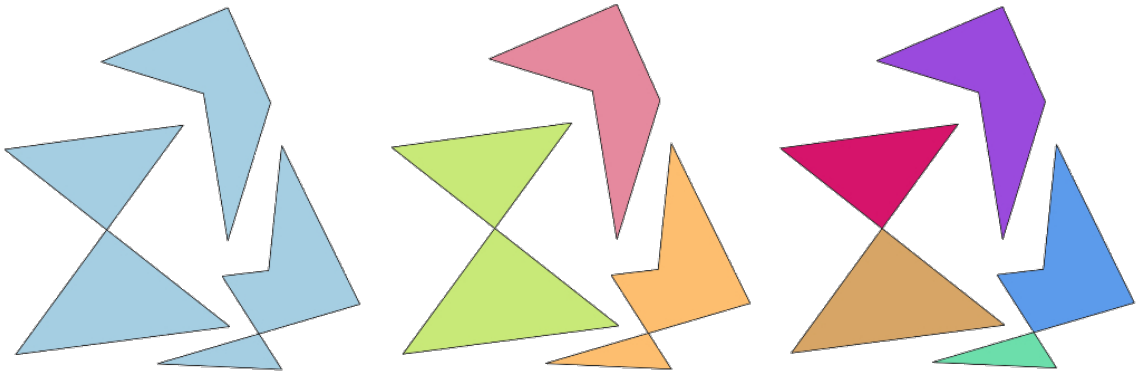


Figura24.23: Fixing self-intersection using various methods: as multi-part (middle), and multiple single-part geometries (right).

Parameters

Basic parameters

Label	Name	Type	Description
Input layer	`INPUT`	[vector: line, polygon]	Layer with the geometries to fix, same as input in the check algorithm
Error layer	`ERRORS`	[vector: point]	Point layer with the errors location, generated by the check algorithm
Method	`METHOD`	[enumeration]	Method to apply to the self-intersecting geometries: • 0 – `Split feature into a multi-part feature`: splits the self-intersecting geometry into multiple parts stored in a single multipart feature. • 1 – `Split feature into multiple single-part features`: splits the self-intersecting geometry into multiple parts, each part stored in a separate feature.
Field of original feature unique identifier	`UNIQUE_ID`	[tablefield: any]	Field storing unique values for feature identification in input layer, as set in the check algorithm
Field of part index	`PART_IDX`	[tablefield: integer] Default: `gc_partidx`	Field storing the erroneous feature's geometry part number.
Field of ring index	`RING_IDX`	[tablefield: integer] Default: `gc_ringidx`	Field storing the erroneous feature's geometry ring number.

continua na próxima página

Tabela 24.141 – continuação da página anterior

Label	Name	Type	Description
Field of vertex index	`VERTEX_IDX`	[tablefield: integer] Default: `gc_vertidx`	Field storing the erroneous feature's geometry vertex number.
Field of segment 1	`SEGMENT_1`	[tablefield: integer] Default: `gc_segment_1`	Field storing the number of the first segment involved in the erroneous intersection.
Field of segment 2	`SEGMENT_2`	[tablefield: integer] Default: `gc_segment_2`	Field storing the number of the second segment involved in the erroneous intersection.
Self-intersections fixed layer	`OUTPUT`	[vector: same as input] Default: `[Create temporary layer]`	Specification of the output layer containing fixed features. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.
Report layer from fixing self-intersections	`REPORT`	[vector: point] Default: `[Create temporary layer]`	Specification of the output layer containing the fixes location. *One of*: • Create Temporary Layer (`TEMPORARY_OUTPUT`) • Save to File… • Save to Geopackage… • Save to Database Table… The file encoding can also be changed here.

Advanced parameters

Label	Name	Type	Description
Tolerance	`TOLERANCE`	[numeric: integer] Default: 8	Numerical precision of geometric operations, given as an integer n, meaning that two vertices less than 10^{-n} apart (in map units) are considered to be merged.

Outputs

Label	Name	Type	Description
Report layer from fixing self-intersections	`REPORT`	[vector: point]	Output point layer representing the error locations and fix applied. The output layer contains the following fields: • `gc_layerid`: the ID of the input layer. • `gc_layername`: the name of the input layer. • `gc_partidx`: the index of the feature's geometry part where the self-intersection occurs. • `gc_ringidx`: the index of the feature's geometry ring where the self-intersection occurs. • `gc_vertidx` • `gc_errorx`: the x coordinate of the self-intersection. • `gc_errory`: the y coordinate of the self-intersection. • `gc_error` • `gc_segment_1`: the index of the first segment involved in the intersection. • `gc_segment_2`: the index of the second segment involved in the intersection. • `UNIQUE_ID` field: the unique ID of the self-intersecting input feature. • `report`: a text field describing the fix applied. • `error_fixed`: a boolean field indicating whether the error was fixed.
Self-intersections fixed layer	`OUTPUT`	[vector: same as input]	Output layer with the geometry fix applied to the input features. The layer contains the same fields as in the input layer.

Python code

Algorithm ID: `native:fixgeometryselfintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

24.1.7 GPS

Converter dados do GPS

Utiliza a ferramenta GPSBabel para converter um arquivo de dados GPS de uma gama de formatos para o formato padrão GPX.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Arquivo de entrada	`INPUT`	[arquivo]	Arquivo contendo os dados a serem convertidos
Formato	`FORMAT`	[enumeração]	Formato do arquivo a ser convertido, a partir desta lista <https://www.gpsbabel.org/capabilities.html>`_.
Tipo de feição	`FEATURE_TYPE`	[enumeração] Padrão: 0	O tipo de dados a serem convertidos • 0 — Pontos de identificação • 1 — Rotas • 2 — Trilhas
Saída	`OUTPUT`	[vetor: qualquer] Padrão: `[Salvar em arquivo temporário]`	Specification of the output GPX file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT_LAYER`	[vetor: qualquer]	Camada de saída com dados em formato padrão GPX

Código Python

Algorithm ID: `native:convertgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Converter tipo de recurso GPX

Utiliza a ferramenta GPSBabel para converter características GPX de um tipo para outro (por exemplo, convertendo todas as características de pontos para uma rota).

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Arquivo de entrada	`INPUT`	[arquivo]	Arquivo contendo os dados a serem convertidos
Conversão	`CONVERSION`	[enumeração] Padrão: 0	O tipo de conversão para aplicar • 0 — Pontos de identificação a partir de rota • 1 — Pontos de identificação a partir de trilha • 2 — Rotas a partir de pontos de identificação • 3 — Trilhas a partir de pontos de identificação
Saída	`OUTPUT`	[vetor: ponto ou linha] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Saída	`OUTPUT`	[vetor: qualquer]	Camada de saída com características GPX convertidas

Código Python

Algorithm ID: `native:convertgpxfeaturetype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Baixar dados de GPS do dispositivo

Utiliza a ferramenta GPSBabel para baixar dados de um dispositivo GPS para o formato padrão GPX.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Dispositivo	`DEVICE`	[enumeração] Default:*Garmin serial*	The GPS device used to create the data. Must be declared in *GPS Settings* dialog.
Porta	`PORT`	[enumeração]	The port the device is connected to. Available ports depend on the OS.
Tipo de feição	`FEATURE_TYPE`	[enumeração] Padrão: 0	O tipo de dados a serem convertidos • 0 — Pontos de identificação • 1 — Rotas • 2 — Trilhas
Saída	`OUTPUT`	[vetor: qualquer] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT_LAYER`	[vetor: qualquer]	Camada de saída com dados em formato padrão GPX

Código Python

ID do algoritmo: `native:downloadgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Carregar dados GPS no dispositivo

Utiliza a ferramenta GPSBabel para carregar dados para um dispositivo GPS a partir do formato padrão GPX.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Arquivo de entrada	`INPUT`	[arquivo]	: arquivo:.*GPX* arquivo contendo os dados a serem carregados
Dispositivo	`DEVICE`	[enumeração] Default:*Garmin serial*	The GPS device you would like to upload the data to. Must be declared in *GPS Settings* dialog.
Porta	`PORT`	[enumeração]	The port the device is connected to. Available ports depend on the OS.
Tipo de feição	`FEATURE_TYPE`	[enumeração] Padrão: 0	O tipo de dados a serem carregados • 0 — Pontos de identificação • 1 — Rotas • 2 — Trilhas

Saídas

Nenhuma saída é fornecida. Se bem-sucedido, os dados são carregados no dispositivo.

Código Python

ID do algoritmo: `native:uploadgpsdata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.8 Interpolação

Heatmap (kernel density estimation)

Creates a density (heatmap) raster of an input point vector layer using kernel density estimation.

The density is calculated based on the number of points in a location, with larger numbers of clustered points resulting in larger values. Heatmaps allow easy identification of *hotspots* and clustering of points.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de pontos	`INPUT`	[vetor: ponto]	Point vector layer to use for the heatmap
Raio	`RADIUS`	[numeric: double] Padrão: 100.0	Heatmap search radius (or kernel bandwidth) in map units. The radius specifies the distance around a point at which the influence of the point will be felt. Larger values result in greater smoothing, but smaller values may show finer details and variation in point density.
Output raster size	`PIXEL_SIZE`	[numeric: double] Padrão: 0.1	Pixel size of the output raster layer in layer units. In the GUI, the size can be specified by the number of rows (`Number of rows`) / columns (`Number of columns`) **or** the pixel size(`Pixel Size X` / `Pixel Size Y`). Increasing the number of rows or columns will decrease the cell size and increase the file size of the output raster. The values in `Rows`, `Columns`, `Pixel Size X` and `Pixel Size Y` will be updated simultaneously - doubling the number of rows will double the number of columns, and the cell size will be halved. The extent of the output raster will remain the same (approximately).
Radius from field Opcional	`RADIUS_FIELD`	[campo de tabela: numérico]	Sets the search radius for each feature from an attribute field in the input layer.
Weight from field Opcional	`WEIGHT_FIELD`	[campo de tabela: numérico]	Allows input features to be weighted by an attribute field. This can be used to increase the influence certain features have on the resultant heatmap.
Kernel shape	`KERNEL`	[enumeração] Padrão: *0*	Controls the rate at which the influence of a point decreases as the distance from the point increases. Different kernels decay at different rates, so a triweight kernel gives features greater weight for distances closer to the point then the Epanechnikov kernel does. Consequently, triweight results in "sharper" hotspots and Epanechnikov results in "smoother" hotspots. There are many shapes available (please see the Wikipedia page for further information): • 0 — Quartico • 1 — Triangular • 2 — Uniforme • 3 — Triweight • 4 — Epanechnikov

continua na próxima página

Tabela 24.151 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Decay ratio (Triangular kernels only) Opcional	`DECAY`	[numeric: double] Padrão: *0.0*	Can be used with Triangular kernels to further control how heat from a feature decreases with distance from the feature. • A value of 0 (=minimum) indicates that the heat will be concentrated in the center of the given radius and completely extinguished at the edge. • A value of 0.5 indicates that pixels at the edge of the radius will be given half the heat as pixels at the center of the search radius. • A value of 1 means the heat is spread evenly over the whole search radius circle. (This is equivalent to the 'Uniform' kernel.) • A value greater than 1 indicates that the heat is higher towards the edge of the search radius than at the center.
Output value scaling	`OUTPUT_VALUE`	[enumeração] Padrão: *0*	Allow to change the values of the output heatmap raster. One of: • 0 — Bruto • 1 — Scaled
Mapa de calor	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer with kernel density values. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Mapa de calor	`OUTPUT`	[raster]	Camada raster com valores de densidade de kernel

Exemplo: Criando um mapa de calor

For the following example, we will use the `airports` vector point layer from the QGIS sample dataset (see *Baixando dados de amostra*). Another excellent QGIS tutorial on making heatmaps can be found at http://qgistutorials.com.

In Fig.24.24, the airports of Alaska are shown.

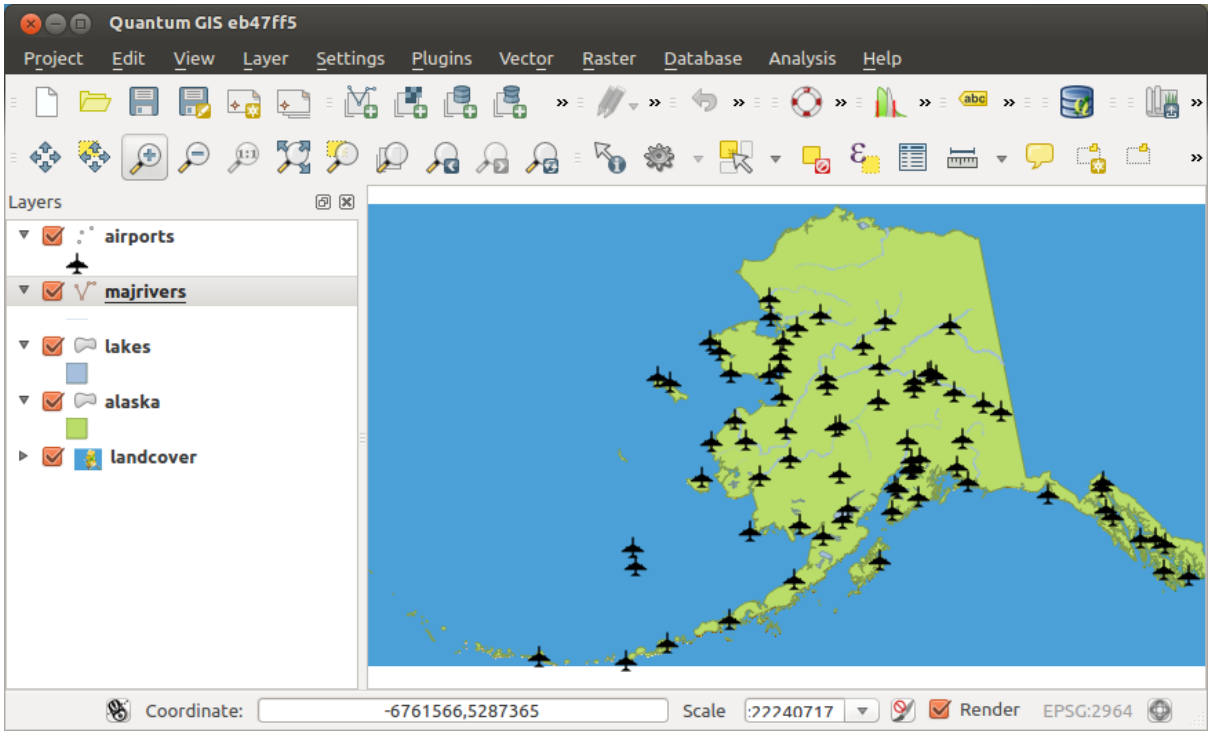


Figura24.24: Aeroportos do Alasca

1. Open the *Heatmap (Kernel Density Estimation)* algorithm from the QGIS *Interpolation* group
2. In the *Point layer* field, select `airports` from the list of point layers loaded in the current project.
3. Change the *Radius* to `1000000` meters.
4. Change the *Pixel size X* to `1000`. The *Pixel size Y*, *Rows* and *Columns* will be automatically updated.
5. Click on *Run* to create and load the airports heatmap (see Fig.24.26).

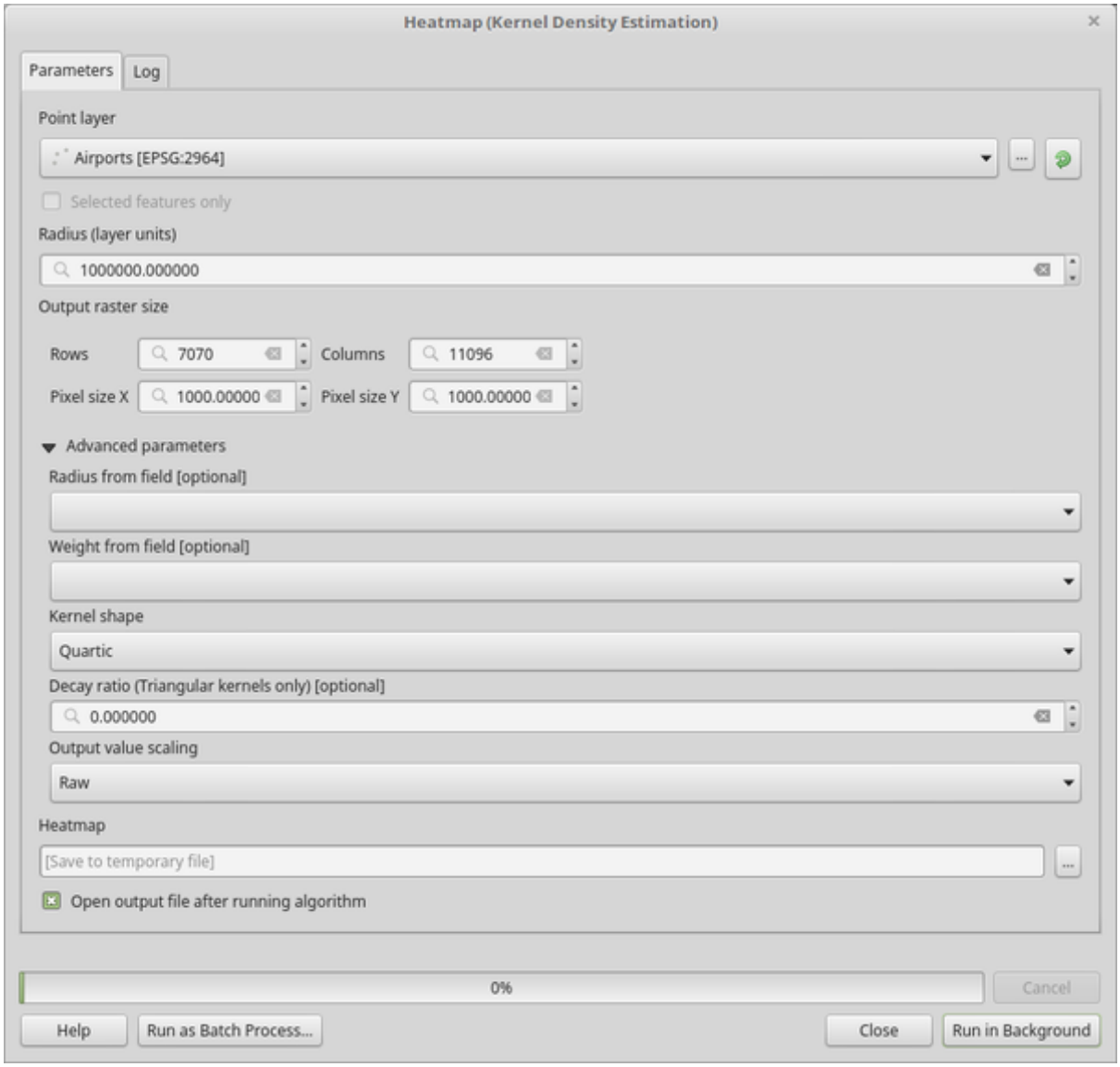


Figura24.25: The Heatmap Dialog

QGIS will generate the heatmap and add it to your map window. By default, the heatmap is shaded in greyscale, with lighter areas showing higher concentrations of airports. The heatmap can now be styled in QGIS to improve its appearance.

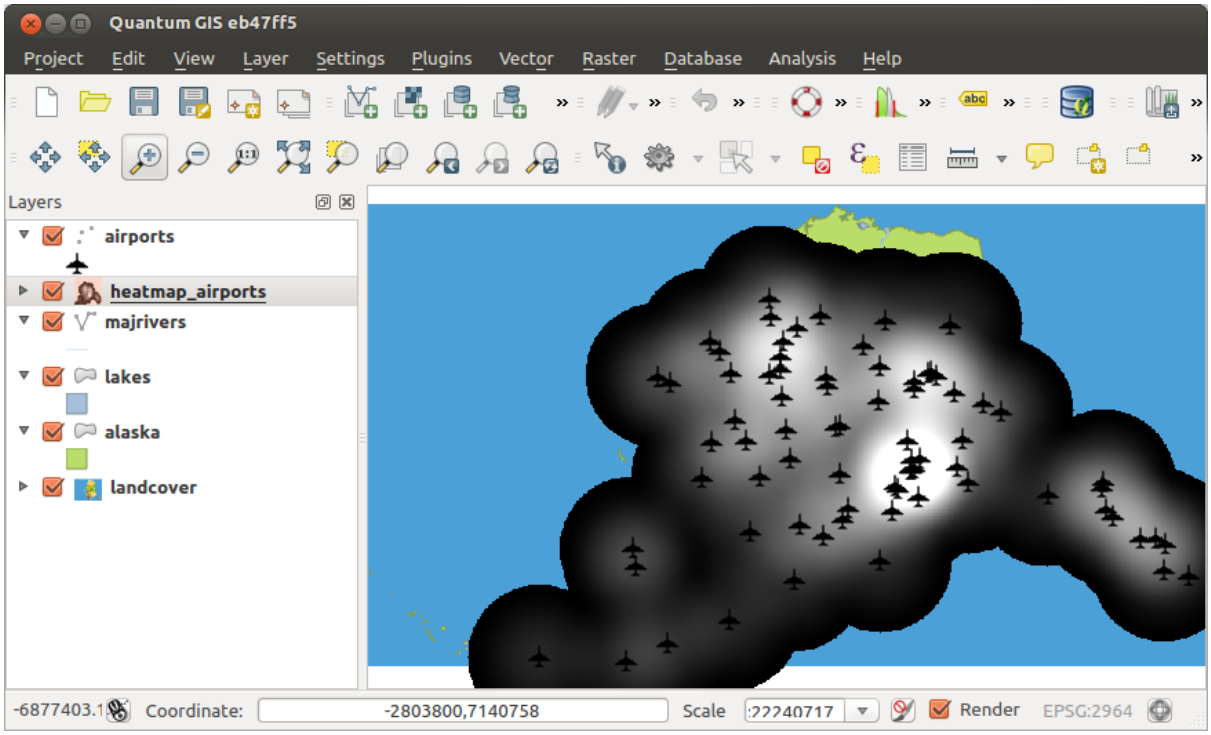


Figura24.26: O mapa de calor após o carregamento parece uma superfície cinza

1. Open the properties dialog of the `heatmap_airports` layer (select the layer `heatmap_airports`, open the context menu with the right mouse button and select *Properties*).
2. Select the *Symbology* tab.
3. Change the *Render type* to 'Singleband pseudocolor'.
4. Select a suitable *Color ramp* , for instance `YlOrRd`.
5. Click the *Classify* button.
6. Press *OK* to update the layer.

The final result is shown in Fig.24.27.

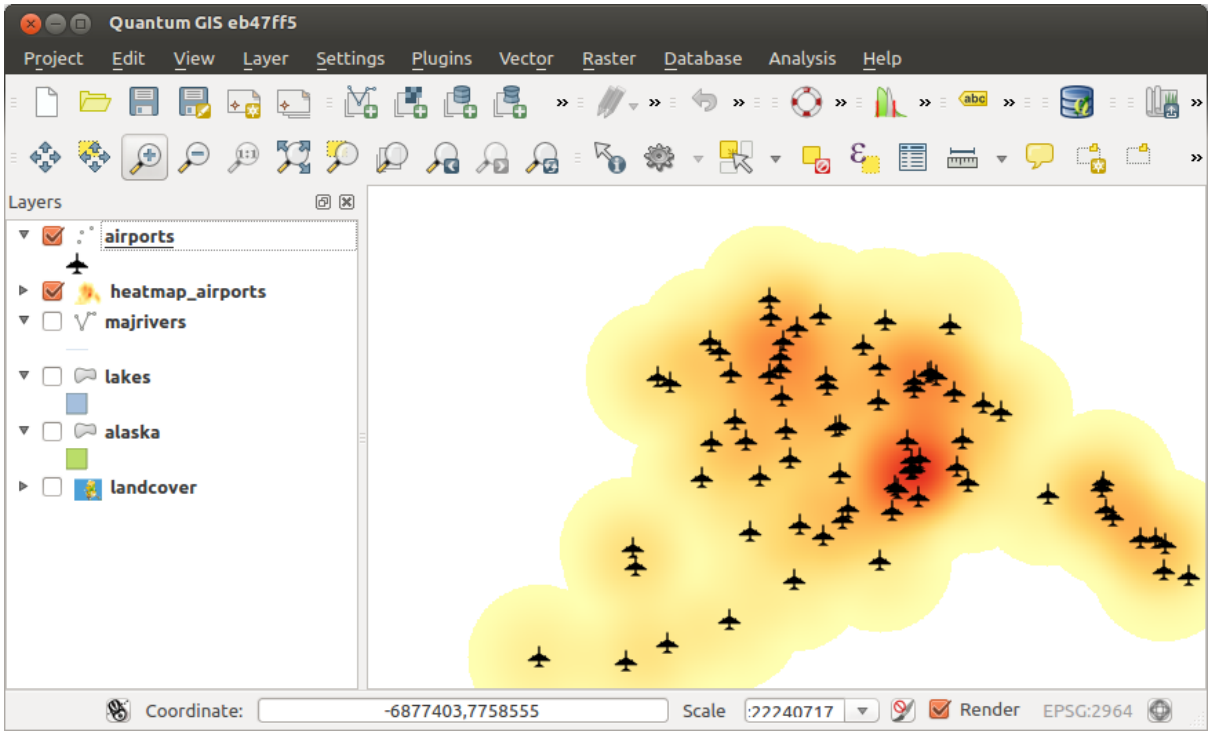


Figura24.27: Mapa de calor estilizado dos aeroportos do Alasca

Código Python

Algorithm ID: `qgis:heatmapkerneldensityestimation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Interpolação IDW

Generates an Inverse Distance Weighted (IDW) interpolation of a point vector layer.

Sample points are weighted during interpolation such that the influence of one point relative to another declines with distance from the unknown point you want to create.

O método de interpolação IDW também tem algumas desvantagens: a qualidade do resultado da interpolação pode diminuir, se a distribuição dos pontos de dados amostrais for desigual.

Furthermore, maximum and minimum values in the interpolated surface can only occur at sample data points.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada(s) de entrada	`INTERPOLATION_DATA`	[string]	Vector layer(s) and field(s) to use for the interpolation, coded in a string (see the `ParameterInterpolationData` class in InterpolationWidgets for more details). The following GUI elements are provided to compose the interpolation data string: • **Vector layer** [vector: any] • **Interpolation attribute** [tablefield: numeric]: Attribute to use in the interpolation • **Use Z-coordinate for interpolation** [boolean]: Uses the layer's stored Z values (Default: False) For each of the added layer-field combinations, a type can be chosen: • *Points*: Elevation sample locations containing the values to be interpolated (e.g., terrain height, Z). These points are the primary input for calculating the output surface. • *Structured lines*: Lines representing known continuous surface features that guide the terrain shape or structure (e.g., ridges, road centerlines, levees). They describe terrain direction while assuming continuity along the line. • *Break lines*: Lines representing known surface discontinuities or hard boundaries (e.g., cliffs, river banks, excavation edges, lake/pond outlines). They mark where the surface may change sharply. It is required if there are significant changes in elevation along the line. In the string, the layer-field elements are separated by `'::|::'`. The sub-elements of the layer-field elements are separated by `'::~::'`.
Coeficiente de distância P	`DISTANCE_COEFFICI`	[numeric: double] Padrão: 2.0	Define o coeficiente de distância para a interpolação. Mínimo: 0,0, máximo: 100.0.

continua na próxima página

Tabela 24.153 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Extensão (xmin, xmax, ymin, ymax)	`EXTENT`	[extensão]	Extent of the output raster layer. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Output raster size	`PIXEL_SIZE`	[numeric: double] Padrão: 0.1	Pixel size of the output raster layer in layer units. In the GUI, the size can be specified by the number of rows (`Number of rows`) / columns (`Number of columns`) **or** the pixel size(`Pixel Size X` / `Pixel Size Y`). Increasing the number of rows or columns will decrease the cell size and increase the file size of the output raster. The values in `Rows`, `Columns`, `Pixel Size X` and `Pixel Size Y` will be updated simultaneously - doubling the number of rows will double the number of columns, and the cell size will be halved. The extent of the output raster will remain the same (approximately).
Interpolado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Raster layer of interpolated values. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Interpolado	`OUTPUT`	[raster]	Camada raster de valores interpolados

Código Python

Algorithm ID: `qgis:idwinterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Densidade de linha

Calculates for each raster cell, the density measure of linear features within a circular neighbourhood. This measure is obtained by summing all the line segments intersecting the circular neighbourhood and dividing this sum by the area of such neighbourhood. A weighting factor can be applied to the line segments.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

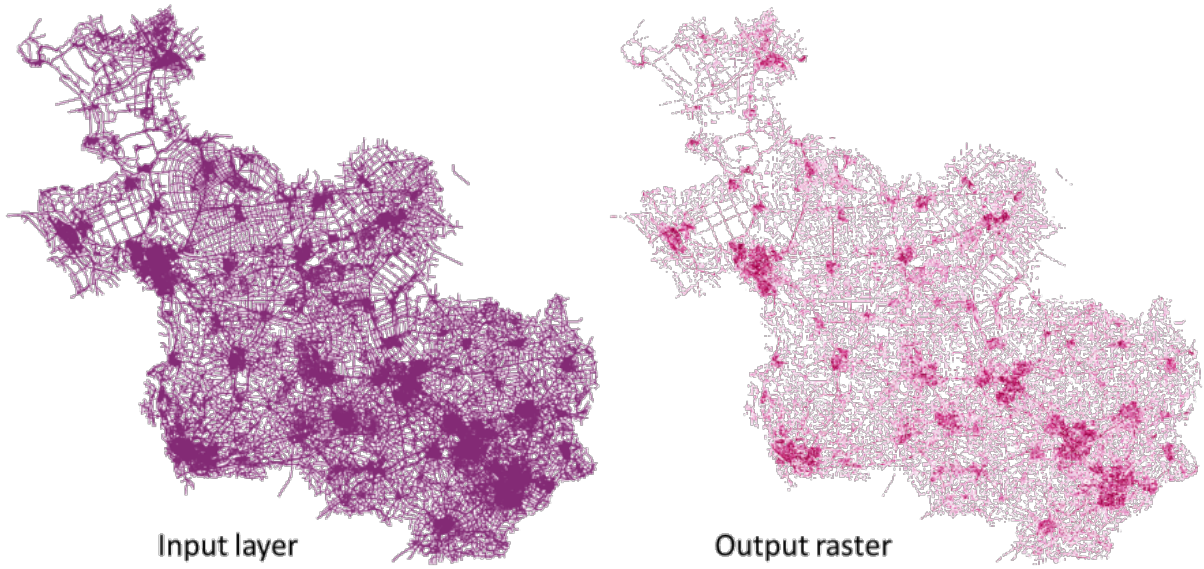


Figura24.28: Line density example. Input layer source: Roads Overijssel - The Netherlands (OSM).

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada de linha	`INPUT`	[vetor: linha]	Input vector layer containing line features
Campo de peso	`WEIGHT`	[campo de tabela: numérico]	Campo da camada que contém o fator de peso a ser usado durante o cálculo

continua na próxima página

Tabela 24.155 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Search Radius	`RADIUS`	[numeric: double] Padrão: 10.0	Radius of the circular neighbourhood. Units can be specified here.
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 10.0	Pixel size of the output raster layer in layer units. The raster has square pixels.
Line density raster	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	The output as a raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Line density raster	`OUTPUT`	[raster]	The output line density raster layer.

Código Python

Algorithm ID: `native:linedensity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

TIN Interpolation

Generates a Triangulated Irregular Network (TIN) interpolation of a point vector layer.

With the TIN method you can create a surface formed by triangles of nearest neighbor points. To do this, circumcircles around selected sample points are created and their intersections are connected to a network of non overlapping and as compact as possible triangles. The resulting surfaces are not smooth.

O algoritmo cria tanto a camada raster dos valores interpolados quanto a camada vetorial de linha com os limites da triangulação.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada(s) de entrada	`INTERPOLATION_DATA`	[string]	Vector layer(s) and field(s) to use for the interpolation, coded in a string (see the `ParameterInterpolationData` class in InterpolationWidgets for more details). The following GUI elements are provided to compose the interpolation data string: • **Vector layer** [vector: any] • **Interpolation attribute** [tablefield: numeric]: Attribute to use in the interpolation • **Use Z-coordinate for interpolation** [boolean]: Uses the layer's stored Z values (Default: False) For each of the added layer-field combinations, a type can be chosen: • *Points*: Elevation sample locations containing the values to be interpolated (e.g., terrain height, Z). These points are the primary input for calculating the output surface. • *Structured lines*: Lines representing known continuous surface features that guide the terrain shape or structure (e.g., ridges, road centerlines, levees). They describe terrain direction while assuming continuity along the line. • *Break lines*: Lines representing known surface discontinuities or hard boundaries (e.g., cliffs, river banks, excavation edges, lake/pond outlines). They mark where the surface may change sharply. It is required if there are significant changes in elevation along the line. In the string, the layer-field elements are separated by `'::|::'`. The sub-elements of the layer-field elements are separated by `'::~::'`.

continua na próxima página

Tabela 24.158 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Método de interpolação	`METHOD`	[enumeração] Padrão: 0	Defina o método de interpolação a ser usado. Um de: • *Linear* • *Clough-Toucher (cubic)*
Extensão (xmin, xmax, ymin, ymax)	`EXTENT`	[extensão]	Extent of the output raster layer. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Output raster size	`PIXEL_SIZE`	[numeric: double] Padrão: 0.1	Pixel size of the output raster layer in layer units. In the GUI, the size can be specified by the number of rows (`Number of rows`) / columns (`Number of columns`) **or** the pixel size(`Pixel Size X` / `Pixel Size Y`). Increasing the number of rows or columns will decrease the cell size and increase the file size of the output raster. The values in `Rows`, `Columns`, `Pixel Size X` and `Pixel Size Y` will be updated simultaneously - doubling the number of rows will double the number of columns, and the cell size will be halved. The extent of the output raster will remain the same (approximately).
Interpolado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	The output TIN interpolation as a raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…
Triangulação	`TRIANGULATION`	[vetor: linha] Default: `[Pular saída]`	The output TIN as a vector layer. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Interpolado	`OUTPUT`	[raster]	The output TIN interpolation as a raster layer
Triangulação	`TRIANGULATION`	[vetor: linha]	The output TIN as a vector layer.

Código Python

Algorithm ID: `qgis:tininterpolation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.9 Ferramentas de camada

Exportar informação(ões) de camada(s)

Cria uma camada de polígono com feições correspondentes à extensão da(s) camada(s) selecionada(s).

Detalhes adicionais da camada (SRC, nome do fornecedor, caminho do arquivo, nome da camada, filtro do subconjunto, resumo e atribuição) são anexados como atributos a cada recurso.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camadas de entrada*	`LAYERS`	[layer] [list]	Input layers to get information on.
Saída	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specification of the output layer with information. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Saída	`OUTPUT`	[vetor: polígono]	Camada vetorial de polígono mostrando a extensão das camadas de entrada e informações associadas nos atributos.

Código Python

Algoritmo ID: `native:exportlayersinformation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exportar para planilha

Exporta os atributos de uma seleção de camadas vetoriais para um documento de planilha ou opcionalmente os anexa a uma planilha existente como folhas adicionais.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camadas de entrada*	`LAYERS`	[vetor: qualquer] [lista]	Camadas vetoriais de entrada. A planilha de saída será composta de uma folha, para cada camada, que contém os atributos desta camada.
Utilizar apelido de campo como títulos de coluna	`USE_ALIAS`	[boleano] Padrão: Falso	Usar os apelidos de campo da tabela de atributos para a planilha.
Exportar valores formatados em vez de valores brutos*	`FORMATTED_VALUES`	[boleano] Padrão: Falso	Se “Verdade”, exporta os valores formatados e legíveis pelo ser humano (por exemplo, a partir de :ref:` um mapa de valor ou relação de valor <edit_widgets>`) para a planilha.
Sobrescrever a planilha existente*	`OVERWRITE`	[boleano] Padrão: Verdadeiro	Se a planilha especificada existir, definir esta opção como `Verdadeiro` substituirá a planilha existente. Se esta opção for `Falso` e a planilha existir, as camadas serão anexadas como planilhas adicionais.
Planilha de destino	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Output spreadsheet with a sheet for every layer. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Planilha de destino	`OUTPUT`	[arquivo]	Planilha com uma aba para cada camada.
Camadas dentro da planilha	`OUTPUT_LAYERS`	[lista]	A lista de abas adicionadas à planilha.

Código Python

Algoritmo ID: `native:exporttospreadsheet`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair extensão da camada

Gera uma camada vetorial com a caixa delimitadora mínima (retângulo com orientação N-S) que cobre todas as feições de entrada.

A camada de saída contém uma única caixa delimitadora para toda a camada de entrada.

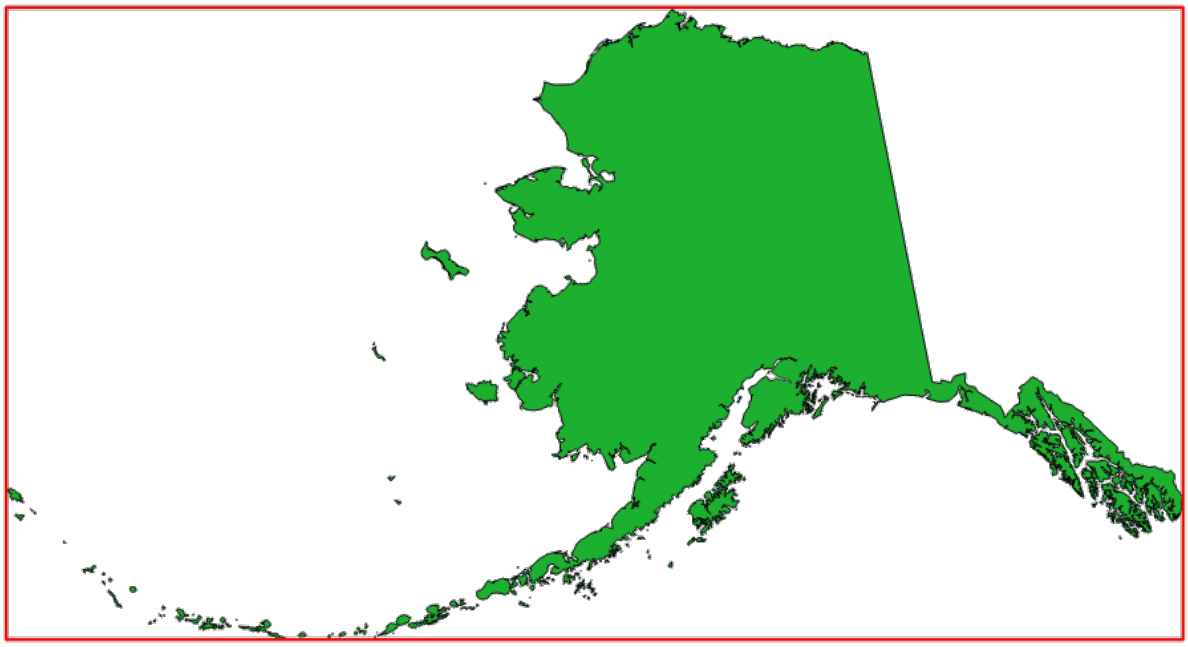


Figura24.29: Em vermelho a caixa delimitadora da camada de origem

Menu padrão: *Vetor ► Ferramentas de pesquisa*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada	`INPUT`	[camada]	Camada de entrada
Extensão	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the polygon vector layer for the output extent. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Extensão	`OUTPUT`	[vetor: polígono]	Camada vetorial de saída (polígono) com a extensão (caixa de delimitação mínima)

Código Python

Algorithm ID: `native:polygonfromlayerextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.10 Malha

Exportar contornos

Creates contours as a vector layer from a mesh scalar dataset.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de malha de entrada	`INPUT`	[malha]	A camada de malha para exportar dados
Grupos de dados	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	O intervalo de tempo a ser considerado • 0 — Tempo atual da tela • 1 — Data/hora definidas • 2 — Etapa de tempo do grupo de dados
Incremento entre curvas de nível Opcional	`INCREMENT`	[numeric: double] Padrão: Não definido	Intervalo entre os níveis gerados.
Curva de nível mínima Opcional	`MINIMUM`	[numeric: double] Padrão: Não definido	Starting level values of contours. No generated levels will be lower than this value.
Curva de nível máxima Opcional	`MAXIMUM`	[numeric: double] Padrão: Não definido	Os valores máximos de contornos, ou seja, nenhum nível gerado será maior do que este valor.
Lista de curvas de nível Opcional	`CONTOUR_LEVEL_LIS`	[string] Padrão: Não definido	List of desired levels of contours (separated by commas). If filled, the increment, minimum, and maximum fields will not be considered.
Sistema de coordenadas de saída Opcional	`CRS_OUTPUT`	[src]	Sistema de referência de coordenadas para atribuir à saída
Curvas de nível exportadas	`OUTPUT_LINES`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line layer representing the contours of the mesh layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Exportar contorno de polígonos	`OUTPUT_POLYGONS`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output polygon layer representing the contours of the mesh layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Curvas de nível exportadas	`OUTPUT_LINES`	[vetor: linha]	Camada de linha representando os contornos da camada de malha.
Exportar contorno de polígonos	`OUTPUT_POLYGONS`	[vetor: polígono]	Camada de polígono representando os contornos da camada de malha.

Código Python

Algorithm ID: `native:meshcontours`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Export cross section dataset values on lines from mesh

Extracts a mesh dataset's values from lines contained in a vector layer.

Each line is discretized with a resolution distance parameter for extraction of values on its vertices.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de malha de entrada	`INPUT`	[malha]	A camada de malha para exportar dados
Grupos de dados	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	O intervalo de tempo a ser considerado • 0 — Tempo atual da tela • 1 — Data/hora definidas • 2 — Etapa de tempo do grupo de dados
Linhas para exportação de dados	`INPUT_LINES`	[vetor: linha]	Lines where the data will be extracted from the dataset mesh
Resolução da segmentação de linha	`RESOLUTION`	[numeric: double] Padrão: 10.0	The distance between points on the lines where the data will be extracted from the dataset mesh. Units can be selected.
Contagem de dígitos para coordenadas	`COORDINATES_DIGITS`	[numeric: integer] Padrão: 2	Number of digits for rounding the coordinate values
Contagem de dígitos para o valor do conjunto de dados	`DATASET_DIGITS`	[numeric: integer] Padrão: 2	Number of digits for rounding the dataset values

continua na próxima página

Tabela 24.168 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Arquivo CSV de dados exportados	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo CSV de dados exportados	`OUTPUT`	[arquivo]	

Código Python

Algorithm ID: `native:meshexportcrosssection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Export mesh edges

Exports a mesh layer's edges to a line vector layer, with the dataset values on edges as attribute values.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de malha de entrada	`INPUT`	[malha]	A camada de malha para exportar dados
Grupos de dados	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	O intervalo de tempo a ser considerado • 0 — Tempo atual da tela • 1 — Data/hora definidas • 2 — Etapa de tempo do grupo de dados
Sistema de coordenadas de saída Opcional	`CRS_OUTPUT`	[src]	Sistema de referência de coordenadas para atribuir à saída

continua na próxima página

Tabela 24.170 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Opção vetorial de exportação	`VECTOR_OPTION`	[enumeração]	Tipo coordenado de exportação de valor vetorial. • 0 — Cartesiano (x,y) • 1 — Polar (magnitude, grau) • 2 — Cartesiano e polar
Camada vetorial de saída*	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specification of the output file. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada vetorial de saída*	`OUTPUT`	[vetor: linha]	Output vector line layer containing the edges of the input mesh layer with associated dataset values

Código Python

Algorithm ID: `native:exportmeshedges`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exportar faces de malha

Exports a mesh layer's faces to a polygon vector layer, with the dataset values on faces as attribute values.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de malha de entrada	`INPUT`	[malha]	A camada de malha para exportar dados
Grupos de dados	`DATASET_GROUPS`	[layer] [list]	The dataset groups

continua na próxima página

Tabela 24.172 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Dataset time	`DATASET_TIME`	[datetime]	O intervalo de tempo a ser considerado • 0 — Tempo atual da tela • 1 — Data/hora definidas • 2 — Etapa de tempo do grupo de dados
Sistema de coordenadas de saída Opcional	`CRS_OUTPUT`	[src]	Sistema de referência de coordenadas para atribuir à saída
Opção vetorial de exportação	`VECTOR_OPTION`	[enumeração]	Tipo coordenado de exportação de valor vetorial. • 0 — Cartesiano (x,y) • 1 — Polar (magnitude, grau) • 2 — Cartesiano e polar
Camada vetorial de saída*	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specification of the output file. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada vetorial de saída*	`OUTPUT`	[vetor: polígono]	Output vector polygon layer containing the faces of the input mesh layer with associated dataset values

Código Python

Algorithm ID: `native:exportmeshfaces`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Export mesh on grid

Exports a mesh layer's dataset values to a gridded point vector layer, with the dataset values on this point as attribute values.

For data on volume (3D stacked dataset values), the exported dataset values are averaged on faces using the method defined in *the mesh layer properties* (default is Multi level averaging method). 1D meshes are not supported.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de malha de entrada	`INPUT`	[malha]	A camada de malha para exportar dados
Grupos de dados	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	O intervalo de tempo a ser considerado • 0 — Tempo atual da tela • 1 — Data/hora definidas • 2 — Etapa de tempo do grupo de dados
Extensão Opcional	`EXTENT`	[extensão]	Especifique a extensão espacial em que os dados devem ser processados. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Espaçamento entre as grades Opcional	`GRID_SPACING`	[numeric: double] Padrão: 10.0	Espaçamento entre os pontos de amostra a usar
Sistema de coordenadas de saída Opcional	`CRS_OUTPUT`	[src]	Sistema de referência de coordenadas para atribuir à saída
Opção vetorial de exportação	`VECTOR_OPTION`	[enumeração]	Tipo coordenado de exportação de valor vetorial. • 0 — Cartesiano (x,y) • 1 — Polar (magnitude, grau) • 2 — Cartesiano e polar

continua na próxima página

Tabela 24.174 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Camada vetorial de saída*	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specification of the output file. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada vetorial de saída*	`OUTPUT`	[vetor: ponto]	Output vector point layer with dataset values computed from the overlaid face.

Código Python

Algorithm ID: `native:exportmeshongrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exportar vértices da malha

Exports a mesh layer's vertices to a point vector layer, with the dataset values on vertices as attribute values.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de malha de entrada	`INPUT`	[malha]	A camada de malha para exportar dados
Grupos de dados	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	O intervalo de tempo a ser considerado • 0 — Tempo atual da tela • 1 — Data/hora definidas • 2 — Etapa de tempo do grupo de dados
Sistema de coordenadas de saída Opcional	`CRS_OUTPUT`	[src]	Sistema de referência de coordenadas para atribuir à saída

continua na próxima página

Tabela 24.176 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Opção vetorial de exportação	`VECTOR_OPTION`	[enumeração]	Tipo coordenado de exportação de valor vetorial. • 0 — Cartesiano (x,y) • 1 — Polar (magnitude, grau) • 2 — Cartesiano e polar
Camada vetorial de saída*	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specification of the output file. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada vetorial de saída*	`OUTPUT`	[vetor: ponto]	Output vector point layer containing the vertices of the input mesh layer with associated dataset values

Código Python

Algorithm ID: `native:exportmeshvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Export time series values from points of a mesh dataset

Extracts a mesh dataset's time series values from points contained in a vector layer.

If the time step is kept to its default value (0 hours), the time step used is the one of the two first datasets of the first selected dataset group.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de malha de entrada	`INPUT`	[malha]	A camada de malha para extrair dados
Grupos de dados	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Tempo de início	`STARTING_TIME`	[datetime]	O início do intervalo de tempo a ser considerado • 0 — Tempo atual da tela • 1 — Data/hora definidas • 2 — Etapa de tempo do grupo de dados
Tempo de conclusão	`FINISHING_TIME`	[datetime]	O fim do intervalo de tempo a ser considerado • 0 — Tempo atual da tela • 1 — Data/hora definidas • 2 — Etapa de tempo do grupo de dados
Time step (hours) Opcional	`TIME_STEP`	[numeric: double] Padrão: 0	Time between two consecutive steps to extract. Keep `0` to use time step of the first selected dataset group.
Pontos para exportação de dados*	`INPUT_POINTS`	[vetor: ponto]	Vector layer containing points where the data will be extracted from the dataset mesh
Contagem de dígitos para coordenadas	`COORDINATES_DIGITS`	[numeric: integer]	Number of digits for rounding coordinate values Padrão: 2
Contagem de dígitos para o valor do conjunto de dados	`DATASET_DIGITS`	[numeric: integer] Padrão: 2	Number of digits for rounding dataset values
Arquivo CSV de dados exportados	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo CSV de dados exportados	`OUTPUT`	[arquivo]	`.CSV` file containing the mesh dataset time series values at the overlaying point features

Código Python

Algorithm ID: `native:meshexporttimeseries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Rasterize mesh dataset

Creates a raster layer from a mesh dataset.

For data on volume (3D stacked dataset values), the exported dataset values are averaged on faces using the method defined in *the mesh layer properties* (default is Multi level averaging method). 1D meshes are not supported.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de malha de entrada	`INPUT`	[malha]	A camada de malha para exportar dados
Grupos de dados	`DATASET_GROUPS`	[layer] [list]	The dataset groups
Dataset time	`DATASET_TIME`	[datetime]	O intervalo de tempo a ser considerado • 0 — Tempo atual da tela • 1 — Data/hora definidas • 2 — Etapa de tempo do grupo de dados
Extensão Opcional	`EXTENT`	[extensão]	Especifique a extensão espacial em que os dados devem ser processados. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Tamanho do pixel da camada raster de saída.

continua na próxima página

Tabela 24.180 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Sistema de coordenadas de saída Opcional	`CRS_OUTPUT`	[src]	Sistema de referência de coordenadas para atribuir à saída
Camada raster de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada raster de saída	`OUTPUT`	[raster]	Camada raster de saída com valores de conjuntos de dados computados a partir da camada de malha.

Código Python

Algorithm ID: `native:meshrasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Surface to Polygon

Exports a polygon file containing the boundary of a mesh layer. The resulting polygon may contain holes and may be a multi-part polygon.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de malha de entrada	`INPUT`	[malha]	The input mesh layer whose boundary will be exported as a polygon
Sistema de coordenadas de saída Opcional	`OUTPUT_CRS`	[src]	The coordinate reference system (CRS) for the output polygon layer. If not specified, the CRS of the input mesh layer will be used.
Camada vetorial de saída*	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specification of the output file. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada vetorial de saída*	`OUTPUT`	[vetor: polígono]	The resulting polygon layer containing the boundary of the mesh layer. The polygon may contain holes and may be multi-part.

Código Python

Algorithm ID: `native:surfacetopolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

TIN mesh creation

Creates a TIN mesh layer from vector layers. The TIN mesh is created using a Delaunay triangulation.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`SOURCE_DATA`	[vetor: qualquer] [lista]	Camadas vetoriais a serem combinadas para gerar a camada de malha
Camada vetorial	GUI ONLY	[vetor: qualquer] [lista]	Um seletor para as camadas vetoriais a serem combinadas para gerar a camada de malha
Valor no vértice	GUI ONLY	[campo da tabela: qualquer]	Um seletor de campo para usar na camada selecionada. A cada vértice é atribuído o valor correspondente de sua feição original.
Utilizar a coordenada Z para valor no vértice	GUI ONLY	[boleano] Padrão: Falso	If checked, the Z value of vector layer points or polygons/lines vertices will be used to assign the Z value of the vertex mesh layer. Only available if the input layers are 3D.
Formato de Saída	`MESH_FORMAT`	[enumeração] Padrão: 2DM	Formato de saída da camada gerada • 0 — 2DM • 1 — SELAFIN • 2 — PLY • 3 — Ugrid • 4 — Mike21
Sistema de coordenadas de saída Opcional	`CRS_OUTPUT`	[src]	Sistema de referência de coordenadas para atribuir à saída
Arquivo de saída	`OUTPUT_MESH`	[malha] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo de saída	`OUTPUT_MESH`	[malha]	Output mesh layer with dataset values computed from the vector layers.

Código Python

Algorithm ID: `native:tinmeshcreation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.11 Metadata tools

Add history metadata

Adds a new history entry to the layer's metadata.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The input layer (vector, raster, etc.) to which the history entry will be added.
History entry	`HISTORY`	[string]	The text to be appended as a new entry in the layer's history metadata. This will be added to any existing history entries.

Outputs

Label	Name	Type	Description
Output	`OUTPUT`	[same as input]	The resulting layer with the updated history in its **Metadata properties**.

Python code

Algorithm ID: `native:addhistorymetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Copy Layer Metadata

Copies metadata from a source layer to a target layer. Any existing metadata in the target layer will be replaced with the metadata from the source layer. This includes all metadata fields, such as history, abstract, and other properties.

Parameters

Label	Name	Type	Description
Source layer	`SOURCE`	[layer]	The layer from which metadata will be copied.
Target layer	`TARGET`	[layer]	The layer to which metadata will be pasted. Any existing metadata in this layer will be replaced.
Save metadata as default	`DEFAULT`	[boolean] Default: False	If checked, the metadata information will be saved with the layer, hence available by default in subsequent projects.

Outputs

Label	Name	Type	Description
Output layer	`OUTPUT`	[layer]	The target layer with the metadata replaced by the metadata from the source layer. This includes all metadata fields, such as history, abstract, and other properties.

Python code

Algorithm ID: `native:copylayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Export Layer Metadata

Exports the metadata of a layer to a QMD file.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The layer whose metadata will be exported.
Output	`OUTPUT`	[file] Default: `[Save to temporary file]`	Specification of the output file. *One of*: • Save to a Temporary File • Save to File…

Outputs

Label	Name	Type	Description
Output file	`OUTPUT`	[file]	The `.qmd` file containing the exported metadata. This file can be used to import metadata into another layer.

Python code

Algorithm ID: `native:exportlayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Set Layer Metadata

Applies metadata to a layer from a `.qmd` file.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The layer to which the metadata will be applied.
Metadata file	`METADATA`	[file]	The `.qmd` file containing the metadata to be applied.
Save metadata as default	`DEFAULT`	[boolean] Default: False	If checked, the metadata information will be saved with the layer, hence available by default in subsequent projects.

Outputs

Label	Name	Type	Description
Output layer	`OUTPUT`	[same as input]	The input layer with the metadata replaced by the metadata from the `.qmd` file.

Python code

Algorithm ID: `native:setlayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Set Metadata Fields

Sets various metadata fields for a layer.

Parameters

Label	Name	Type	Description
Layer	`INPUT`	[layer]	The layer whose metadata fields will be updated
Identifier Optional	`IDENTIFIER`	[string]	A unique identifier for the layer
Parent identifier Optional	`PARENT_IDENTIFIER`	[string]	The identifier of the parent resource, if applicable
Title Optional	`TITLE`	[string]	The title of the layer
Type Optional	`TYPE`	[string]	The type of data stored in the layer
Language Optional	`LANGUAGE`	[string]	The language of the metadata
Encoding Optional	`ENCODING`	[string]	The character encoding used in the metadata
Abstract Optional	`ABSTRACT`	[string]	A brief description or abstract of the layer
Coordinate reference system Optional	`CRS`	[crs]	The coordinate reference system of the layer
Fees Optional	`FEES`	[string]	Information about any fees associated with accessing the layer
Ignore empty fields	`IGNORE_EMPTY`	[boolean] Default: False	If checked, no update will be done to metadata fields that are not filled

Outputs

Label	Name	Type	Description
Output layer	`OUTPUT`	[same as input]	The input layer with the specified metadata fields updated.

Python code

Algorithm ID: `native:setmetadatafields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

Update Layer Metadata

Copies all non-empty metadata fields from a source layer to a target layer. Leaves empty input fields unchanged in the target.

Parameters

Label	Name	Type	Description
Source layer	`SOURCE`	[layer]	The layer from which non-empty metadata fields will be copied
Target layer	`TARGET`	[layer]	The layer to which non-empty metadata fields will be pasted. Empty fields in the source layer will not overwrite existing metadata in the target layer.

Outputs

Label	Name	Type	Description
Target layer	`OUTPUT`	[layer]	The target layer with updated metadata.

Python code

Algorithm ID: `native:updatelayermetadata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

The *algorithm id* is displayed when you hover over the algorithm in the Processing Toolbox. The *parameter dictionary* provides the parameter NAMEs and values. See *Usando os algoritmos do processamento a partir do Terminal Python.* for details on how to run processing algorithms from the Python console.

24.1.12 Ferramentas do modelador

Aviso

These tools are only available in the model designer. They are not available in the Processing Toolbox.

Calculate expression

It calculates the result of a QGIS expression and eliminates the need to use the same expression multiple times throughout a model when the same result is required more than once. Additionally, it enables use cases that would otherwise not be possible. For instance, you can generate a timestamp value once and use it multiple times within the model, if the timestamp were recalculated every time, the values would vary during the model's runtime.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Input	`INPUT`	[expressão]	Expression to calculate

Saídas

Etiqueta	Nome	Tipo	Descrição
Valor	`OUTPUT`	[Result Value]	Calculated result value, the data type of the output will vary based on the specific expression used in the algorithm.

Código Python

Algorithm ID: `native:calculateexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Ramificação condicional

Adiciona uma ramificação condicional em um modelo, permitindo que partes do modelo sejam executadas com base no resultado de uma avaliação de expressão. Principalmente usando dependências de ferramentas para controlar o fluxo de um modelo.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Campo	`BRANCH`	[string]	Nome da condição
Campo	`CONDITION`	[expressão]	Expressão para avaliar

Saídas

Nenhum.

Código Python

ID do algoritmo: `native:condition`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar diretório

Cria um novo diretório em um sistema de arquivos. Os diretórios serão criados recursivamente, criando todos os diretórios pai necessários para construir o caminho completo do diretório especificado. Nenhum erro será gerado se o diretório já existir.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Caminho de diretório	`PATH`	[string]	Caminho da pasta a ser criada

Saídas

Etiqueta	Nome	Tipo	Descrição
Saída	`OUTPUT`	[pasta]	Pasta criada

Código Python

ID do algoritmo: `native:createdirectory`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Filtro de feição

Filtra recursos da camada de entrada e os redireciona para uma ou várias saídas. Se você não conhece nenhum nome de atributo que seja comum a todas as camadas de entrada possíveis, a filtragem só é possível na geometria do recurso e nos mecanismos gerais de registro, como `$id` e `uuid`.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada de entrada.
Saídas e filtros (um ou mais)	`OUTPUT_<name of the filter>`	[o mesmo que entrada]	As camadas de saída com filtros (tantos quantos filtros houver).

Saídas

Etiqueta	Nome	Tipo	Descrição
Saída (um ou mais)	`native:filter_1: of filter>`	[o mesmo que entrada]	As camadas de saída com recursos filtrados (tantos quantos filtros existem).

Código Python

ID do algoritmo: `native:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Filtrar camadas por tipo

Filtra camadas por tipo. As camadas de entrada serão direcionadas para diferentes saídas com base no fato de serem uma camada vetorial ou raster.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[camada]	Camada de Mapa Genérico

Saídas

Etiqueta	Nome	Tipo	Descrição
Feições de vetor Opcional	`VECTOR`	[vetor]	Uma Camada Vetorial da entrada, se compatível
Camada raster Opcional	`RASTER`	[raster]	Uma Camada Raster da entrada, se compatível

Código Python

ID do algoritmo: `native:filterlayersbytype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Carregar camada no projeto

Carrega uma camada para o projeto atual.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada	`INPUT`	[camada]	Camada para carregar na legenda
Nome da camada carregada	`NAME`	[string]	Nome da camada carregada

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada	`OUTPUT`	[o mesmo que entrada]	A camada carregada (renomeada)

Código Python

ID do algoritmo: `native:loadlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Gerar exceção

Gera uma exceção e cancela a execução de um modelo. A mensagem de exceção pode ser customizada e, opcionalmente, uma condição baseada em expressão pode ser especificada. Se uma condição de expressão for usada, a exceção só será gerada se o resultado da expressão for verdadeiro. Um resultado falso indica que nenhuma exceção será gerada e a execução do modelo pode continuar ininterruptamente.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Mensagem	`MESSAGE`	[string]	Mensagem a ser exibida
condição Opcional	`CONDITION`	[expressão]	Expressão para avaliar se verdadeira

Saídas

Uma mensagem no painel de registro.

Código Python

ID do algoritmo: `native:raiseexception`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Raise message

Raises an information message in the log. The message can be customized, and optionally an expression based condition can be specified. If an expression condition is used, then the message will only be logged if the expression result is true. A false result indicates that no message will be logged.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Information message	`MESSAGE`	[string]	Mensagem a ser exibida
condição Opcional	`CONDITION`	[expressão]	Expressão para avaliar se verdadeira

Saídas

Uma mensagem no painel de registro.

Código Python

Algorithm ID: `native:raisemessage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Aumentar aviso

Gera uma mensagem de aviso no log. A mensagem de aviso pode ser personalizada e, opcionalmente, uma condição baseada em expressão pode ser especificada. Se uma condição de expressão for usada, o aviso só será registrado se o resultado da expressão for verdadeiro. Um resultado falso indica que nenhum aviso será registrado.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Mensagem	`MESSAGE`	[string]	Mensagem a ser exibida
condição Opcional	`CONDITION`	[expressão]	Expressão para avaliar se verdadeira

Saídas

Uma mensagem no painel de registro.

Código Python

ID do algoritmo: `native:raisewarning`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Renomear camada

Renomeia uma camada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada	`INPUT`	[camada]	Camada para renomear
Novo nome	`NAME`	[string]	O novo nome da camada

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada	`OUTPUT`	[o mesmo que entrada]	A camada de saída (renomeada)

Código Python

ID do algoritmo: `native:renamelayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Salvar log em arquivo

Salva o log de execução do modelo em um arquivo. Opcionalmente, o log pode ser salvo em uma versão formatada em HTML.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Usar HTML	`USE_HTML`	[Boleano] Padrão: Falso	Usar formatação HTML

Saídas

Etiqueta	Nome	Tipo	Descrição
Arquivo	`OUTPUT`	[string]	Destino do registro

Código Python

ID do algoritmo: `native:savelog`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Definir variável do projeto

Define uma variável de expressão para o projeto atual.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Nome variável	`NAME`	[string]	Nome da variável
Valor variável	`VALUE`	[string]	Valor a ser armazenado

Saídas

Nenhum.

Código Python

ID do algoritmo: `native:setprojectvariable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Concatenação de strings

Concatena duas strings em uma única no Modelador de Processamento.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Entrada 1	`INPUT_1`	[string]	Primeira sequência
Entrada 2	`INPUT_2`	[string]	Segunda sequência

Saídas

Etiqueta	Nome	Tipo	Descrição
Concatenação	`CONCATENATION`	[string]	A string concatenada

Código Python

Algorithm ID: `native:stringconcatenation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Buffer de distância variável

Aviso

Este algoritmo está obsoleto e pode ser removido a qualquer momento. Prefira usar o algoritmo *Buffer*.

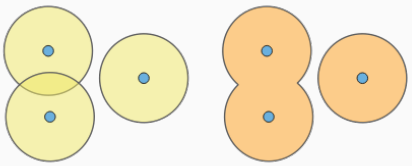
Calcula uma área de buffer para todos os recursos em uma camada de entrada.

O tamanho do buffer para um determinado recurso é definido por um atributo, portanto, permite que recursos diferentes tenham tamanhos de buffer diferentes.

Ver também

Buffer

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
Campo de distância	`DISTANCE`	[campo de tabela: numérico]	Atributo para o raio de distância do buffer
Segmentos	`SEGMENTS`	[numeric: integer] Padrão: *5*	Controla o número de segmentos usadas para aproximar um quarte de círculo quando criando desvios redondos.
Dissolver resultado	`DISSOLVE`	[boleano] Padrão: *Falso*	Escolha dissolver o buffer final, resultando em um único recurso cobrindo todos os recursos de entrada.  Figura24.30: Tampão normal e dissolvido

continua na próxima página

Tabela 24.220 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
End cap style	`END_CAP_STYLE`	[enumeração] Padrão: redondo	Controla como os finais de linha são tratados no buffer. Figura24.31: Estilos de cobertura redonda, plana e quadrada.
Join style	`JOIN_STYLE`	[enumeração] Padrão: redondo	Especifica se as junções arredondadas, de meia-esquadria ou chanfradas devem ser usadas ao deslocar cantos em uma linha.
Miter limit	`MITER_LIMIT`	[numeric: double] Padrão: 2.0	Only applicable for mitered join styles, and controls the maximum distance from the offset curve to use when creating a mitered join.

Saídas

Etiqueta	Nome	Tipo	Descrição
Buffer	`OUTPUT`	[vetor: polígono]	Camada de vetor de polígono de buffer.

Código Python

Algorithm ID: `qgis:variabledistancebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.13 Análise de rede

Área de serviço (a partir da camada)

Returns all the edges or parts of edges of a network that can be reached within a distance or a time, starting from a point layer. This allows evaluation of accessibility within a network, e.g., what are the places I can navigate to on a road network without spending cost greater than a given value (the cost can be distance or time).

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada vetorial representando a rede	`INPUT`	[vetor: linha]	Camada de vetor de linha representando a rede a ser coberta
Camada vetorial com pontos iniciais	`START_POINTS`	[vetor: ponto]	Camada vetorial de pontos cujas feições são usadas como pontos de partida para gerar as áreas de serviço
Tipo de caminho a ser calculado	`STRATEGY`	[enumeração] Padrão: 0	O tipo de caminho a ser calculado. Um de: • 0 — Mais curto • 1 — Mais rápido
Custo da viagem (distância para "Mais curto", tempo para "Mais rápido")	`TRAVEL_COST`	[numeric: double] Padrão: 0.0	The value is estimated as a distance (in the network layer units) when looking for the *Shortest* path and as time (in hours) for the *Fastest* path.
Service area (lines)	`OUTPUT_LINES`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line layer for the service area. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Service area (boundary nodes)	`OUTPUT`	[vetor: ponto] Default: `[Pular saída]`	Specify the output point layer for the service area boundary nodes. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Non-routable features	`OUTPUT_NON_ROUTAB`	[vetor: ponto] Default: `[Pular saída]`	Specify the output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

continua na próxima página

Tabela 24.222 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Campo de direção Opcional	`DIRECTION_FIELD`	[campo da tabela: qualquer]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used.
Valor para a direção direta Opcional	`VALUE_FORWARD`	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a forward direction
Valor para a direção para trás Opcional	`VALUE_BACKWARD`	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a backward direction
Valor para ambas as direções Opcional	`VALUE_BOTH`	[string] Padrão: '' (string vazia)	Value set in the direction field to identify bidirectional edges
Direção padrão	`DEFAULT_DIRECTIO`	[enumeração] Padrão: 2	Se uma feição não tiver valor definido no campo de direção ou se nenhum campo de direção estiver definido, esse valor de direção será usado. Um de: • 0 — Direção para frente • 1 — Sentido inverso • 2 — Ambas as direções
Campo de velocidade Opcional	`SPEED_FIELD`	[campo de tabela: numérico]	Field providing the speed value (in `km/h`) for the edges of the network when looking for the fastest path. Se uma feição não tiver um valor nesse campo ou nenhum campo estiver definido, o valor da velocidade padrão (fornecido com o parâmetro `Velocidade padrão`) será usado.
Velocidade padrão (km/h)	`DEFAULT_SPEED`	[numeric: double] Padrão: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field
** Tolerância de Topologia**	`TOLERANCE`	[numeric: double] Padrão: 0.0	Duas linhas com nós mais próximos que a tolerância especificada são consideradas conectadas

Include upper/lower bound points	`INCLUDE_BOUNDS`	[boleano] Padrão: Falso	Creates a point layer output with two points for each edge at the boundaries of the service area. One point is the start of that edge, the other is the end.
Maximum point distance from network Opcional	`POINT_TOLERANC`	[numeric: double] Padrão: Não definido	Specifies an optional limit on the distance from the points to the network layer. If a point is further from the network than this distance it will be treated as non-routable. If not set, endpoints will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are.

Saídas

Rótulo	Nome	Tipo	Descrição
Service area (boundary nodes)	`OUTPUT`	[vetor: ponto]	The output point layer with the service area boundary nodes.
Service area (lines)	`OUTPUT_LINES`	[vetor: linha]	Line layer representing the parts of the network that can be serviced by the start points, for the given cost.
Non routable features	`OUTPUT_NON_ROUTAB`	[vetor: ponto]	An optional output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer).

Código Python

Algorithm ID: `native:serviceareafromlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Área de serviço (a partir do ponto)

Returns all the edges or parts of edges of a network that can be reached within a given distance or time, starting from a point feature. This allows the evaluation of accessibility within a network, e.g., what are the places I can navigate to on a road network without spending a cost greater than a given value (the cost can be distance or time).

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Vector layer representing the network	`INPUT`	[vetor: linha]	Camada de vetor de linha representando a rede a ser coberta
Tipo de caminho a ser calculado	`STRATEGY`	[enumeração] Padrão: 0	O tipo de caminho a ser calculado. Um de: • 0 — Mais curto • 1 — Mais rápido
Start point	`START_POINT`	[coordenadas]	Coordenada do ponto para calcular a área de serviço ao redor. Press the … button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.
Custo da viagem (distância para "Mais curto", tempo para "Mais rápido")	`TRAVEL_COST`	[numeric: double] Padrão: 0.0	The value is estimated as a distance (in the network layer units) when looking for the *Shortest* path and as time (in hours) for the *Fastest* path.
Service area (lines) Opcional	`OUTPUT_LINES`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line layer for the service area. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Service area (boundary nodes) Opcional	`OUTPUT`	[vetor: ponto] Default: `[Pular saída]`	Specify the output point layer for the service area boundary nodes. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Campo de direção Opcional	`DIRECTION_FIELD`	[campo da tabela: qualquer]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used.
Valor para a direção direta Opcional	`VALUE_FORWARD`	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a forward direction
Valor para a direção para trás Opcional	`VALUE_BACKWARD`	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a backward direction
Valor para ambas as direções Opcional	`VALUE_BOTH`	[string] Padrão: '' (string vazia)	Value set in the direction field to identify bidirectional edges
Direção padrão	`DEFAULT_DIRECTIO`	[enumeração] Padrão: 2	Se uma feição não tiver valor definido no campo de direção ou se nenhum campo de direção estiver definido, esse valor de direção será usado. Um de: • 0 — Direção para frente • 1 — Sentido inverso • 2 — Ambas as direções
Campo de velocidade Opcional	`SPEED_FIELD`	[campo de tabela: numérico]	Field providing the speed value (in `km/h`) for the edges of the network when looking for the fastest path. Se uma feição não tiver um valor nesse campo ou nenhum campo estiver definido, o valor da velocidade padrão (fornecido com o parâmetro `Velocidade padrão`) será usado.
Velocidade padrão (km/h)	`DEFAULT_SPEED`	[numeric: double] Padrão: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field
** Tolerância de Topologia**	`TOLERANCE`	[numeric: double] Padrão: 0.0	Duas linhas com nós mais próximos que a tolerância especificada são consideradas conectadas

Maximum point distance from network Opcional	POINT_TOLERANC	[numeric: double] Padrão: 0.0	Specifies an optional limit on the distance from the start point to the network layer. If the point is further from the network than this distance an error will be raised. If not set, the point will be snapped to the nearest point on the network layer, regardless of how far away from the network it actually is.
Include upper/lower bound points	IN-CLUDE_BOUNDS	[boleano] Padrão: Falso	Creates a point layer output with two points for each edge at the boundaries of the service area. One point is the start of that edge, the other is the end.

Saídas

Rótulo	Nome	Tipo	Descrição
Service area (boundary nodes)	OUTPUT	[vetor: ponto]	The output point layer with the service area boundary nodes.
Service area (lines)	OUTPUT_LINES	[vetor: linha]	Camada de linha representando as partes da rede que podem ser atendidas pelo ponto inicial, pelo custo determinado.

Código Python

Algorithm ID: `native:serviceareafrompoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Caminho mais curto (camada a ponto)

Calcula as rotas ideais (mais curtas ou mais rápidas) de vários pontos iniciais definidos por uma camada vetorial e um determinado ponto final.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada vetorial representando a rede	`INPUT`	[vetor: linha]	Camada de vetor de linha representando a rede a ser coberta
Tipo de caminho a ser calculado	`STRATEGY`	[enumeração] Padrão: 0	O tipo de caminho a ser calculado. Um de: • 0 — Mais curto • 1 — Mais rápido
Camada vetorial com pontos iniciais	`START_POINTS`	[vetor: ponto]	Camada vetorial de pontos cujas feições são usadas como pontos iniciais das rotas
Ponto final (x, y)	`END_POINT`	[coordenadas]	Feição de ponto que representa o ponto final das rotas Press the … button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.
Caminho mais curto	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line layer for the shortest paths. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Non-routable features Opcional	`OUTPUT_NON_ROUTAB`	[vetor: ponto] Default: `[Pular saída]`	Specify the output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Campo de direção Opcional	DIREC- TION_FIELD	[campo da tabela: qualquer]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters Value for forward direction, Value for backward direction and Value for both directions. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the Default direction parameter) is used.
Valor para a direção direta Opcional	VALUE_FORWARD	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a forward direction
Valor para a direção para trás Opcional	VALUE_BACKWARD	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a backward direction
Valor para ambas as direções Opcional	VALUE_BOTH	[string] Padrão: '' (string vazia)	Value set in the direction field to identify bidirectional edges
Direção padrão	DE- FAULT_DIRECTIO	[enumeração] Padrão: 2	Se uma feição não tiver valor definido no campo de direção ou se nenhum campo de direção estiver definido, esse valor de direção será usado. Um de: • 0 — Direção para frente • 1 — Sentido inverso • 2 — Ambas as direções
Campo de velocidade Opcional	SPEED_FIELD	[campo de tabela: numérico]	Field providing the speed value (in km/h) for the edges of the network when looking for the fastest path. Se uma feição não tiver um valor nesse campo ou nenhum campo estiver definido, o valor da velocidade padrão (fornecido com o parâmetro Velocidade padrão) será usado.
Velocidade padrão (km/h)	DEFAULT_SPEED	[numeric: double] Padrão: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field
** Tolerância de Topologia**	TOLERANCE	[numeric: double] Padrão: 0.0	Duas linhas com nós mais próximos que a tolerância especificada são consideradas conectadas

Maximum point distance from network Opcional	POINT_TOLERANC	[numeric: double] Padrão: Não definido	Specifies an optional limit on the distance from the start and end points to the network layer. If a start feature is further from the network than this distance it will be treated as non-routable. If the end point is further from the network than this distance an error will be raised. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are.

Saídas

Rótulo	Nome	Tipo	Descrição
Caminho mais curto	`OUTPUT`	[vetor: linha]	Camada de linha do caminho mais curto ou mais rápido de cada um dos pontos iniciais até o ponto final
Non routable features	`OUTPUT_NON_ROUTAB`	[vetor: ponto]	An optional output layer which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer).

Código Python

Algorithm ID: `native:shortestpathlayertopoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Caminho mais curto (ponto a camada)

Calcula as rotas ideais (mais curtas ou mais rápidas) entre um determinado ponto inicial e vários pontos finais definidos por uma camada vetorial de ponto.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada vetorial representando a rede	`INPUT`	[vetor: linha]	Camada de vetor de linha representando a rede a ser coberta
Tipo de caminho a ser calculado	`STRATEGY`	[enumeração] Padrão: 0	O tipo de caminho a ser calculado. Um de: • 0 — Mais curto • 1 — Mais rápido

continua na próxima página

Tabela 24.231 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Start point	START_POINT	[coordenadas]	Feição de ponto que representa o ponto inicial das rotas Press the … button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.
Camada vetorial com pontos finais	END_POINTS	[vetor: ponto]	Camada vetorial de ponto cujas feições são usadas como pontos finais das rotas
Caminho mais curto	OUTPUT	[vetor: linha] Padrão: [Criar camada temporária]	Specify the output line layer for the shortest paths. *One of*: • Criar Camada Temporária (TEMPORARY_OUTPUT) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Non-routable features	OUTPUT_NON_ROUTAB Opcional	[vetor: ponto] Default: [Pular saída]	Specify the output which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer). *One of*: • Ignorar Saída • Criar Camada Temporária (TEMPORARY_OUTPUT) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Campo de direção Opcional	DIRECTION_FIELD	[campo da tabela: qualquer]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters Value for forward direction, Value for backward direction and Value for both directions. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the Default direction parameter) is used.
Valor para a direção direta Opcional	VALUE_FORWARD	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a forward direction

continua na próxima página

Tabela 24.232 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Valor para a direção para trás Opcional	VALUE_BACKWARD	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a backward direction
Valor para ambas as direções Opcional	VALUE_BOTH	[string] Padrão: '' (string vazia)	Value set in the direction field to identify bidirectional edges
Direção padrão	DEFAULT_DIRECTIO	[enumeração] Padrão: 2	Se uma feição não tiver valor definido no campo de direção ou se nenhum campo de direção estiver definido, esse valor de direção será usado. Um de: • 0 — Direção para frente • 1 — Sentido inverso • 2 — Ambas as direções
Campo de velocidade Opcional	SPEED_FIELD	[campo de tabela: numérico]	Field providing the speed value (in km/h) for the edges of the network when looking for the fastest path. Se uma feição não tiver um valor nesse campo ou nenhum campo estiver definido, o valor da velocidade padrão (fornecido com o parâmetro Velocidade padrão) será usado.
Velocidade padrão (km/h)	DEFAULT_SPEED	[numeric: double] Padrão: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field
** Tolerância de Topologia**	TOLERANCE	[numeric: double] Padrão: 0.0	Duas linhas com nós mais próximos que a tolerância especificada são consideradas conectadas
Maximum point distance from network Opcional	POINT_TOLERANC	[numeric: double] Padrão: Não definido	Specifies an optional limit on the distance from the start and end points to the network layer. If the start point is further from the network than this distance an error will be raised. If an end feature is further from the network than this distance it will be treated as non-routable. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are.

Saídas

Rótulo	Nome	Tipo	Descrição
Caminho mais curto	OUTPUT	[vetor: linha]	Camada de linha do caminho mais curto ou mais rápido de cada um dos pontos iniciais até o ponto final
Non routable features	OUTPUT_NON_ROUTAB	[vetor: ponto]	An optional output layer which will be used to store any input features which could not be routed (e.g., those which are too far from the network layer).

Código Python

Algorithm ID: `native:shortestpathpointtolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Caminho mais curto (ponto a ponto)

Calcula a rota ideal (mais curta ou mais rápida) entre um determinado ponto inicial e um determinado ponto final.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada vetorial representando a rede	`INPUT`	[vetor: linha]	Camada de vetor de linha representando a rede a ser coberta
Tipo de caminho a ser calculado	`STRATEGY`	[enumeração] Padrão: 0	O tipo de caminho a ser calculado. Um de: • 0 — Mais curto • 1 — Mais rápido
Ponto inicial (x, y)	`START_POINT`	[coordenadas]	Point feature representing the start point of the routes. Press the … button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.
Ponto final (x, y)	`END_POINT`	[coordenadas]	Point feature representing the end point of the routes. Press the … button next to the option and click on the canvas to fill the parameter with the clicked point coordinate.
Caminho mais curto	`OUTPUT`	[vetor: linha]	Specify the output line layer for the shortest paths. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Campo de direção Opcional	`DIRECTION_FIELD`	[campo da tabela: qualquer]	The field used to specify directions for the network edges. The values used in this field are specified with the three parameters `Value for forward direction`, `Value for backward direction` and `Value for both directions`. Forward and reverse directions correspond to a one-way edge, "both directions" indicates a two-way edge. If a feature does not have a value in this field, or no field is set then the default direction setting (provided with the `Default direction` parameter) is used.
Valor para a direção direta Opcional	`VALUE_FORWARD`	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a forward direction
Valor para a direção para trás Opcional	`VALUE_BACKWARD`	[string] Padrão: '' (string vazia)	Value set in the direction field to identify edges with a backward direction
Valor para ambas as direções Opcional	`VALUE_BOTH`	[string] Padrão: '' (string vazia)	Value set in the direction field to identify bidirectional edges
Direção padrão	`DEFAULT_DIRECTIO`	[enumeração] Padrão: 2	Se uma feição não tiver valor definido no campo de direção ou se nenhum campo de direção estiver definido, esse valor de direção será usado. Um de: • 0 — Direção para frente • 1 — Sentido inverso • 2 — Ambas as direções
Campo de velocidade Opcional	`SPEED_FIELD`	[campo de tabela: numérico]	Field providing the speed value (in `km/h`) for the edges of the network when looking for the fastest path. Se uma feição não tiver um valor nesse campo ou nenhum campo estiver definido, o valor da velocidade padrão (fornecido com o parâmetro `Velocidade padrão`) será usado.
Velocidade padrão (km/h)	`DEFAULT_SPEED`	[numeric: double] Padrão: 50.0	Value to use to calculate the travel time if no speed value is provided for an edge in the specified field
** Tolerância de Topologia**	`TOLERANCE`	[numeric: double] Padrão: 0.0	Duas linhas com nós mais próximos que a tolerância especificada são consideradas conectadas

Maximum point distance from network Opcional	POINT_TOLERANC	[numeric: double] Padrão: Não definido	Specifies an optional limit on the distance from the start and end points to the network layer. If either point is further from the network than this distance an error will be raised. If not set, points will be snapped to the nearest point on the network layer, regardless of how far away from the network they actually are.

Saídas

Rótulo	Nome	Tipo	Descrição
Caminho mais curto	OUTPUT	[vetor: linha]	Camada de linha do caminho mais curto ou mais rápido de cada ponto inicial até o ponto final

Código Python

Algorithm ID: `native:shortestpathpointtopoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.14 Plots

Gráfico de barras

Cria um gráfico de barras a partir de uma categoria e de um campo de camadas.

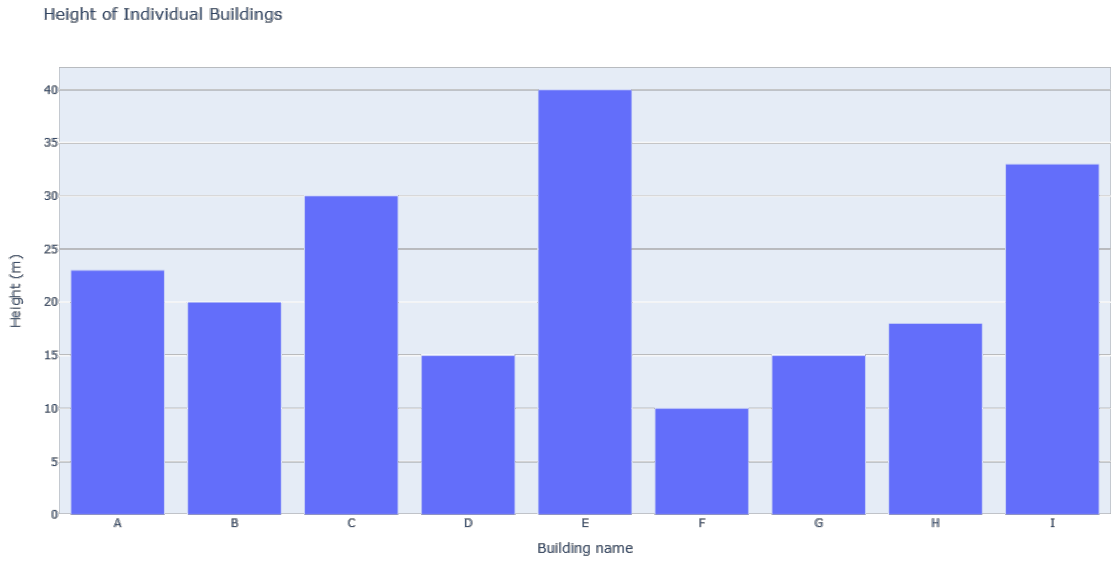


Figura24.32: Bar plot example

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Nome do campo de categoria	`NAME_FIELD`	[campo da tabela: qualquer]	Campo categórico a ser usado para agrupar as barras (eixo X)
Campo valor	`VALUE_FIELD`	[campo de tabela: numérico]	Valor a ser utilizado para o gráfico (eixo Y).
Title Optional	`TITLE`	[string] Default: “”	Title of the plot
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: “”	If empty, the name of the category field is used. With a single space, the axis title is hidden.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: “”	If empty, the name of the value field is used. With a single space, the axis title is hidden.
Gráfico de barras	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the HTML file for the plot. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Gráfico de barras	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Código Python

Algorithm ID: `qgis:barplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Box plot

Creates a box plot from a category field and a numerical layer field.

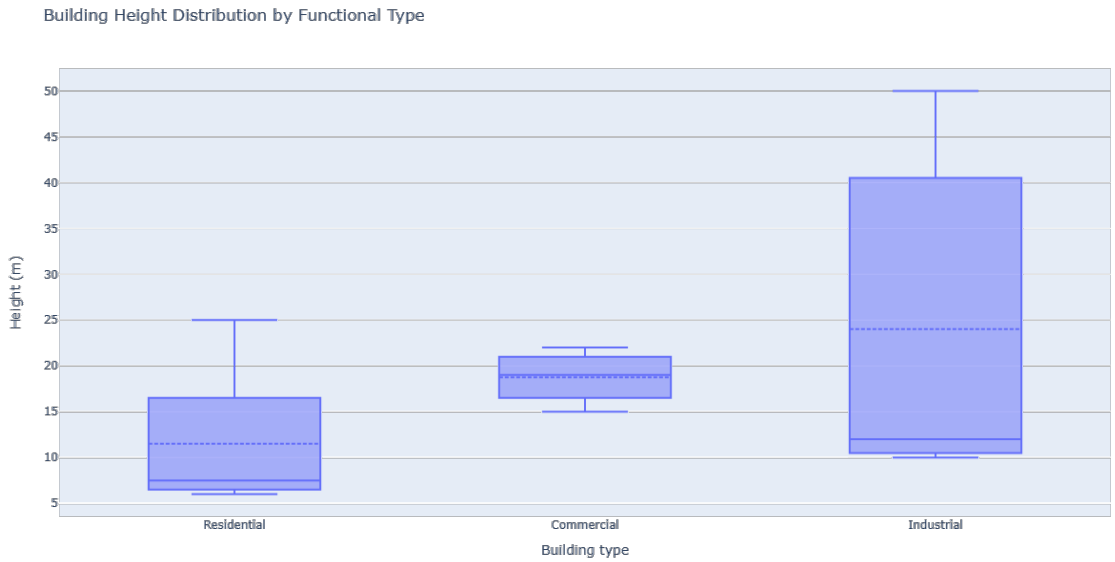


Figura24.33: Example of box plot

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Campo de nome da categoria	`NAME_FIELD`	[campo da tabela: qualquer]	Categorical field to use for grouping the boxes (X axis)
Campo valor	`VALUE_FIELD`	[campo de tabela: numérico]	Valor a ser utilizado para o gráfico (eixo Y).
Linhas estatísticas adicionais	`MSD`	[enumeração] Padrão: 0	Additional statistics information to add to the plot. One of: • 0 — Mostrar Média • 1 — Mostrar Desvio Padrão • 2 — Não mostrar média e desvio padrão

continua na próxima página

Tabela 24.239 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Title Optional	`TITLE`	[string] Default: “”	Title of the plot
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: “”	If empty, the name of the category field is used. With a single space, the axis title is not shown.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: “”	If empty, the name of the value field is used. With a single space, the axis title is not shown.
Box plot	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the HTML file for the plot. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Box plot	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Código Python

Algorithm ID: `qgis:boxplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Generate elevation profile image

Creates an elevation profile image from a list of map layers and an optional terrain layer.

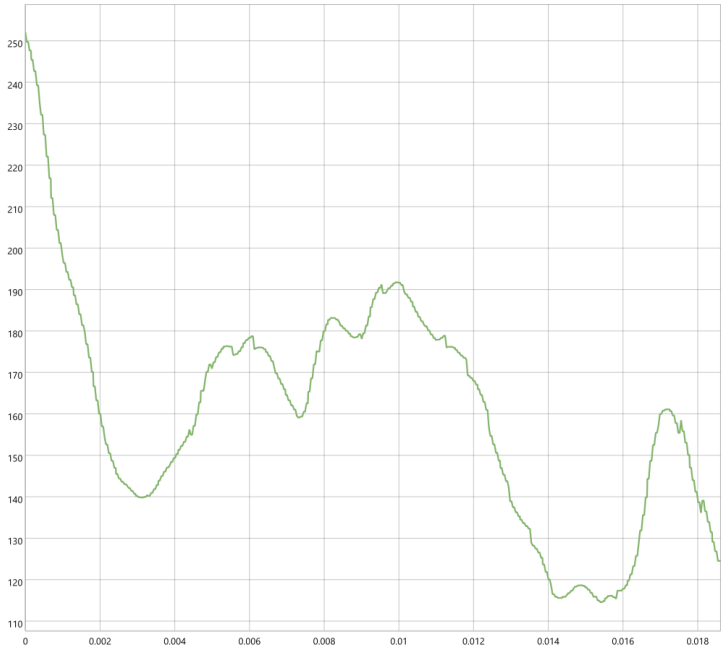


Figura24.34: Example of elevation profile

Parâmetros

Basic parameters

Rótulo	Nome	Tipo	Descrição
Profile curve	`CURVE`	[geometry: line]	The LineString geometry (WKT or GeoJSON) representing the profile curve along which the elevation profile will be generated
Map layers	`MAP_LAYERS`	[layer] [list]	The list of map layers to include in the elevation profile
Chart width (in pixels)	`WIDTH`	[numeric: integer] Default: 400 Minimum value: 0	The width of the output chart in pixels.
Chart height (in pixels)	`HEIGHT`	[numeric: integer] Default: 300 Minimum value: 0	The height of the output chart in pixels.
Terrain layer Optional	`TERRAIN_LAYER`	[raster]	A terrain layer (e.g., DEM) to use for elevation data. If not provided, elevation data will be derived from the map layers.

Advanced parameters

Rótulo	Nome	Tipo	Descrição
Chart minimum distance (X axis) Optional	`MINIMUM_DISTANCE`	[numeric: double]	The minimum distance (X axis) to display on the chart. If not specified, the chart will auto-scale.
Chart maximum distance (X axis) Optional	`MAXIMUM_DISTANCE`	[numeric: double]	The maximum distance (X axis) to display on the chart. If not specified, the chart will auto-scale.
Chart minimum elevation (Y axis) Optional	`MINIMUM_ELEVATION`	[numeric: double]	The minimum elevation (Y axis) to display on the chart. If not specified, the chart will auto-scale.
Chart maximum elevation (Y axis) Optional	`MAXIMUM_ELEVATION`	[numeric: double]	The maximum elevation (Y axis) to display on the chart. If not specified, the chart will auto-scale.
Chart text color Optional	`TEXT_COLOR`	[color]	The color of the text in the chart (e.g., axis labels, titles).
Chart background color Optional	`BACKGROUND_COLOR`	[color]	The background color of the chart.
Chart border color Optional	`BORDER_COLOR`	[color]	The color of the chart border.
Profile tolerance	`TOLERANCE`	[numeric: double] Default: 5.0 Minimum value: 0	Defines how far a feature (vector point, line, polygon, or point cloud) can be from the profile line to be included in the results. It uses map units and does not affect other layer types.
Chart DPI	`DPI`	[numeric: integer] Default: 96 Minimum value: 0	The resolution of the output image in dots per inch (DPI).
Output image	`OUTPUT`	[file] Padrão: `[Salvar em arquivo temporário]`	Specify the image file for the plot. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Output image	`OUTPUT`	[file]	The generated elevation profile image in the specified format.

Código Python

Algorithm ID: `native:generateelevationprofileimage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Gráfico de média e desvio padrão

Creates a box plot with mean and standard deviation values.

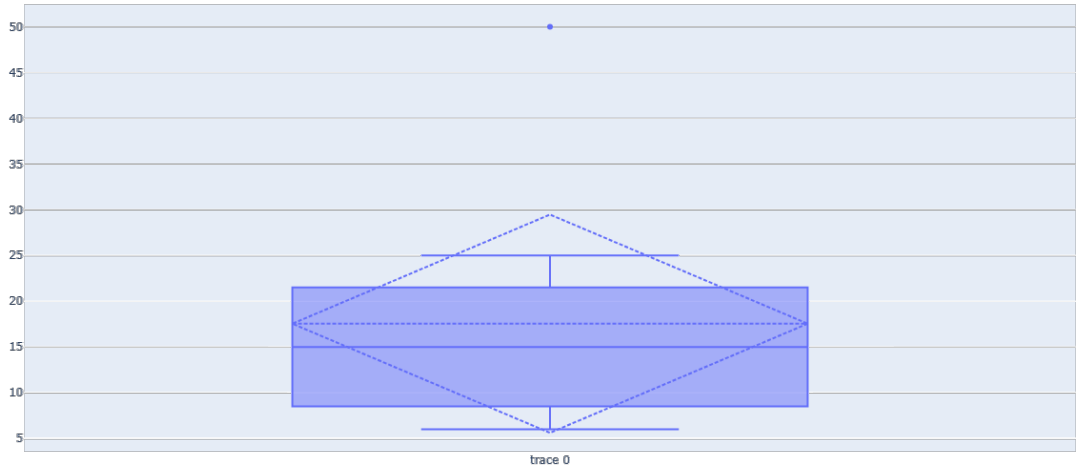


Figura24.35: Example of mean and standard deviation plot

Parâmetros

Rótulo	Nome	Tipo	Descrição
Tabela de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Campo de nome da categoria	`NAME_FIELD`	[campo da tabela: qualquer]	Categorical field to use for grouping the boxes (X axis)
Campo valor	`VALUE_FIELD`	[campo de tabela: numérico]	Valor a ser utilizado para o gráfico (eixo Y).
Plot	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the HTML file for the plot. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Plot	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Código Python

Algorithm ID: `qgis:meanandstandarddeviationplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Gráfico polar

Gera um gráfico polar com base no valor de uma camada vetorial de entrada.

Dois campos devem ser inseridos como parâmetros: um que define a categoria de cada característica (para agrupar as características) e outro com a variável a ser plotada (este tem que ser numérico).

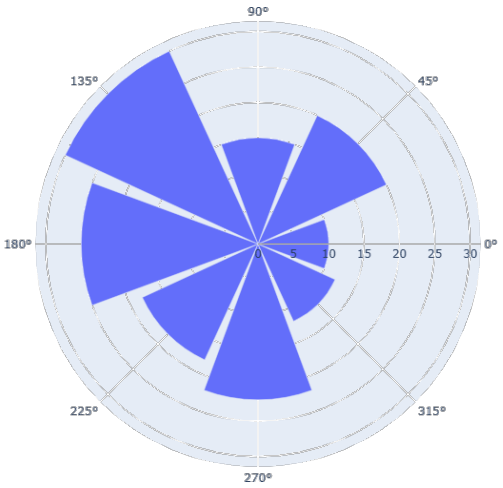


Figura24.36: Example of polar plot

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Campo de nome da categoria	`NAME_FIELD`	[campo da tabela: qualquer]	Campo categórico a ser usado para agrupar as características (eixo X)
Campo valor	`VALUE_FIELD`	[campo de tabela: numérico]	Valor a ser utilizado para o gráfico (eixo Y).
Gráfico polar	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the HTML file for the plot. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Gráfico polar	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Código Python

ID do algoritmo: `qgis:graficoPolar`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Histograma da camada raster

Gera um histograma com os valores de uma camada rasterizada.

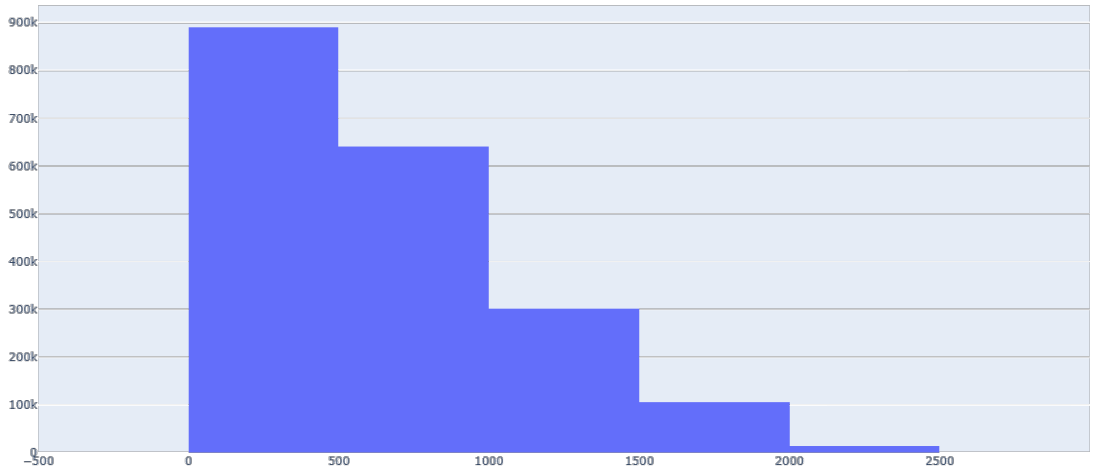


Figura24.37: Raster layer histogram example

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster]	Faixa raster a ser usada para o histograma
número de caixas	`BINS`	[numeric: integer] Padrão: 10	O número de caixas a serem utilizados no histograma (eixo X). Mínimo 2.
Histograma	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the HTML file for the plot. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Histograma	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Código Python

Algorithm ID: `qgis:rasterlayerhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Histograma da camada vetorial

Gera um histograma com os valores do atributo de uma camada vetorial.

O atributo a ser utilizado para o cálculo do histograma deve ser numérico.

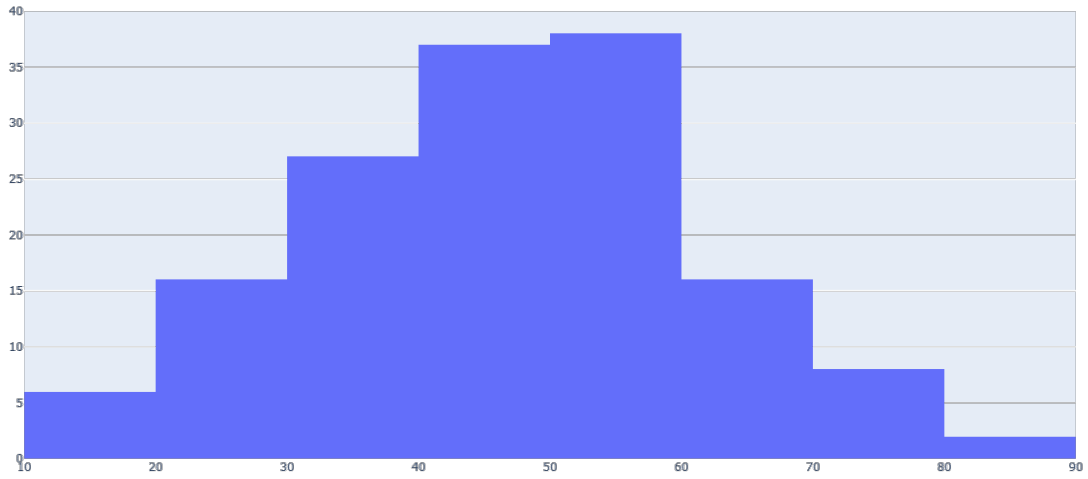


Figura24.38: Example of vector layer histogram

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
Atributo	`FIELD`	[campo de tabela: numérico]	Valor a ser utilizado para o gráfico (eixo Y).
número de caixas	`BINS`	[numeric: integer] Padrão: 10	O número de caixas a serem utilizados no histograma (eixo X). Mínimo 2.
Histograma	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the HTML file for the plot. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Histograma	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Código Python

Algorithm ID: `qgis:vectorlayerhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Gráfico de dispersão da camada vetorial

Cria um simples gráfico de dispersão `X` - `Y` para uma camada vetorial.

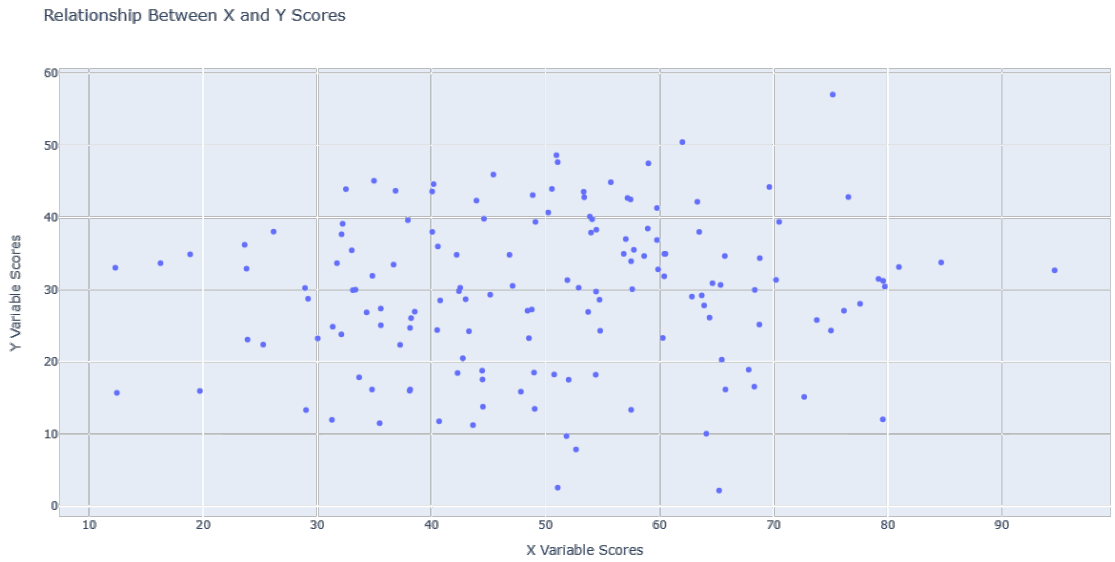


Figura24.39: Example of vector layer scatterplot

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Atributo X	`XFIELD`	[campo de tabela: numérico]	Campo a ser utilizado para o eixo X
Atributo Y	`YFIELD`	[campo de tabela: numérico]	Campo a ser utilizado para o eixo Y
Hover text Optional	`HOVERTEXT`	[expression] Default: “”	Text to be shown when hovering with the mouse over data points. It can be picked from a field or based on an expression.
Title Optional	`TITLE`	[string] Default: “”	Title of the plot

continua na próxima página

Tabela 24.252 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: “”	If empty, the field name of the x attribute is used. With a single space, the axis title is not shown.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: “”	If empty, the field name of the y attribute is used. With a single space, the axis title is not shown.
Use logarithmic scale for x-axis	`XAXIS_LOG`	[boolean] Default: False	When enabled, uses logarithmic scale for the x-axis
Use logarithmic scale for y-axis	`YAXIS_LOG`	[boolean] Default: False	When enabled, uses logarithmic scale for the y-axis
Gráfico de dispersão	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the HTML file for the plot. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Gráfico de dispersão	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Código Python

Algorithm ID: `qgis:vectorlayerscatterplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Gráfico de dispersão 3D da camada vetorial

Cria um gráfico de dispersão 3D para uma camada vetorial.

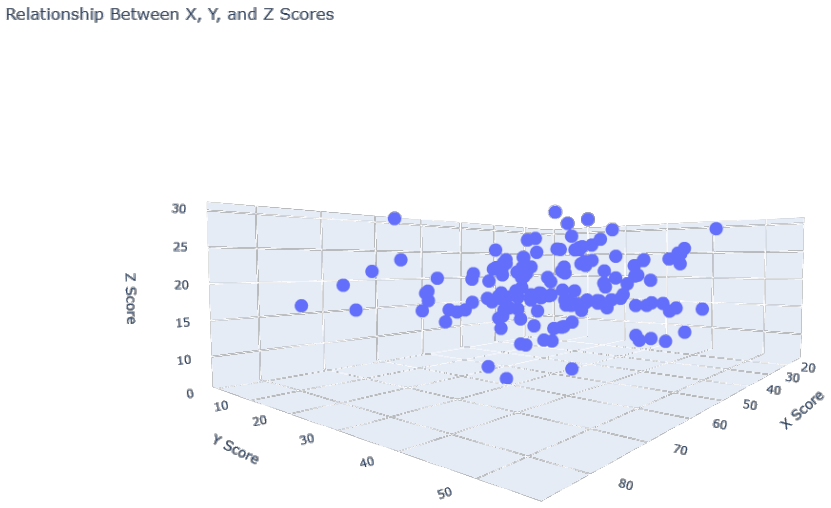


Figura24.40: Example of 3D scatter plot

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Atributo X	`XFIELD`	[campo de tabela: numérico]	Campo a ser utilizado para o eixo X
Atributo Y	`YFIELD`	[campo de tabela: numérico]	Campo a ser utilizado para o eixo Y
Z atributo	`ZFIELD`	[campo de tabela: numérico]	Campo a ser utilizado para o eixo Z
Title Optional	`TITLE`	[string] Default: “”	Title of the plot
X-axis Title Optional	`XAXIS_TITLE`	[string] Default: “”	If empty, the field name of the X attribute is used.
Y-axis Title Optional	`YAXIS_TITLE`	[string] Default: “”	If empty, the field name of the Y attribute is used.
Z-axis Title Optional	`ZAXIS_TITLE`	[string] Default: “”	If empty, the field name of the Z attribute is used.
Scatterplot 3D	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the HTML file for the plot. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Scatterplot 3D	`OUTPUT`	[html]	HTML file with the plot. Available in the *Processing* ► *Result Viewer*.

Código Python

Algorithm ID: `qgis:scatter3dplot`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.15 Point Cloud Conversion

Atenção

Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* ► *About* menu).

Converter formato

Converts a point cloud to a different file format, e.g. creates a compressed `.LAZ`.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to convert
Convertido	`OUTPUT`	[point cloud] Padrão: `[Salvar em arquivo temporário]`	Specify the point cloud file to use as output. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Convertido	`OUTPUT`	[point cloud]	Output point cloud layer in a modified file format. Currently supported formats are `.LAS`, `.LAZ`, `.COPC.LAZ` and `.VPC`.

Código Python

Algorithm ID: `pdal:convertformat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Export to raster

Exports point cloud data to a 2D raster grid having cell size of given resolution, writing values from the specified attribute.

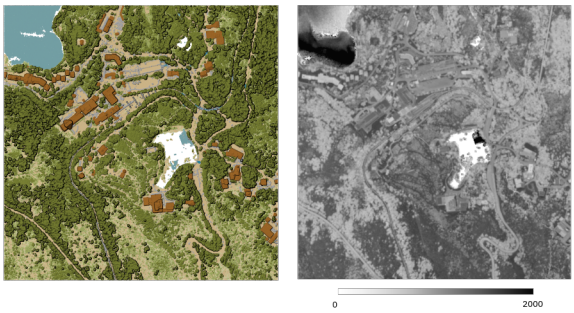


Figura24.41: Raster output using Intensity attribute of points

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to export
Atributo	`ATTRIBUTE`	[field] [enumeration]	A Field of the point cloud layer to extract the values from
Resolution of the density raster	`RESOLUTION`	[numeric: double] Padrão: 1.0	Cell size of the output raster
Tile size for parallel runs	`TILE_SIZE`	[numeric: integer] Padrão: 1000	

continua na próxima página

Tabela 24.258 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Exported	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the raster file to export the data to. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Inserir as coordenadas como `xmin, xmax, ymin, ymax`
X origin of a tile for parallel runs Opcional	`ORIGIN_X`	[numeric: double]	
Y origin of a tile for parallel runs Opcional	`ORIGIN_Y`	[numeric: double]	

Saídas

Rótulo	Nome	Tipo	Descrição
Exported	`OUTPUT`	[raster]	Output raster layer features of the point cloud layer are exported to. Currently supported format is `.TIF`.

Código Python

Algorithm ID: `pdal:exportraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Export raster (using triangulation)

Exports point cloud data to a 2D raster grid using a triangulation of points and then interpolating cell values from triangles.

Nota

Using this algorithm can be slower if you are dealing with a large dataset. If your point cloud is dense, you can export your ground points as a raster using the *Export to raster* algorithm.

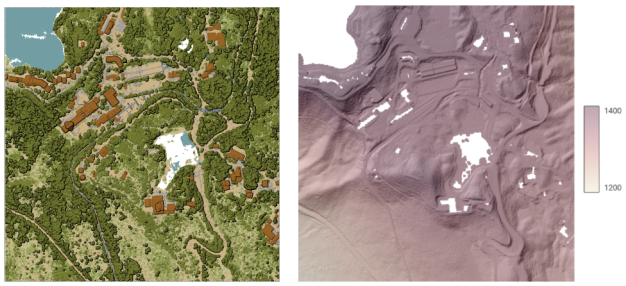


Figura24.42: Terrain raster output generated by point cloud triangulation

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to export
Resolution of the density raster	`RESOLUTION`	[numeric: double] Padrão: 1.0	Cell size of the output raster
Tile size for parallel runs	`TILE_SIZE`	[numeric: integer] Padrão: 1000	
Exported	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the raster file to export the data to. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Inserir as coordenadas como `xmin, xmax, ymin, ymax`
X origin of a tile for parallel runs Opcional	`ORIGIN_X`	[numeric: double]	
Y origin of a tile for parallel runs Opcional	`ORIGIN_Y`	[numeric: double]	

Saídas

Rótulo	Nome	Tipo	Descrição
Exported (using triangulation)	`OUTPUT`	[raster]	Output raster layer features of the point cloud layer are exported to. Currently supported format is `.TIF`.

Código Python

Algorithm ID: `pdal:exportrastertin`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Export to vector

Exports point cloud data to a vector layer with 3D points (a GeoPackage), optionally with extra attributes.

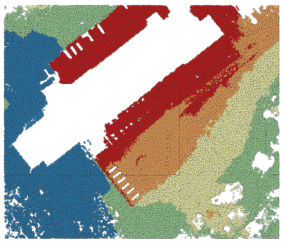


Figura24.43: Exporting point cloud (ground points) to a vector layer styled based on the elevation

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to export
Atributo Opcional	`ATTRIBUTE`	[field] [list]	One or more fields of the point cloud layer to export with the points.
Exported	`OUTPUT`	[vetor] Padrão: `[Salvar em arquivo temporário]`	Specify the vector file to export the data to. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data

continua na próxima página

Tabela 24.265 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Inserir as coordenadas como `xmin, xmax, ymin, ymax`

Saídas

Rótulo	Nome	Tipo	Descrição
Exported	`OUTPUT`	[vetor]	Output vector layer features of the point cloud layer are exported to. Currently supported format is `.GPKG`.

Código Python

Algorithm ID: `pdal:exportvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.16 Point Cloud Data Management

Atenção

Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* ► *About* menu).

Atribuir projeção

Assigns a Coordinate Reference System to a point cloud layer, if it is missing or wrong. A new layer is created.

Ver também

Reproject

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to assign a CRS to
SRC desejada	`CRS`	[src]	The CRS to apply to the layer
Camada de saída	`OUTPUT`	[point cloud] Padrão: `[Salvar em arquivo temporário]`	Specify the point cloud file to use as output. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Convertido	`OUTPUT`	[point cloud]	Output point cloud layer with a new CRS. Currently supported formats are `.LAS`, `.LAZ`, `.COPC.LAZ` and `.VPC`.

Código Python

Algorithm ID: `pdal:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Build virtual point cloud (VPC)

Creates a *virtual point cloud (VPC)* from input point cloud data.

If you leave the optional parameters unchecked, the VPC file will be built very quickly as the algorithm will only read metadata of input files. With any of the optional parameters set, the algorithm will read all points which can take some time.

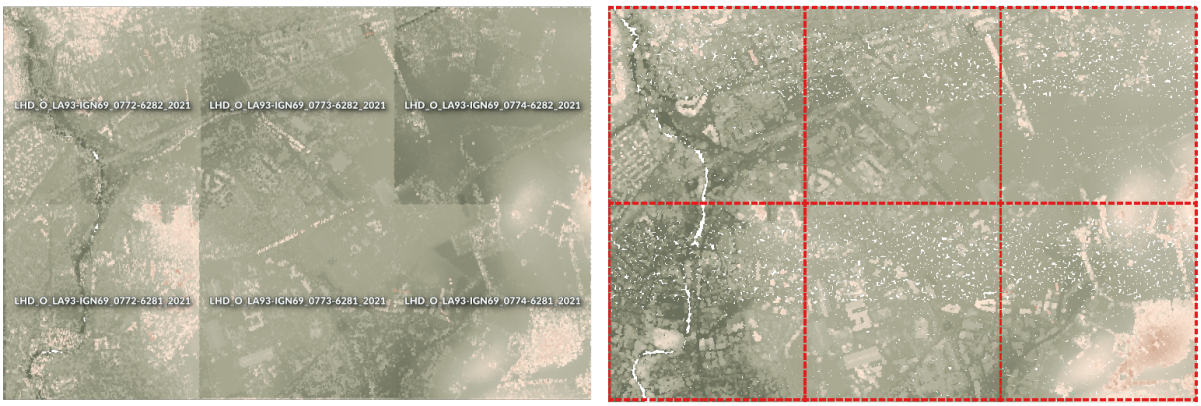


Figura24.44: Generating Virtual Point Cloud with overview from a set of point cloud tiles

Ver também

Mesclar

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`LAYERS`	[point cloud] [list]	Input point cloud layers to combine inside a virtual point cloud layer
Calculate boundary polygon	`BOUNDARY`	[boleano] Padrão: Falso	Set to True to show the exact boundaries of data (rather than just rectangular extent)
Calculate statistics	`STATISTICS`	[boleano] Padrão: Falso	Set to True to understand ranges of values of the various attributes
Build overview point cloud	`OVERVIEW`	[boleano] Padrão: Falso	Generates a single “thinned” point cloud of all the input data (using only every 1000th point from original data). The overview point cloud will be created next to the VPC file - for example, for `mydata.vpc`, the overview point cloud would be named `mydata-overview.copc.laz`.
Virtual point cloud	`OUTPUT`	[point cloud] Padrão: `[Salvar em arquivo temporário]`	Specify the point cloud file to build the data into. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Virtual point cloud	`OUTPUT`	[raster]	Output point cloud layer combining all the input data, as a virtual file.

Código Python

Algorithm ID: `pdal:virtualpointcloud`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Clip

Clips a point cloud layer by a polygon layer so that the resulting point cloud contains only points within the polygons.

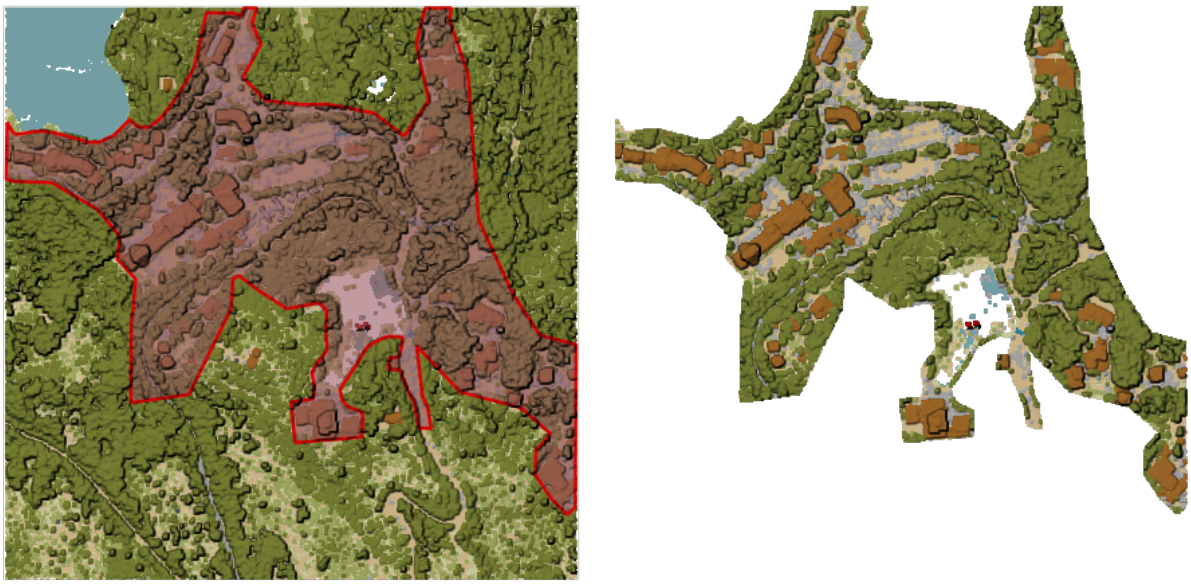


Figura24.45: Clipping an input point cloud layer with a polygon coverage

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to clip
Clipping polygons	`OVERLAY`	[vetor: polígono]	Polygon vector layer to use as coverage for clipping the points
Clipped	`OUTPUT`	[point cloud] Padrão: `[Salvar em arquivo temporário]`	Specify the point cloud file to export the clipped points to. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`

Saídas

Etiqueta	Nome	Tipo	Descrição
Clipped	`OUTPUT`	[point cloud]	Output point cloud whose features are the points within the coverage polygon layer.

Código Python

Algorithm ID: `pdal:clip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Create COPC

Creates the index for all the input point cloud files in a batch mode.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`LAYERS`	[point cloud] [list]	Input point cloud layers to create an index for
Diretório de saída Opcional	`OUTPUT`	[pasta] Default: `[Pular saída]`	Specify the folder to create the new files in. *One of*: • Ignorar Saída • Salvar como Diretório Temporário • Salvar no Diretório

Saídas

Etiqueta	Nome	Tipo	Descrição
Diretório de saída	`OUTPUT`	[pasta]	Output folder containing point cloud layers with accompanying COPC index files.

Código Python

Algorithm ID: `pdal:createcopc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Information

Outputs basic metadata from an input point cloud file.

Example of output information:

```
LAS           1.4
point format  6
count         56736130
scale         0.001 0.001 0.001
offset        431749.999 5440919.999 968.898
extent        431250 5440420 424.266
              432249.999 5441419.999 1513.531
crs           ETRS89 / UTM zone 34N (N-E) (EPSG:3046)  (vertical CRS missing!)
units         horizontal=metre  vertical=unknown

Attributes:
- X floating 8
- Y floating 8
- Z floating 8
- Intensity unsigned 2
- ReturnNumber unsigned 1
- NumberOfReturns unsigned 1
- ScanDirectionFlag unsigned 1
- EdgeOfFlightLine unsigned 1
- Classification unsigned 1
- ScanAngleRank floating 4
- UserData unsigned 1
- PointSourceId unsigned 2
- GpsTime floating 8
- ScanChannel unsigned 1
- ClassFlags unsigned 1
```

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to extract metadata information from
Informação da camada	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Specify the file to store the metadata information. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Informação da camada	`OUTPUT`	[html]	`HTML` file to store the metadata information.

Código Python

Algorithm ID: `pdal:info`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Mesclar

Merges multiple point cloud files into a single one.

Ver também

Build virtual point cloud (VPC)

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`LAYERS`	[point cloud] [list]	Input point cloud layers to merge into a single one
Mesclado	`OUTPUT`	[point cloud] Padrão: `[Salvar em arquivo temporário]`	Specify the output point cloud merging input files. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data

continua na próxima página

Tabela 24.279 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`

Saídas

Etiqueta	Nome	Tipo	Descrição
Mesclado	`OUTPUT`	[point cloud]	Output point cloud layer merging all the input files.

Código Python

Algorithm ID: `pdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Reproject

Reprojects a point cloud to a different Coordinate Reference System (CRS).

Ver também

Atribuir projeção

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to reproject to a different CRS
SRC Alvo	`CRS`	[src]	The CRS to apply to the layer
Reprojetado	`OUTPUT`	[point cloud] Padrão: `[Salvar em arquivo temporário]`	Specify the reprojected point cloud file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Coordinate operation Opcional	`OPERATION`	[datum]	The *datum transformation* to use to reproject the data between the origin and target systems.

Saídas

Etiqueta	Nome	Tipo	Descrição
REPROJECTED	`OUTPUT`	[point cloud]	Output point cloud layer in the target CRS.

Código Python

Algorithm ID: `pdal:reproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Thin (by sampling radius)

Creates a thinned version of the point cloud by performing sampling by distance point (reduces the number of points within a certain radius).

Ver também

Thin (by skipping points)

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to create a thinned version from
Sampling radius (in map units)	`SAMPLING_RADIUS`	[numeric: double] Padrão: 1,0	Distance within which points are sampled to a unique point
Thinned (by radius)	`OUTPUT`	[point cloud] Padrão: `[Salvar em arquivo temporário]`	Specify the output point cloud with reduced points. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`

Saídas

Etiqueta	Nome	Tipo	Descrição
Thinned (by radius)	`OUTPUT`	[point cloud]	Output point cloud layer with reduced points.

Código Python

Algorithm ID: `pdal:thinbyradius`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Thin (by skipping points)

Creates a thinned version of the point cloud by keeping only every N-th point (reduces the number of points by skipping nearby points).

Ver também

Thin (by sampling radius)

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to create a thinned version from
Number of points to skip	`POINTS_NUMBER`	[numeric: integer] Padrão: 1	Keep only every N-th point in the input layer
Thinned (by decimation)	`OUTPUT`	[point cloud] Padrão: `[Salvar em arquivo temporário]`	Specify the output point cloud with reduced points. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data

continua na próxima página

Tabela 24.288 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`

Saídas

Etiqueta	Nome	Tipo	Descrição
Thinned (by decimation)	`OUTPUT`	[point cloud]	Output point cloud layer with reduced points.

Código Python

Algorithm ID: `pdal:thinbydecimate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Tile

Creates tiles from input point cloud files, recommended for best performance (in display or analysis) with such datasets in QGIS.

Ver também

Build virtual point cloud (VPC), *Create COPC*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`LAYERS`	[point cloud] [list]	Input point cloud layers to create tiles from
Tile length	`LENGTH`	[numeric: double] Default: 1000.0	Size of the edge of each generated tile
Diretório de saída	`OUTPUT`	[pasta] Padrão: `[Salvar em pasta temporária]`	Specify the folder to store the generated tiles. *One of*: • Salvar como Diretório Temporário • Salvar no Diretório

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Assign CRS Opcional	`CRS`	[src]	The CRS to apply to the layer

Saídas

Etiqueta	Nome	Tipo	Descrição
Diretório de saída	`OUTPUT`	[pasta]	Output folder containing the tiles generated from input files.

Código Python

Algorithm ID: `pdal:tile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.17 Point Cloud Extraction

> **Atenção**
>
> Running these algorithms requires QGIS installed with PDAL >= 2.5.0 (see *Help* ► *About* menu).

Limite

Exports a polygon file containing point cloud layer boundary. It may contain holes and it may be a multi-part polygon.

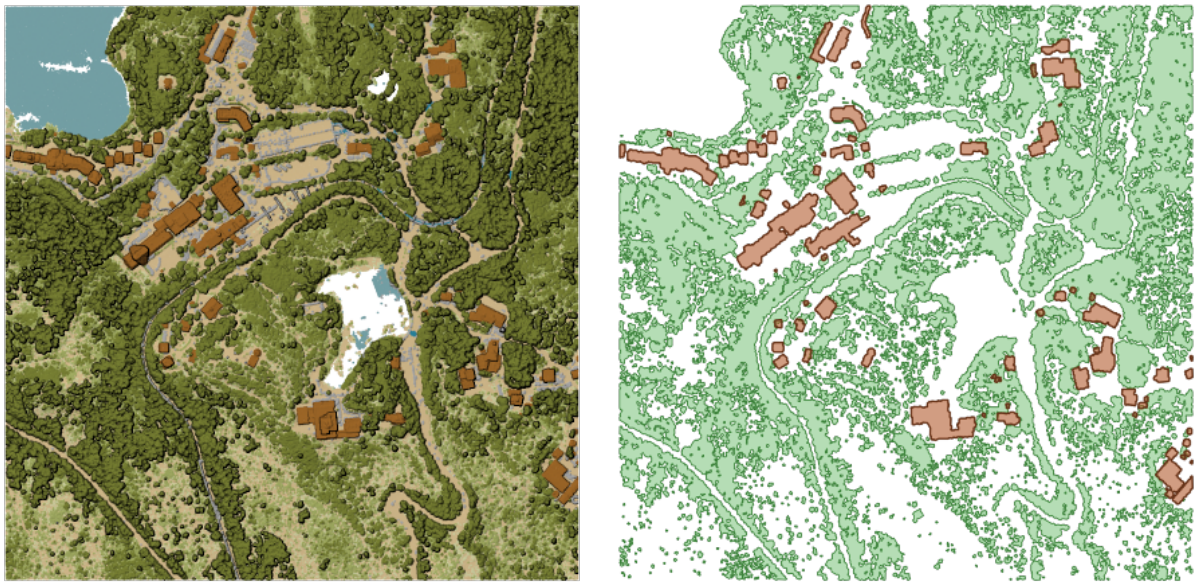


Figura24.46: Extracting high vegetation and building polygons from an input point cloud layer

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to calculate boundary for
Resolution of cells used to calculate boundary Opcional	`RESOLUTION`	[numeric: double]	Resolution of cells used to calculate boundary
Minimal number of points in a cell to consider cell occupied Opcional	`THRESHOLD`	[numeric: integer]	Minimal number of points in a cell to consider cell occupied

continua na próxima página

Tabela 24.293 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Limite	`OUTPUT`	[vetor: polígono] Padrão: `[Salvar em arquivo temporário]`	Specify the point cloud file to use as output. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`

Saídas

Etiqueta	Nome	Tipo	Descrição
Limite	`OUTPUT`	[vetor: polígono]	Output polygon vector layer with the point cloud boundary. Currently supported format is `.GPKG`.

Código Python

Algorithm ID: `pdal:boundary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Density

Exports a raster file based on the number of points within each raster cell - useful for quality checking of point cloud datasets.

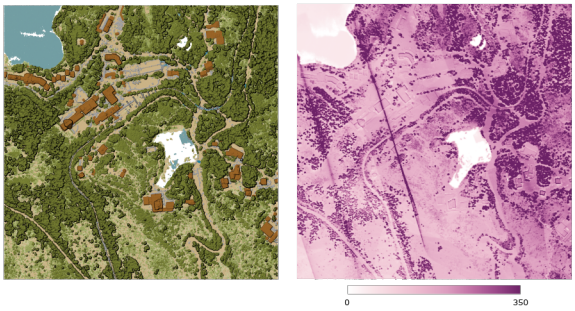


Figura24.47: Point density (number of points per 2x2 m) as a raster

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to export
Resolution of the density raster	`RESOLUTION`	[numeric: double] Padrão: 1,0	Cell size of the output raster
Tile size for parallel runs	`TILE_SIZE`	[numeric: integer] Padrão: 1000	Size of the tiles to split the data into for parallel runs
Density	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the raster file to export the data to. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
X origin of a tile for parallel runs Opcional	`ORIGIN_X`	[numeric: double]	
Y origin of a tile for parallel runs Opcional	`ORIGIN_Y`	[numeric: double]	

Saídas

Etiqueta	Nome	Tipo	Descrição
Density	`OUTPUT`	[raster]	Output raster layer with number of points within each cell. Currently supported format is `.TIF`.

Código Python

Algorithm ID: `pdal:density`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Filter

Extracts point from the input point cloud which match PDAL expression and/or are inside of a cropping rectangle.

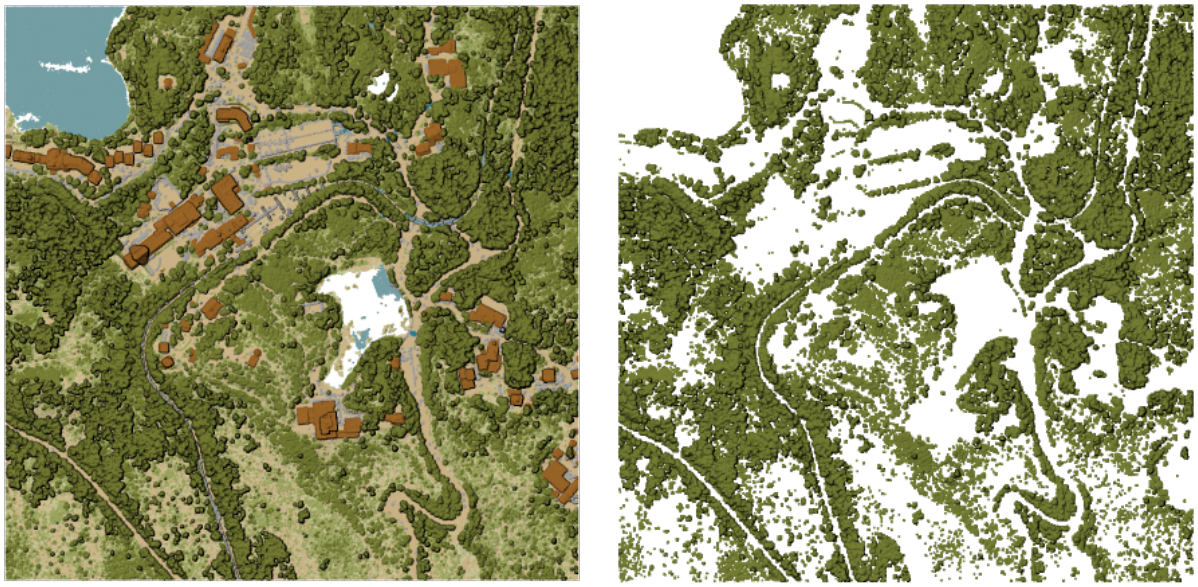


Figura24.48: Filtering of high vegetation class from an input point cloud layer

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[point cloud]	Input point cloud layer to export
Expressão do filtro Opcional	`FILTER_EXPRESSION`	[expressão]	A *PDAL expression* for selecting a subset of features in the point cloud data
Cropping extent Opcional	`FILTER_EXTENT`	[extensão]	A map extent for selecting a subset of features in the point cloud data Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`

continua na próxima página

Tabela 24.299 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Filtrado	`OUTPUT`	[point cloud] Padrão: `[Salvar em arquivo temporário]`	Specify the point cloud file to export the data to. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Filtrado	`OUTPUT`	[point cloud]	Output point cloud layer with the filtered features of the input point cloud layer. Currently supported formats are `.LAS`, `.LAZ`, `.COPC.LAZ` and `.VPC`.

Código Python

Algorithm ID: `pdal:filter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.18 Análise de dados Raster

Porcentagem da pilha de células em relação ao valor

Calculates the cell-wise percentrank value of a stack of rasters based on a single input value and writes them to an output raster.

At each cell location, the specified value is ranked among the respective values in the stack of all overlaid and sorted cell values from the input rasters. For values outside of the stack value distribution, the algorithm returns NoData because the value cannot be ranked among the cell values.

Há dois métodos para o cálculo do percentil:

- Interpolação linear inclusiva (PERCENTRANK.INC)
- Interpolação linear exclusiva (PERCENTRANK.EXC)

The linear interpolation methods return the unique percent rank for different values. Both interpolation methods follow their counterpart methods implemented by LibreOffice or Microsoft Excel.

The output raster's extent and resolution is defined by a reference raster. Input raster layers that do not match the cell size of the reference raster layer will be resampled using nearest neighbor resampling. NoData values in any of the input layers will result in a NoData cell output if the "Ignore NoData values" parameter is not set. The output raster data type will always be `Float32`.

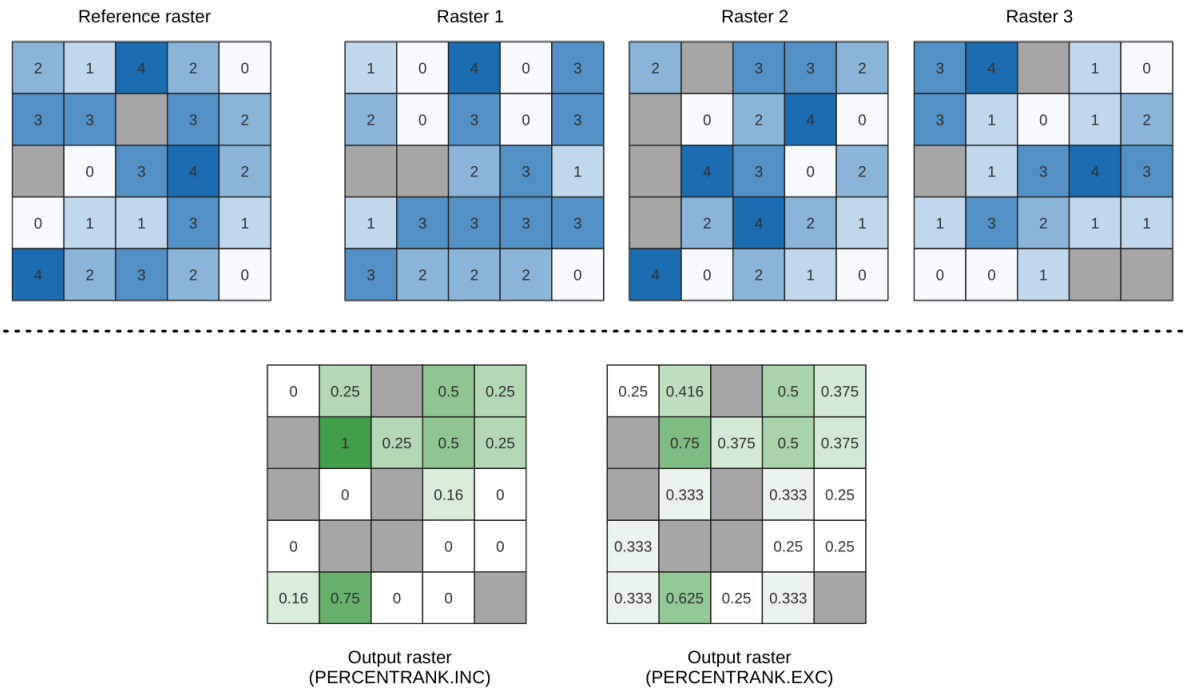


Figura24.49: Percent ranking Value = 1. `NoData` cells (grey) are ignored.

Ver também

Percentil de pilha celular, *Cell stack percentrank from raster layer*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Método	`METHOD`	[enumeração] Padrão: 0	Método para cálculo de percentil: • 0 — Interpolação linear inclusiva (PERCENTRANK.INC) • 1 — Interpolação linear exclusiva (PERCENTRANK.INC)
Valor	`VALUE`	[numeric: double] Padrão: 10.0	Value to rank among the respective values in the stack of all overlaid and sorted cell values from the input rasters
Ignorar valores sem dados	`IGNORE_NODATA`	[boleano] Padrão: Verdadeiro	If unchecked, any NoData cells in the input layers will result in a NoData cell in the output raster
Camada de referência	`REFERENCE_LAYER`	[raster]	A camada de referência para a criação da camada de saída (extensão, SRC, dimensões dos pixels)

continua na próxima página

Tabela 24.301 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value	`OUTPUT_NODATA_VAL`	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:cellstackpercentrankfromvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Percentil de pilha celular

Calculates the cell-wise percentile value of a stack of rasters and writes the results to an output raster. The percentile to return is determined by the percentile input value (ranges between 0 and 1). At each cell location, the specified percentile is obtained using the respective value from the stack of all overlaid and sorted cell values of the input rasters.

Há três métodos para o cálculo do percentil:

- Classificação mais próxima: retorna o valor mais próximo do percentil especificado
- Interpolação linear inclusiva (PERCENTRANK.INC)
- Interpolação linear exclusiva (PERCENTRANK.EXC)

Os métodos de interpolação linear retornam os valores únicos para diferentes percentis. Ambos os métodos de interpolação seguem seus métodos de contrapartida implementados pelo LibreOffice ou pelo Microsoft Excel.

The output raster's extent and resolution is defined by a reference raster. Input raster layers that do not match the cell size of the reference raster layer will be resampled using nearest neighbor resampling. NoData values in any of the input layers will result in a NoData cell output if the "Ignore NoData values" parameter is not set. The output raster data type will always be `Float32`.

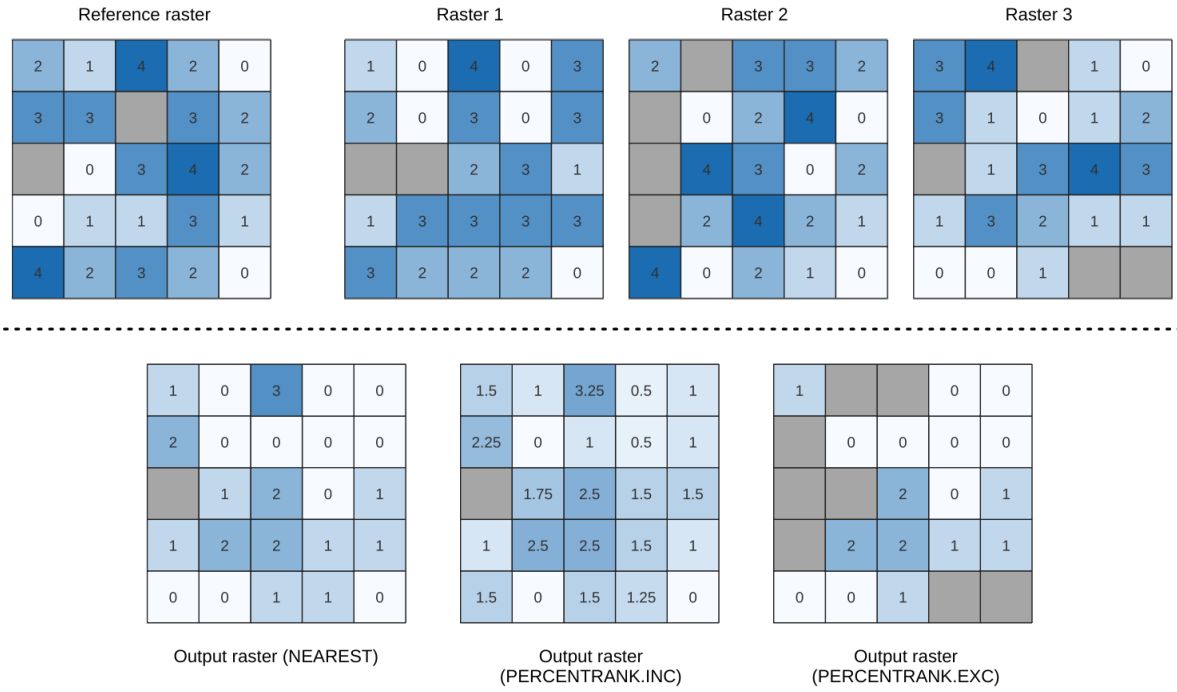


Figura24.50: Percentil = 0,25. As células "SemDado" (cinza) são ignoradas.

Ver também

Percentil de pilha celular, *Cell stack percentrank from raster layer*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Método	`METHOD`	[enumeração] Padrão: 0	Método para cálculo de percentil: • 0 — Classificação do mais próximo: retorna o valor que está mais próximo do percentil especificado • 1 — Interpolação linear inclusiva (PERCENTIL.INC) • 1 — Interpolação linear exclusiva (PERCENTIL.INC)
Percentil	`VALUE`	[numeric: double] Padrão: 0.25	Value to rank among the respective values in the stack of all overlaid and sorted cell values from the input rasters. Between 0 and 1.
Ignorar valores sem dados	`IGNORE_NODATA`	[boleano] Padrão: Verdadeiro	If unchecked, any NoData cells in the input layers will result in a NoData cell in the output raster
Camada de referência	`REFERENCE_LAYER`	[raster]	A camada de referência para a criação da camada de saída (extensão, SRC, dimensões dos pixels)
Camada de saída	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value	`OUTPUT_NODATA_VAL`	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:cellstackpercentile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Cell stack percentrank from raster layer

Calculates the cell-wise percentrank value of a stack of rasters based on an input value raster and writes them to an output raster.

At each cell location, the current value of the value raster is ranked among the respective values in the stack of all overlaid and sorted cell values of the input rasters. For values outside of the the stack value distribution, the algorithm returns NoData because the value cannot be ranked among the cell values.

Há dois métodos para o cálculo do percentil:

- Interpolação linear inclusiva (PERCENTRANK.INC)
- Interpolação linear exclusiva (PERCENTRANK.EXC)

Os métodos de interpolação linear retornam os valores únicos para diferentes percentis. Ambos os métodos de interpolação seguem seus métodos de contrapartida implementados pelo LibreOffice ou pelo Microsoft Excel.

The output raster's extent and resolution is defined by a reference raster. Input raster layers that do not match the cell size of the reference raster layer will be resampled using nearest neighbor resampling. NoData values in any of the input layers will result in a NoData cell output if the "Ignore NoData values" parameter is not set. The output raster data type will always be `Float32`.

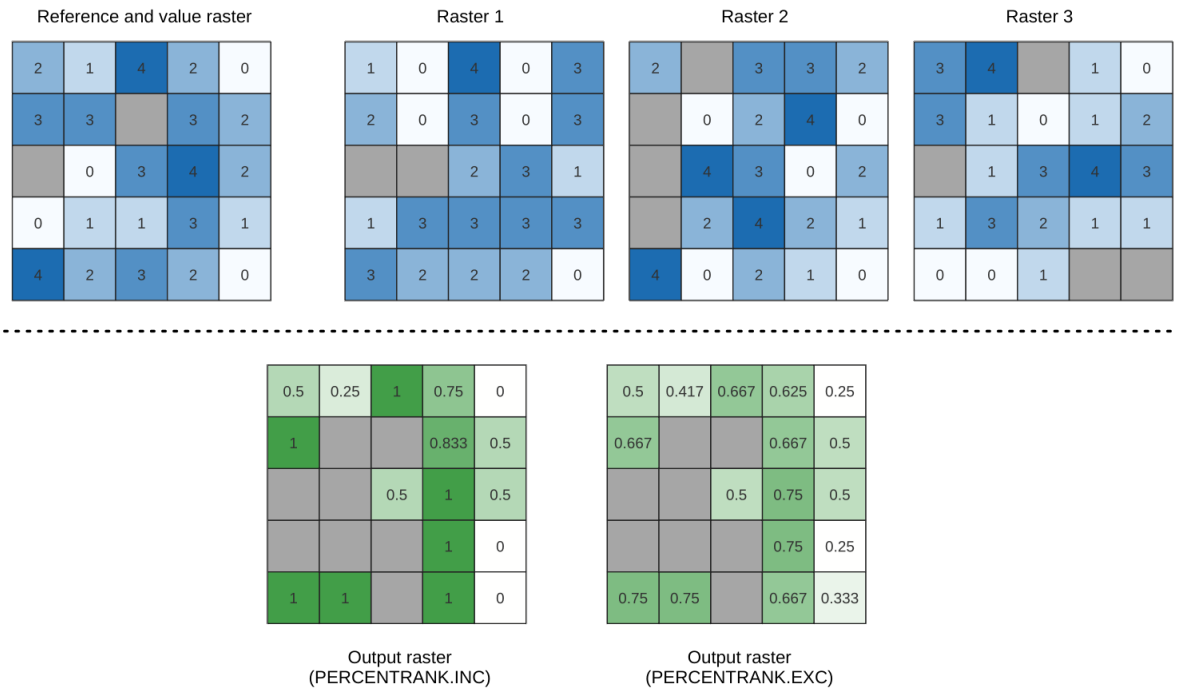


Figura24.51: Ranking the value raster layer cells. `NoData` cells (grey) are ignored.

Ver também

Percentil de pilha celular, *Porcentagem da pilha de células em relação ao valor*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Valor da camada raster	`INPUT_VALUE_RAST`	[raster]	The layer to rank the values among the stack of all overlaid layers
Faixa de valor raster	`VALUE_RASTER_B`	[numeric: integer] Padrão: 1	Banda da "camada de valor raster" para comparar com
Método	`METHOD`	[enumeração] Padrão: 0	Método para cálculo de percentil: • 0 — Interpolação linear inclusiva (PERCENTRANK.INC) • 1 — Interpolação linear exclusiva (PERCENTRANK.INC)
Ignorar valores sem dados	`IGNORE_NODATA`	[boleano] Padrão: Verdadeiro	If unchecked, any NoData cells in the input layers will result in a NoData cell in the output raster

continua na próxima página

Tabela 24.307 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Camada de referência	`REFERENCE_LAYER`	[raster]	A camada de referência para a criação da camada de saída (extensão, SRC, dimensões dos pixels)
Camada de saída	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value	`OUTPUT_NODATA_VAL`	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:cellstackpercentrankfromrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Estatísticas de células

Computes per-cell statistics based on input raster layers and for each cell writes the resulting statistics to an output raster. At each cell location, the output value is defined as a function of all overlaid cell values of the input rasters.

By default, a NoData cell in ANY of the input layers will result in a NoData cell in the output raster. If the *Ignore NoData values* option is checked, then NoData inputs will be ignored in the statistic calculation. This may result in NoData output for locations where all cells are NoData.

O parâmetro *Camada de referência* especifica uma camada raster existente a ser utilizada como referência ao criar o raster de saída. O raster de saída terá a mesma extensão, SRC, e dimensões de pixel que esta camada.

Calculation details: Input raster layers that do not match the cell size of the reference raster layer will be resampled using `nearest neighbor resampling`. The output raster data type will be set to the most complex data type present in the input datasets except when using the functions `Mean`, `Standard deviation` and `Variance` (data type is always `Float32` or `Float64` depending on input float type) or `Count` and `Variety` (data type is always `Int32`).

- `Count`: The count statistic will always result in the number of cells without NoData values at the current cell location.
- `Mediana`: Se o número de camadas de entrada for par, a mediana será calculada como a média aritmética dos dois valores centrais dos valores de entrada da célula ordenada.
- `Minority/Majority`: If no unique minority or majority could be found, the result is NoData, except all input cell values are equal.

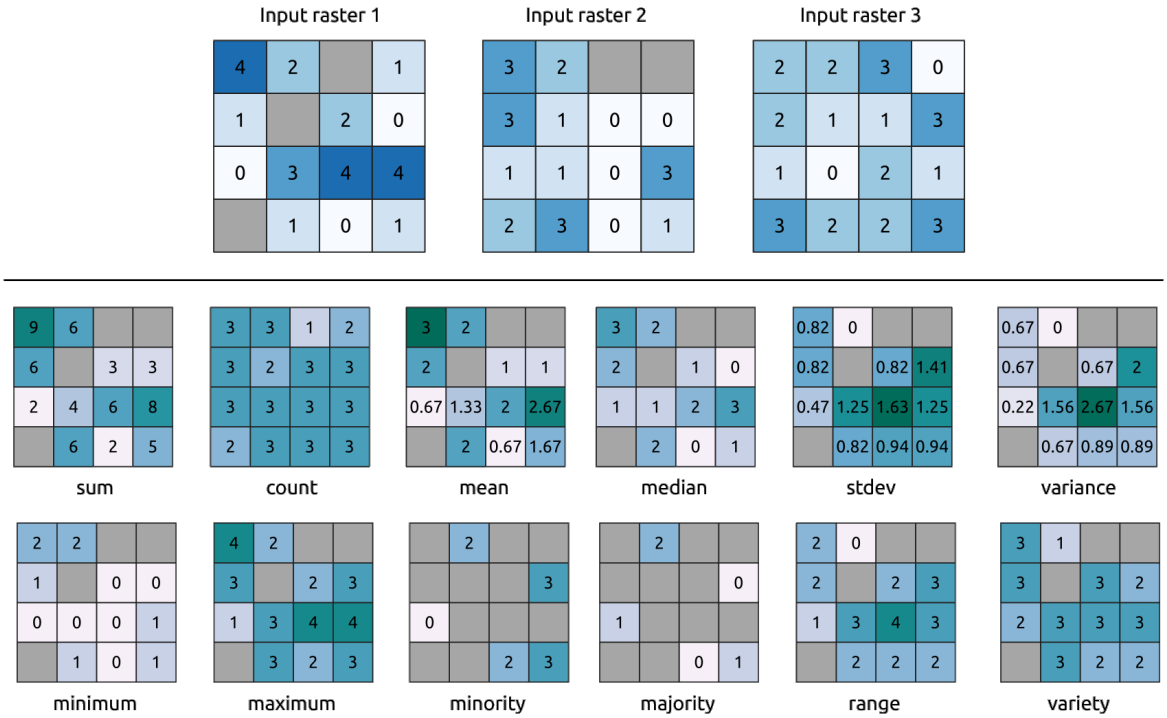


Figura24.52: Example with all the statistic functions. `NoData` cells (grey) are taken into account.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Camadas raster de entrada
Statistica	`STATISTIC`	[enumeração] Padrão: 0	Estatísticas disponíveis. Opções: • 0 — Soma • 1 — Contagem • 2 — Média • 3 — Mediana • 4 — Desvio padrão • 5 — Variação • 6 — Mínimo • 7 — Máximo • 8 — Minoria (valor mínimo comum) • 9 — Maioria (valor mais comum) • 10 — Faixa (max - min) • 11 — Variedade (contagem de valores únicos)
Ignorar valores sem dados	`IGNORE_NODATA`	[boleano] Padrão: Verdadeiro	Calculate statistics also for all cells stacks, ignoring NoData occurrence.
Camada de referência	`REF_LAYER`	[raster]	A camada de referência para criar a camada de saída (extensão, SRC, dimensões de pixel)

continua na próxima página

Tabela 24.310 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value Opcional	`OUTPUT_NO_DATA_VA`	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Identificador da autoridade SRC	`CRS_AUTHID`	[src]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Matriz de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster

Código Python

Algorithm ID: `native:cellstatistics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Igual a frequência

Evaluates on a cell-by-cell basis the frequency (number of times) the values of an input stack of rasters are equal to the value of a value layer. The output raster extent and resolution are defined by the input raster layer and is always of `Int32` type.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. The output NoData value can be set manually.

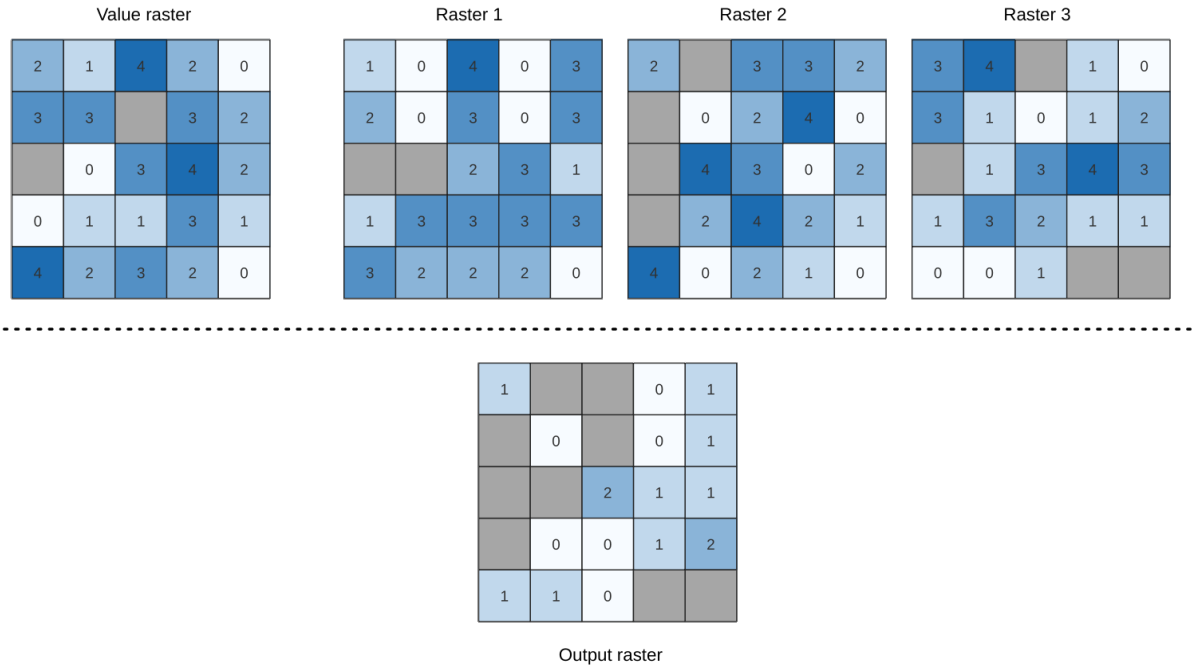


Figura24.53: Para cada célula na matriz de saída, o valor representa o número de vezes que as células correspondentes na lista de matrizes são iguais à matriz de valor. As células `SemDados` (cinza) são levadas em consideração.

Ver também

Maior do que a freqüência, *Menor do que a freqüência*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Valor de entrada raster	`INPUT_VALUE_RAST`	[raster]	A camada de valor de entrada serve como camada de referência para as camadas de amostra
Faixa de valor raster	`INPUT_VALUE_RAST`	[banda raster] Padrão: A primeira faixa da camada raster	Selecione a faixa que você deseja usar como amostra
Camadas de entrada raster	`INPUT_RASTERS`	[raster] [lista]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Ignorar valores sem dados	`IGNORE_NODATA`	[boleano] Padrão: Falso	If unchecked, any NoData cells in the value raster or the data layer stack will result in a NoData cell in the output raster
Camada de saída	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value Opcional	`OUTPUT_NO_DATA_VA`	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída

continua na próxima página

Tabela 24.315 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Contagem de células com ocorrências de valor igual	`FOUND_LOCATION`	[numeric: integer]	
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster
Mean frequency at valid cell locations	`MEAN_FREQUENCY`	[numeric: double]	
Contagem de ocorrências de valor	`OCCURRENCE_COUNT`	[numeric: integer]	
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster

Código Python

ID do algoritmo: `native:equaltofrequency`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Fuzzify raster (gaussian membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Gaussian membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The gaussian membership function is defined as $\mu(x) = e^{-f1*(x-f2)^2}$, where *f1* is the spread and *f2* the midpoint.

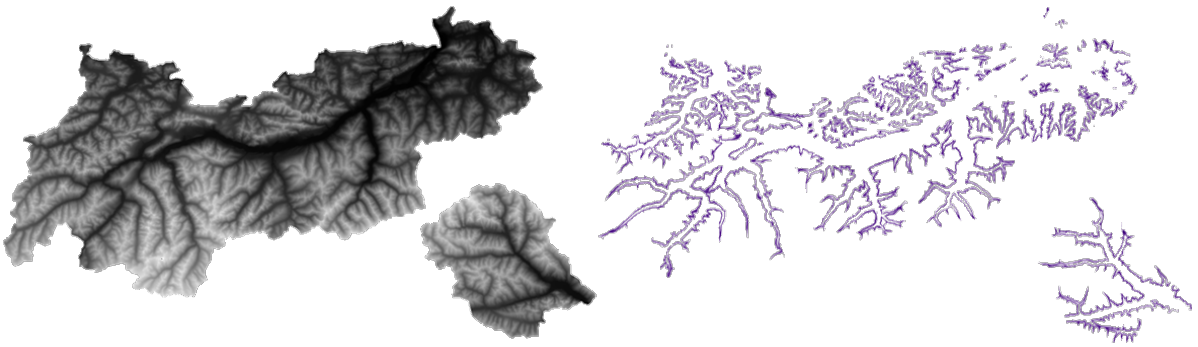


Figura24.54: Fuzzify raster example. Input raster source: Land Tirol - data.tirol.gv.at.

Ver também

Fuzzify raster (large membership), *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Entrada raster	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada raster	If the raster is multiband, choose the band that you want to fuzzify.
Função ponto médio	`FUZZYMIDPOINT`	[numeric: double] Padrão: 10.0	Ponto médio da função gaussiana
Função de espalhamento	`FUZZYSPREAD`	[numeric: double] Padrão: 0,01	Difusão da função gaussiana
Fuzzified raster	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ”	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Fuzzified raster	`OUTPUT`	[o mesmo que entrada]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[src]	O sistema de referência de coordenadas da camada raster de saída

continua na próxima página

Tabela 24.318 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TO-TAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:fuzzifyrastergaussianmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Fuzzify raster (large membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Large membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The large membership function is defined as $\mu(x) = \frac{1}{1+\left(\frac{x}{f2}\right)^{-f1}}$, where *f1* is the spread and *f2* the midpoint.

Ver também

Fuzzify raster (gaussian membership), *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Entrada raster	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada raster	If the raster is multiband, choose the band that you want to fuzzify.
Função ponto médio	`FUZZYMIDPOINT`	[numeric: double] Padrão: 50.0	Midpoint of the large function
Função de espalhamento	`FUZZYSPREAD`	[numeric: double] Padrão: 5.0	Espalhamento da grande função

continua na próxima página

Tabela 24.319 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Fuzzified raster	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Fuzzified raster	`OUTPUT`	[o mesmo que entrada]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[src]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:fuzzifyrasterlargemembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Fuzzify raster (linear membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Linear membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The linear function is defined as $\mu(X)\begin{cases}0 & x \leq a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & x \geq b\end{cases}$, where a is the low bound and b the high bound. This equation assigns membership values using a linear transformation for pixel values between the low and high bounds. Pixels values smaller than the low bound are given 0 membership whereas pixel values greater than the high bound are given 1 membership.

Ver também

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Entrada raster	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada raster	If the raster is multiband, choose the band that you want to fuzzify.
Baixo nível de adesão	`FUZZYLOWBOUND`	[numeric: double] Padrão: 0.0	Baixo limite da função linear
High fuzzy membership bound	`FUZZYHIGHBOUND`	[numeric: double] Padrão: 1,0	Alto limite da função linear
Fuzzified raster	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ”	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Fuzzified raster	`OUTPUT`	[o mesmo que entrada]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[src]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:fuzzifyrasterlinearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Fuzzify raster (near membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Near membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The near membership function is defined as $\mu(x) = \frac{1}{1 + f1 * (x - f2)^2}$, where *f1* is the spread and *f2* the midpoint.

Ver também

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)*, *Fuzzify raster (linear membership)*, *Fuzzify raster (power membership)*, *Fuzzify raster (small membership)*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Entrada raster	`INPUT`	[raster]	Camada raster de entrada

continua na próxima página

Tabela 24.325 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada raster	If the raster is multiband, choose the band that you want to fuzzify.
Função ponto médio	`FUZZYMIDPOINT`	[numeric: double] Padrão: 50.0	Ponto médio da função próxima
Função de espalhamento	`FUZZYSPREAD`	[numeric: double] Padrão: 0,01	Espalhamento da função próxima
Fuzzified raster	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Fuzzified raster	`OUTPUT`	[o mesmo que entrada]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[src]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:fuzzifyrasternearmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Fuzzify raster (power membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Power membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The power function is defined as $\mu(x)\begin{cases}0 & x \leq a \\ \left(\frac{x-a}{b-a}\right)^{f1} & a < x < b \\ 1 & x \geq b\end{cases}$, where a is the low bound, b is the high bound, and $f1$ the exponent. This equation assigns membership values using the power transformation for pixel values between the low and high bounds. Pixels values smaller than the low bound are given 0 membership whereas pixel values greater than the high bound are given 1 membership.

Ver também

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)*, *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (small membership)*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Entrada raster	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada raster	If the raster is multiband, choose the band that you want to fuzzify.
Baixo nível de adesão	`FUZZYLOWBOUND`	[numeric: double] Padrão: 0.0	Low bound of the power function
High fuzzy membership bound	`FUZZYHIGHBOUND`	[numeric: double] Padrão: 1,0	High bound of the power function
High fuzzy membership bound	`FUZZYEXPONENT`	[numeric: double] Padrão: 2.0	Expoente da função de potência
Fuzzified raster	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Fuzzified raster	`OUTPUT`	[o mesmo que entrada]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[src]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:fuzzifyrasterpowermembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Fuzzify raster (small membership)

Transforms an input raster to a fuzzified raster by assigning a membership value to each pixel, using a Small membership function. Membership values range from 0 to 1. In the fuzzified raster, a value of 0 implies no membership of the defined fuzzy set, whereas a value of 1 means full membership. The small membership function is defined as $\mu(x) = \frac{1}{1 + \left(\frac{x}{f2}\right)^{f1}}$, where *f1* is the spread and *f2* the midpoint.

Ver também

Fuzzify raster (gaussian membership), *Fuzzify raster (large membership)* *Fuzzify raster (linear membership)*, *Fuzzify raster (near membership)*, *Fuzzify raster (power membership)*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Entrada raster	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada raster	If the raster is multiband, choose the band that you want to fuzzify.
Função ponto médio	`FUZZYMIDPOINT`	[numeric: double] Padrão: 50.0	Ponto médio da função pequena
Função de espalhamento	`FUZZYSPREAD`	[numeric: double] Padrão: 5.0	Espalhamento da pequena função
Fuzzified raster	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ‘’	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Fuzzified raster	`OUTPUT`	[o mesmo que entrada]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[src]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:fuzzifyrastersmallmembership`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Maior do que a freqüência

Evaluates on a cell-by-cell basis the frequency (number of times) the values of an input stack of rasters are equal to the value of a value raster. The output raster extent and resolution is defined by the input raster layer and is always of `Int32` type.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. The output NoData value can be set manually.

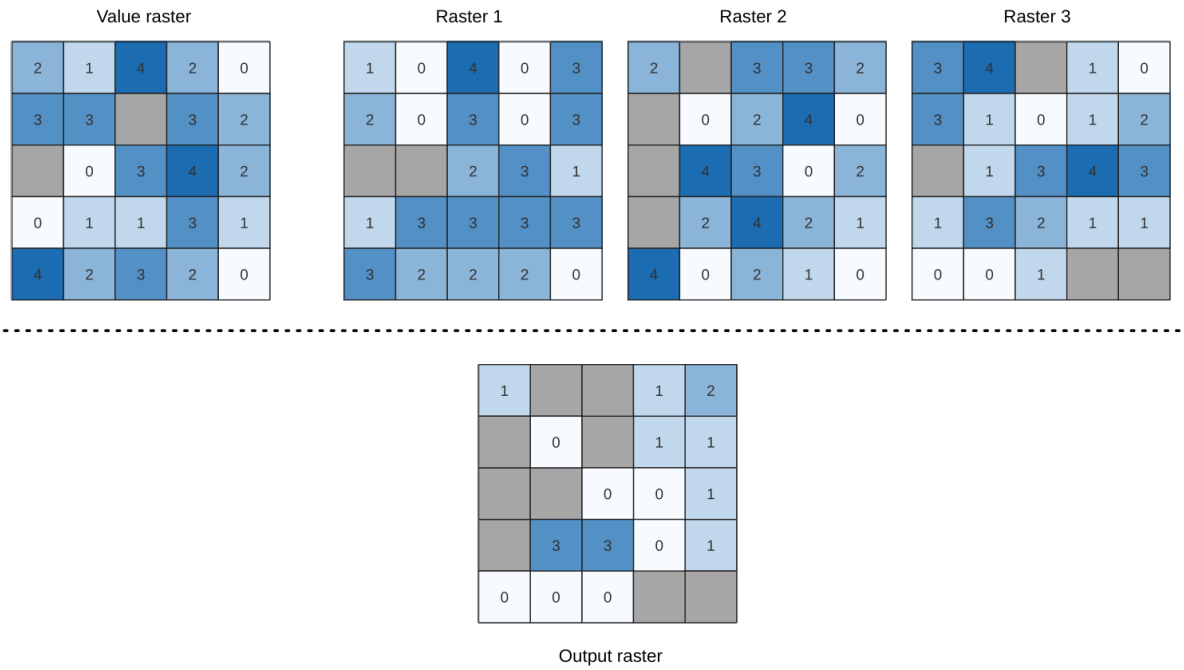


Figura24.55: For each cell in the output raster, the value represents the number of times that the corresponding cells in the list of rasters are greater than the value raster. `NoData` cells (grey) are taken into account.

Ver também

ref:`qgisigualaFreqüência', :ref:`qgisMenosQueFreqüência'

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Valor de entrada raster	`INPUT_VALUE_RAST`	[raster]	A camada de valor de entrada serve como camada de referência para as camadas de amostra
Faixa de valor raster	`INPUT_VALUE_RAST`	[banda raster] Padrão: A primeira faixa da camada raster	Selecione a faixa que você deseja usar como amostra
Camadas de entrada raster	`INPUT_RASTERS`	[raster] [lista]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Ignorar valores sem dados	`IGNORE_NODATA`	[boleano] Padrão: Falso	If unchecked, any NoData cells in the value raster or the data layer stack will result in a NoData cell in the output raster

continua na próxima página

Tabela 24.334 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value Opcional	`OUTPUT_NO_DATA_VA`	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Contagem de células com ocorrências de valor igual	`FOUND_LOCATION`	[numeric: integer]	
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster
Mean frequency at valid cell locations	`MEAN_FREQUENCY`	[numeric: double]	
Contagem de ocorrências de valor	`OCCURRENCE_COUNT`	[numeric: integer]	
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster

Código Python

Algorithm ID: `native:greaterthanfrequency`

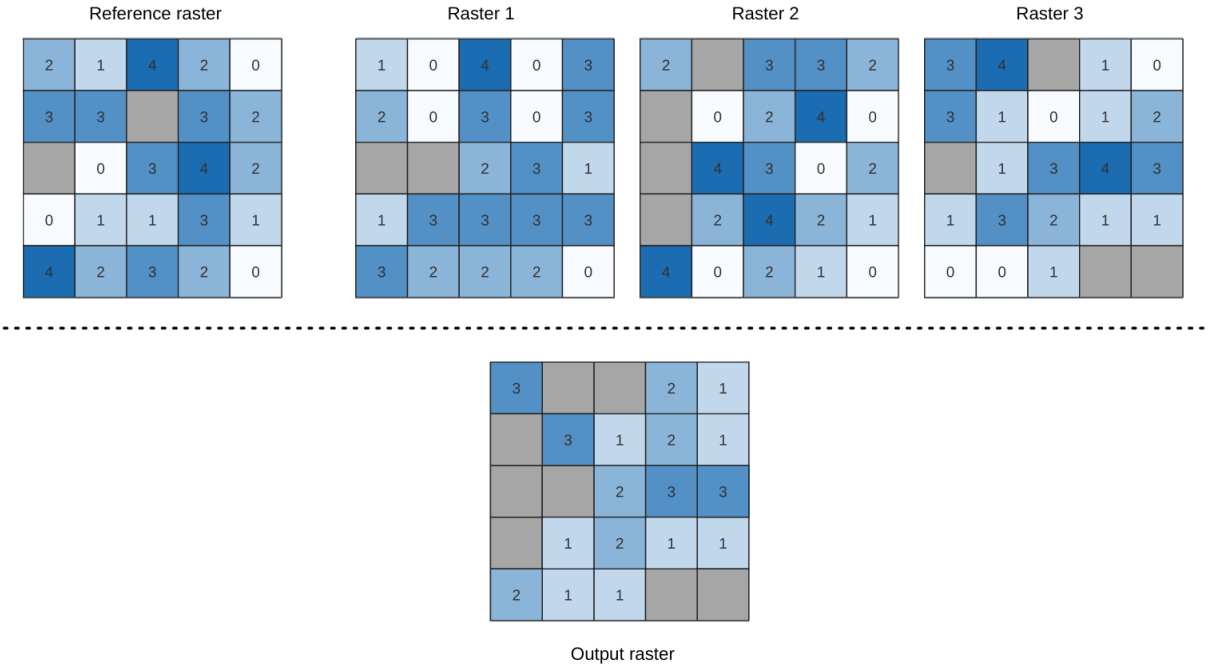
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Posição mais alta na pilha raster

Evaluates on a cell-by-cell basis the position of the raster with the highest value in a stack of rasters. Position counts start with 1 and range to the total number of input rasters. The order of the input rasters is relevant for the algorithm. If multiple rasters feature the highest value, the first raster will be used for the position value.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. Any NoData cells in the raster layer stack will result in a NoData cell in the output raster unless the "ignore NoData" parameter is checked. The output NoData value can be set manually. The output rasters extent and resolution is defined by a reference raster layer and is always of `Int32` type.



Ver também

Posição mais baixa na pilha rasterizada

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada raster	`INPUT_RASTERS`	[raster] [lista]	Lista de camadas rasterizadas para comparar com
Camada de referência	`REFERENCE_LAYER`	[raster]	A camada de referência para a criação da camada de saída (extensão, SRC, dimensões dos pixels)
Ignorar valores sem dados	`IGNORE_NODATA`	[boleano] Padrão: Falso	If unchecked, any NoData cells in the data layer stack will result in a NoData cell in the output raster
Camada de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster containing the result. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value	`OUTPUT_NODATA_VAL`	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:highestpositioninrasterstack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Menor do que a freqüência

Evaluates on a cell-by-cell basis the frequency (number of times) the values of an input stack of rasters are less than the value of a value raster. The output raster extent and resolution is defined by the input raster layer and is always of `Int32` type.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. The output NoData value can be set manually.

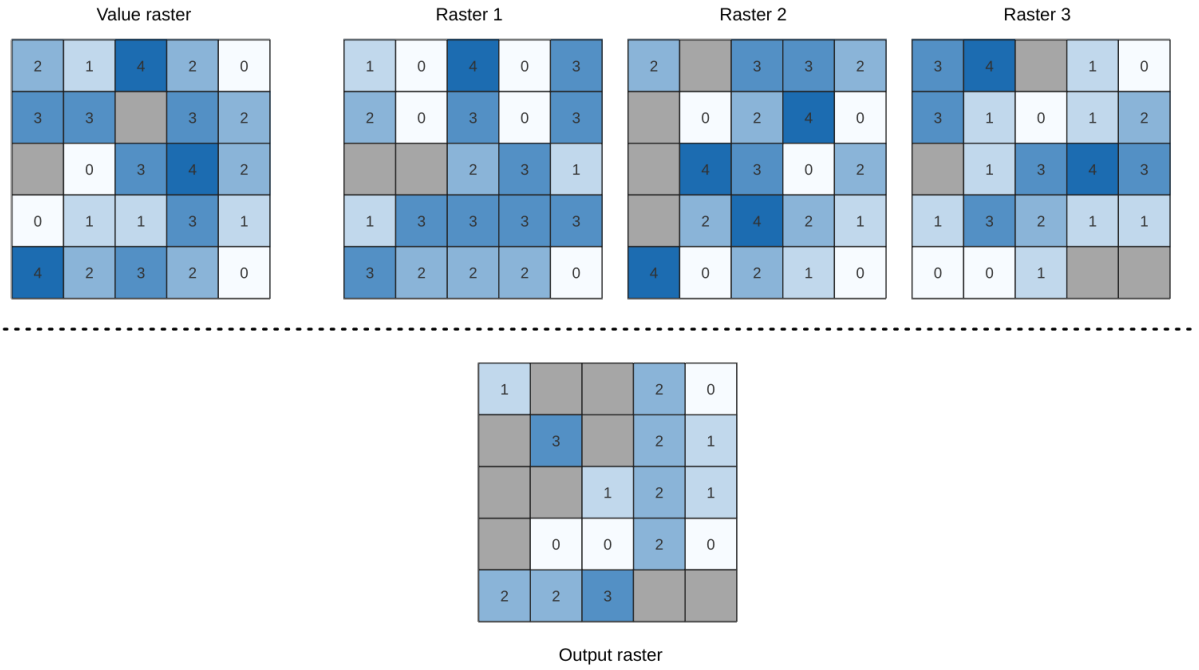


Figura24.56: For each cell in the output raster, the value represents the number of times that the corresponding cells in the list of rasters are less than the value raster. `NoData` cells (grey) are taken into account.

Ver também

Igual a frequência, *Maior do que a freqüência*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Valor de entrada raster	`INPUT_VALUE_RAST`	[raster]	A camada de valor de entrada serve como camada de referência para as camadas de amostra
Faixa de valor raster	`INPUT_VALUE_RAST`	[banda raster] Padrão: A primeira faixa da camada raster	Selecione a faixa que você deseja usar como amostra
Camadas de entrada raster	`INPUT_RASTERS`	[raster] [lista]	Raster layers to evaluate. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters
Ignorar valores sem dados	`IGNORE_NODATA`	[boleano] Padrão: Falso	If unchecked, any NoData cells in the value raster or the data layer stack will result in a NoData cell in the output raster
Camada de saída	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value Opcional	`OUTPUT_NO_DATA_VA`	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída

continua na próxima página

Tabela 24.342 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Contagem de células com ocorrências de valor igual	`FOUND_LOCATION`	[numeric: integer]	
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster
Mean frequency at valid cell locations	`MEAN_FREQUENCY`	[numeric: double]	
Contagem de ocorrências de valor	`OCCURRENCE_COUNT`	[numeric: integer]	
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster

Código Python

Algorithm ID: `native:lessthanfrequency`

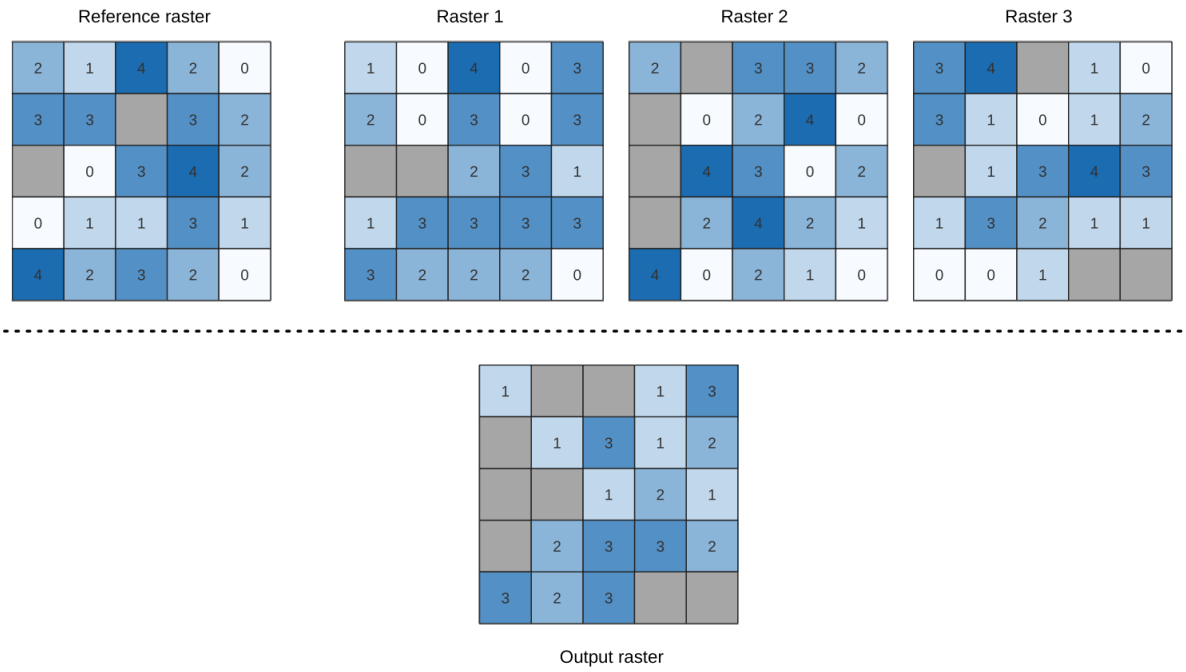
```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Posição mais baixa na pilha rasterizada

Evaluates on a cell-by-cell basis the position of the raster with the lowest value in a stack of rasters. Position counts start with 1 and range to the total number of input rasters. The order of the input rasters is relevant for the algorithm. If multiple rasters feature the lowest value, the first raster will be used for the position value.

If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters - use GDAL to use other bands in the analysis. Any NoData cells in the raster layer stack will result in a NoData cell in the output raster unless the “ignore NoData” parameter is checked. The output NoData value can be set manually. The output rasters extent and resolution is defined by a reference raster layer and is always of `Int32` type.



Ver também

Posição mais alta na pilha raster

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada raster	`INPUT_RASTERS`	[raster] [lista]	Lista de camadas rasterizadas para comparar com
Camada de referência	`REFERENCE_LAYER`	[raster]	A camada de referência para a criação da camada de saída (extensão, SRC, dimensões dos pixels)
Ignorar valores sem dados	`IGNORE_NODATA`	[boleano] Padrão: Falso	If unchecked, any NoData cells in the data layer stack will result in a NoData cell in the output raster
Camada de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster containing the result. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value	OUTPUT_NODATA_VAL	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Creation options Opcional	CREATION_OPTIONS (for QGIS <= 3.42, this was CREATE_OPTIONS)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos ...). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	OUTPUT	[raster]	Camada raster de saída contendo o resultado
Identificador da autoridade SRC	CRS_AUTHID	[string]	O sistema de referência de coordenadas da camada raster de saída
Extensão	EXTENT	[string]	A extensão espacial da camada raster de saída
Largura em pixels	WIDTH_IN_PIXEL	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	HEIGHT_IN_PIXE	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	TOTAL_PIXEL_COUN	[numeric: integer]	A contagem de pixels na camada de saída raster

Código Python

Algorithm ID: `native:lowestpositioninrasterstack`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Raster Booleano E

Calculates the boolean `AND` for a set of input rasters. If all of the input rasters have a non-zero value for a pixel, that pixel will be set to `1` in the output raster. If any of the input rasters have `0` values for the pixel it will be set to `0` in the output raster.

The reference layer parameter specifies an existing raster layer to use as a reference when creating the output raster. The output raster will have the same extent, CRS, and pixel dimensions as this layer.

By default, a NoData pixel in ANY of the input layers will result in a NoData pixel in the output raster. If the *Treat NoData values as false* option is checked, then NoData inputs will be treated the same as a `0` input value.

Ver também

Raster Booleano OU

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Lista de camadas rasterizadas de entrada
Camada de referência	`REF_LAYER`	[raster]	A camada de referência para criar a camada de saída (extensão, SRC, dimensões de pixel)
Treat NoData values as false	`NO-DATA_AS_FALSE`	[boleano] Padrão: Falso	Treat NoData values in the input files as 0 when performing the operation
Camada de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster containing the result. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value	`NO_DATA`	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer

continua na próxima página

Tabela 24.347 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de dados de saída	DATA_TYPE	[enumeração] Padrão: 5	Tipo de dados raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — Int32 (Thirty two bit signed integer (qint32)) • 4 — UInt32 (Thirty two bit unsigned integer (quint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Creation options Opcional	CREATION_OPTIONS (for QGIS <= 3.42, this was CREATE_OPTIONS)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Extensão	EXTENT	[string]	A extensão espacial da camada raster de saída
Identificador da autoridade SRC	CRS_AUTHID	[src]	O sistema de referência de coordenadas da camada raster de saída
Largura em pixels	WIDTH_IN_PIXEL	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	HEIGHT_IN_PIXE	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	TOTAL_PIXEL_COUN	[numeric: integer]	A contagem de pixels na camada de saída raster
NoData pixel count	NODATA_PIXEL_COU	[numeric: integer]	The count of NoData pixels in the output raster layer
Contagem de pixels verdadeiros	"CONTAGEM DE PIXELS VERDADEIROS"	[numeric: integer]	A contagem de pixels verdadeiros (valor = 1) na camada raster de saída

continua na próxima página

Tabela 24.348 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
"Contagem de pixels falsos"	"CONTAGEM DE PIXELS FALSOS"	[numeric: integer]	A contagem de pixels falsos (valor = 0) na camada raster de saída
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado

Código Python

Algorithm ID: `native:rasterbooleanand`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Raster Booleano OU

Calculates the boolean `OR` for a set of input rasters. If all of the input rasters have a zero value for a pixel, that pixel will be set to `0` in the output raster. If any of the input rasters have `1` values for the pixel it will be set to `1` in the output raster.

The reference layer parameter specifies an existing raster layer to use as a reference when creating the output raster. The output raster will have the same extent, CRS, and pixel dimensions as this layer.

By default, a NoData pixel in ANY of the input layers will result in a NoData pixel in the output raster. If the *Treat NoData values as false* option is checked, then NoData inputs will be treated the same as a `0` input value.

Ver também

Raster Booleano E

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Lista de camadas rasterizadas de entrada
Camada de referência	`REF_LAYER`	[raster]	A camada de referência para criar a camada de saída (extensão, SRC, dimensões de pixel)
Treat NoData values as false	`NO-DATA_AS_FALSE`	[boleano] Padrão: Falso	Treat NoData values in the input files as 0 when performing the operation
Camada de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster containing the result. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value	NO_DATA	[numeric: double] Padrão: -9999.0	Value to use for NoData in the output layer
Tipo de dados de saída	DATA_TYPE	[enumeração] Padrão: 5	Tipo de dados raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — Int32 (Thirty two bit signed integer (qint32)) • 4 — UInt32 (Thirty two bit unsigned integer (quint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Creation options Opcional	CREATION_OPTIONS (for QGIS <= 3.42, this was CREATE_OPTIONS)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Extensão	EXTENT	[string]	A extensão espacial da camada raster de saída
Identificador da autoridade SRC	CRS_AUTHID	[src]	O sistema de referência de coordenadas da camada raster de saída
Largura em pixels	WIDTH_IN_PIXEL	[numeric: integer]	O número de colunas na camada de saída raster
Altura em pixels	HEIGHT_IN_PIXE	[numeric: integer]	O número de linhas na camada de saída raster
Contagem total de pixels	TOTAL_PIXEL_COUN	[numeric: integer]	A contagem de pixels na camada de saída raster

continua na próxima página

Tabela 24.351 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
NoData pixel count	`NO-DATA_PIXEL_COU`	[numeric: integer]	The count of NoData pixels in the output raster layer
Contagem de pixels verdadeiros	"CONTAGEM DE PIXELS VERDADEIROS"	[numeric: integer]	A contagem de pixels verdadeiros (valor = 1) na camada raster de saída
"Contagem de pixels falsos"	"CONTAGEM DE PIXELS FALSOS"	[numeric: integer]	A contagem de pixels falsos (valor = 0) na camada raster de saída
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída contendo o resultado

Código Python

Algorithm ID: `native:rasterbooleanor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Calculadora Raster

Executa operações algébricas usando camadas matriciais.

The resulting layer will have its values computed according to an expression. The expression can contain numerical values, operators and references to any of the layers in the current project.

Ver também

Raster calculator (virtual), *Calculadora Raster*, *Calculadora raster*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Lista de camadas rasterizadas de entrada
Expressão	`EXPRESSION`	[expressão]	*Raster-based expression* that will be used to calculate the output raster layer.

continua na próxima página

Tabela 24.352 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Extensão da produção Opcional	`EXTENT`	[extensão]	Especifique a extensão espacial da camada rasterizada de saída. Se a extensão não for especificada, será utilizada a extensão mínima que cobre todas as camadas de referência selecionadas. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Output cell size (leave empty to set automatically) Opcional	`CELLSIZE`	[numeric: double]	Cell size of the output raster layer. If the cell size is not specified, the minimum cell size of the selected reference layer(s) will be used. The cell size will be the same for the X and Y axes.
SRC de Saída Opcional	`CRS`	[src]	SRC da camada raster da saída. Se o SRC de saída não for especificado, será usado o SRC da primeira camada de referência.
Calculado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS`	[string] Padrão: ”	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Calculado	`OUTPUT`	[raster]	Arquivo raster de saída com os valores calculados.

Código Python

Algorithm ID: `native:rastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Raster calculator (virtual)

Performs algebraic operations using raster layers and generates in-memory result.

The resulting layer will have its values computed according to an expression. The expression can contain numerical values, operators and references to any of the layers in the current project.

A virtual raster layer is a raster layer defined by its URI and whose pixels are calculated on-the-fly. It's not a new file on disk; the virtual layer is still connected to the rasters used in the calculation meaning that deleting or moving these rasters would break it. A *Layer name* can be provided, otherwise the calculation expression is used as such. Removing the virtual layer from the project deletes it, and it can be made persistent in file using the layer *Export* ► *Save as…* contextual menu.

Ver também

Calculadora Raster, *Calculadora Raster*, *Calculadora raster*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Lista de camadas rasterizadas de entrada
Expressão	`EXPRESSION`	[expressão]	*Raster-based expression* that will be used to calculate the output raster layer.

continua na próxima página

Tabela 24.355 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Extensão da produção Opcional	`EXTENT`	[extensão]	Especifique a extensão espacial da camada rasterizada de saída. Se a extensão não for especificada, será utilizada a extensão mínima que cobre todas as camadas de referência selecionadas. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Output cell size (leave empty to set automatically) Opcional	`CELLSIZE`	[numeric: double]	Cell size of the output raster layer. If the cell size is not specified, the minimum cell size of the selected reference layer(s) will be used. The cell size will be the same for the X and Y axes.
SRC de Saída Opcional	`CRS`	[src]	SRC da camada raster da saída. Se o SRC de saída não for especificado, será usado o SRC da primeira camada de referência.
Output layer name Opcional	`LAYER_NAME`	[string]	The name to assign to the generated layer. If not set, the text of the calculation expression is used.

Saídas

Etiqueta	Nome	Tipo	Descrição
Calculado	`OUTPUT`	[raster]	Output virtual raster layer with the calculated values.

Código Python

Algorithm ID: `native:virtualrastercalc`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Propriedades da camada rasterizada

Returns basic properties of the given raster layer, including the extent, size in pixels and dimensions of pixels (in map units), number of bands, and NoData value.

Este algoritmo destina-se a ser usado como um meio de extrair estas propriedades úteis para usar como valores de entrada para outros algoritmos em um modelo - por exemplo, para permitir a passagem dos tamanhos de pixel de um raster existente para um algoritmo raster GDAL.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Número da banda Opcional	`BAND`	[banda raster] Padrão: Não definido	Whether to also return properties of a specific band. If a band is specified, the noData value for the selected band is also returned.

Saídas

Etiqueta	Nome	Tipo	Descrição
Número de bandas no raster	`BAND_COUNT`	[numeric: integer]	O número de bandas na matriz
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão da camada raster no SRC
Band has a NoData value set	`HAS_NODATA_VAL`	[Boleano]	Indicates whether the raster layer has a value set for NoData pixels in the selected band
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de colunas na camada matriz
"Banda sem dados de valores"	`NODATA_VALUE`	[numeric: double]	The value (if set) of the NoData pixels in the selected band
Tamanho Pixel (altura) em unidades do mapa	`PIXEL_HEIGHT`	[numeric: integer]	Tamanho vertical em unidades do mapa do pixel
Tamanho Pixel (largura) em unidades do mapa	`PIXEL_WIDTH`	[numeric: integer]	Tamanho horizontal em unidades de mapa do pixel
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de linhas na camada raster
Coordenada x máxima	`X_MAX`	[numeric: double]	
Coordenada x mínima	`X_MIN`	[numeric: double]	
Coordenada y máxima	`Y_MAX`	[numeric: double]	
Coordenada y-mínima	`Y_MIN`	[numeric: double]	

Código Python

Algorithm ID: `native:rasterlayerproperties`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Estatísticas da camada rasterizada

Calculates basic statistics from the values in a given band of the raster layer. The output is loaded in the *Processing* ► *Results viewer* menu.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada de entrada	Se o raster for multibanda, escolha a banda para a qual deseja obter estatísticas.
Estatísticas	`OUTPUT_HTML_FILE`	[html] Padrão: `[Salvar em arquivo temporário]`	Especificação do arquivo de saída: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Valor máximo	`MAX`	[numeric: double]	
Valor médio	`MEAN`	[numeric: double]	
Valor mínimo	`MIN`	[numeric: double]	

continua na próxima página

Tabela 24.360 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Estatísticas	`OUTPUT_HTML_FILE`	[html]	O arquivo de saída contém as seguintes informações: • Arquivo analisado: caminho da camada raster • Valor mínimo: valor mínimo do raster • Valor máximo: valor máximo do raster • Faixa de valores: diferença entre os valores máximo e mínimo • Soma: soma total dos valores • Valor médio: média dos valores • Desvio padrão: desvio padrão dos valores • Soma dos quadrados: soma das diferenças quadráticas de cada observação a partir da média geral
Intervalo	`RANGE`	[numeric: double]	
Desvio padrão	`STD_DEV`	[numeric: double]	
Soma	`SUM`	[numeric: double]	
Soma dos quadrados	`SUM_OF_SQUARES`	[numeric: double]	

Código Python

Algorithm ID: `native:rasterlayerstatistics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Relatório de valores únicos da camada raster

Returns the count and area of each unique value in a given raster layer. The calculation of the area is done in the area unit of the layer's CRS.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada de entrada	Se o raster for multibanda, escolha a banda para a qual deseja obter estatísticas.

continua na próxima página

Tabela 24.361 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Relatório de valores únicos	`OUTPUT_HTML_FILE`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Especificação do arquivo de saída: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…
Tabela de valores únicos	`OUTPUT_TABLE`	[vector: table] Default: `[Pular saída]`	Especificação da tabela para valores únicos: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
NoData pixel count	`NODATA_PIXEL_COU`	[numeric: integer]	The number of NoData pixels in the output raster layer
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster
Relatório de valores únicos	`OUTPUT_HTML_FILE`	[html]	O arquivo HTML de saída contém as seguintes informações: • Arquivo analisado: o caminho da camada raster • Extensão: coordenadas xmin, ymin, xmax, ymax da extensão • Projeção: projeção da camada • Largura em pixels: número de colunas e tamanho de píxeis • Altura em pixels: número de linhas e largura de pixels • Contagem total de pixels: contagem de todos os pixels • NoData pixel count: count of pixels with NoData value

continua na próxima página

Tabela 24.362 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tabela de valores únicos	`OUTPUT_TABLE`	[vector: table]	Uma tabela com três colunas: • *valor*: valor do pixel • *contagem*: contagem de pixels com este valor • *m2* or *deg2* or *ft2* or … : total area of pixels with this value. The column name depends on the area unit of the layer's CRS and the calculation is done in that unit.
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster

Código Python

Algorithm ID: `native:rasterlayeruniquevaluesreport`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Estatísticas zonais da camada rasterizada

Calcula estatísticas para os valores de uma camada raster, categorizados por zonas definidas em outra camada raster.

Ver também

Estatísticas zonais

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada raster	Se o raster for multibanda, escolha a banda para a qual deseja calcular as estatísticas.
** Camada de zonas**	`ZONES`	[raster]	Camada raster definindo zonas. As zonas são dadas por pixels contíguos com o mesmo valor de pixel.

continua na próxima página

Tabela 24.363 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Número da banda das zonas	`ZONES_BAND`	[banda raster] Padrão: A primeira faixa da camada raster	Se o raster for multibandas, escolha a banda que define as zonas
Estatísticas	`OUTPUT_TABLE`	[vector: table] Padrão: `[Criar camada temporária]`	Specification of the output report. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Camada de referência Opcional	`REF_LAYER`	[enumeração] Padrão: 0	Camada raster usada para calcular os centróides que serão usados como referência ao determinar as zonas na camada de saída. Um de: • 0 — Camada de entrada: as zonas são determinadas pela amostragem do valor da camada raster da zona no centroide de cada pixel da camada raster de origem • 1 — Camada de zonas: a camada raster de entrada será amostrada no centroide de cada pixel da camada raster de zonas

Saídas

Etiqueta	Nome	Tipo	Descrição
Identificador da autoridade SRC	`CRS_AUTHID`	[string]	O sistema de referência de coordenadas da camada raster de saída
Extensão	`EXTENT`	[string]	A extensão espacial da camada raster de saída
Altura em pixels	`HEIGHT_IN_PIXE`	[numeric: integer]	O número de linhas na camada de saída raster
NoData pixel count	`NODATA_PIXEL_COU`	[numeric: integer]	The number of NoData pixels in the output raster layer

continua na próxima página

Tabela 24.365 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Estatísticas	`OUTPUT_TABLE`	[vector: table]	A camada de saída contém as seguintes informações **para cada zona**: • Área: a área em unidades quadradas raster na zona; • Soma: a soma total dos valores dos pixels na zona; • Contagem: o número de pixels na zona; • Min: o valor mínimo de píxeis na zona; • Max: o valor máximo de pixel na zona; • Média: a média dos valores de pixel na zona;
Contagem total de pixels	`TOTAL_PIXEL_COUN`	[numeric: integer]	A contagem de pixels na camada de saída raster
Largura em pixels	`WIDTH_IN_PIXEL`	[numeric: integer]	O número de colunas na camada de saída raster

Código Python

Algorithm ID: `native:rasterlayerzonalstats`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Raster minimum/maximum

Extracts the minimum and maximum pixel values from a specified band of a raster layer. If multiple pixels share the same minimum or maximum value, only one of them will be included in the output.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Raster layer from which to extract the minimum and maximum pixel values.
Número da banda	`BAND`	[banda raster] Padrão: 1	The band of the raster to analyze. If the raster is multiband, specify the band number (starting from 1).
Extract	`EXTREMA`	[enumeração] Default: 0 (Minimum and Maximum)	Choose which extrema to extract: • 0: Minimum and Maximum • 1: Minimum only • 2: Maximum only

continua na próxima página

Tabela 24.366 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specification of the output layer. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[vetor: ponto]	Vector layer with point features at the locations of the minimum and/or maximum pixel values.

Código Python

Algorithm ID: `native:rasterminmax`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Raster rank

`NEW in 3.44`

Performs a cell-by-cell analysis in which output values match the rank of a sorted list of overlapping cell values from input layers. The output raster will be multi-band if multiple ranks are provided. If multiband rasters are used in the data raster stack, the algorithm will always perform the analysis on the first band of the rasters.

Ver também

Porcentagem da pilha de células em relação ao valor, *Percentil de pilha celular*, *Cell stack percentrank from raster layer*, *Estatísticas de células*, *Calculadora Raster*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT_RASTERS`	[raster] [lista]	Input raster layer(s) to calculate the rank from.
Rank	`RANKS`	[numeric: integer] [list] Default: [1]	List of ranks to calculate for each pixel in the sorted list of cells values from the input rasters. Cells whose values match the given rank(s) are kept in the output layer. The ranks are 1-based, meaning that the first rank is 1. Negative values are also supported.
NoData value handling	`NODATA_HANDLING`	[enumeração] Padrão: 0	Choose how to handle NoData values in the input raster: • 0 — Exclude NoData from values lists: will not include NoData values in the list of pixel values when calculating ranks. • 1 — Presence of NoData in a values list results in NoData output cells: the output raster will have NoData values in the corresponding locations.
Ranked	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster containing the rank values. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Extensão da produção Opcional	`EXTENT`	[extensão]	Especifique a extensão espacial da camada rasterizada de saída. Se a extensão não for especificada, será utilizada a extensão mínima que cobre todas as camadas de referência selecionadas. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Output cell size Opcional	`CELL_SIZE`	[numeric: double]	Cell size of the output raster layer. If the cell size is not specified, the minimum cell size of the selected reference layer(s) will be used. The cell size will be the same for the X and Y axes.
SRC de Saída Opcional	`CRS`	[src]	SRC da camada raster da saída. Se o SRC de saída não for especificado, será usado o SRC da primeira camada de referência.

Saídas

Etiqueta	Nome	Tipo	Descrição
Ranked	`OUTPUT`	[raster]	Output raster layer containing the result.

Código Python

Algorithm ID: `native:rasterrank`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Volume de superfície rasterizada

Calcula o volume sob uma superfície rasterizada em relação a um determinado nível de base. Isto é útil principalmente para os Modelos Digitais de Elevação (MDE).

Parâmetros

Etiqueta	Nome	Tipo	Descrição
**CAMADA de entrada””	`INPUT`	[raster]	Raster de entrada, representando uma superfície
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada raster	Se o raster for multibandas, escolha a banda que define a superfície
Nível base	`LEVEL`	[numeric: double] Padrão: 0.0	Definir um valor base ou de referência. Esta base é utilizada no cálculo do volume de acordo com o parâmetro `Método` (veja abaixo).
Método	`METHOD`	[enumeração] Padrão: 0	Define the method for the volume calculation given by the difference between the raster pixel value and the `Base level`. Options: • 0 — Contar apenas acima do nível base: apenas pixels acima do nível base irão adicionar ao volume. • 0 — Contar apenas abaixo do nível base: apenas pixels abaixo do nível base irão adicionar ao volume. • 2 — Subtrair volumes abaixo do nível de base: pixels acima do nível de base irão adicionar ao volume, pixels abaixo do nível de base irão subtrair do volume. • 3 — Adicionar Volumes Abaixo do Nível Base: Adicione o volume independentemente de o pixel estar acima ou abaixo do nível base. Isto é equivalente à soma dos valores absolutos da diferença entre o valor do pixel e o nível de base.
Relatório de volume de superfície	`OUTPUT_HTML_FILE`	[html] Padrão: `[Salvar em arquivo temporário]`	Specification of the output HTML report. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

continua na próxima página

Tabela 24.371 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tabela de volume de superfície	`OUTPUT_TABLE`	[vector: table] Default: `[Pular saída]`	Specification of the output table. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Volume	`VOLUME`	[numeric: double]	O volume calculado
Área	`AREA`	[numeric: double]	A área em unidades de mapa quadrado
Contagem_pixel	`PIXEL_COUNT`	[numeric: integer]	O número total de pixels que foram analisados
Relatório de volume de superfície	`OUTPUT_HTML_FILE`	[html]	O relatório de saída (contendo volume, área e contagem de pixels) em formato HTML
Tabela de volume de superfície	`OUTPUT_TABLE`	[vector: table]	A tabela de saída (contendo volume, área e contagem de pixels)

Código Python

Algorithm ID: `native:rastersurfacevolume`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Reclassificar por camada

Reclassificar uma banda matricial atribuindo novos valores de classe com base nos intervalos especificados em uma tabela de vetores.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada raster	`INPUT_RASTER`	[raster]	Camada raster para reclassificar
Número da banda	`RASTER_BAND`	[banda raster] Padrão: A primeira faixa da camada raster	Se o raster for multibandas, escolha a banda que você deseja reclassificar.
Camada que contém quebras de classe	`INPUT_TABLE`	[vetor: qualquer]	Camada vetorial contendo os valores a serem usados para classificação.
Campo de valor mínimo da classe	`MIN_FIELD`	[campo de tabela: numérico]	Field with the minimum value of the range for the class. Use `-inf` to capture the lowest possible value.
Campo de valor máximo da classe	`MAX_FIELD`	[campo de tabela: numérico]	Field with the maximum value of the range for the class. Use `inf` to capture the highest possible value.
Campo de valor de saída	`VALUE_FIELD`	[campo de tabela: numérico]	Field with the value that will be assigned to the pixels that fall in the class (between the corresponding min and max values). Use `nan` to set the value of the range to NoData.
Raster reclassificado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value	`NO_DATA`	[numeric: double] Padrão: -9999.0	Value to apply to NoData values.
Limites de alcance	`RANGE_BOUNDARI`	[enumeração] Padrão: 0	Define regras de comparação para a classificação. Opções: • 0 — min < valor <= max • 1 — min <= valor < max • 2 — min <= valor <= max • 3 — min < valor < max
Use NoData when no range matches value	`NO-DATA_FOR_MISSI`	[boleano] Padrão: Falso	Applies the NoData value to band values that do not fall in any class. If False, the original value is kept.

continua na próxima página

Tabela 24.374 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o formato do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — Int32 (Thirty two bit signed integer (qint32)) • 4 — UInt32 (Thirty two bit unsigned integer (quint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Raster reclassificado	`OUTPUT`	[raster]	Camada raster de saída com valores de faixa reclassificados

Código Python

Algorithm ID: `native:reclassifybylayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Reclassificar por tabela

Reclassifies a raster band by assigning new class values based on the ranges specified in a fixed table.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada raster	`INPUT_RASTER`	[raster]	Camada raster para reclassificar
Número da banda	`RASTER_BAND`	[banda raster] Padrão: 1	Banda raster para a qual você deseja recalcular valores.
Tabela de reclassificação	`TABLE`	[vector: table]	A 3-columns table to fill with the values to set the boundaries of each class (`Minimum` and `Maximum`) and the new `Value` to assign to the band values that fall in the class. The value `-inf` can be used as minimum, `inf` as maximum, and `nan` can be used to set the output value to NoData.
Raster reclassificado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output NoData value	`NO_DATA`	[numeric: double] Padrão: -9999.0	Value to apply to NoData values.
Limites de alcance	`RANGE_BOUNDARI`	[enumeração] Padrão: 0	Define regras de comparação para a classificação. Opções: • 0 — min < valor <= max • 1 — min <= valor < max • 2 — min <= valor <= max • 3 — min < valor < max

continua na próxima página

Tabela 24.377 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Use NoData when no range matches value	`NO-DATA_FOR_MISSI`	[boleano] Padrão: Falso	Applies the NoData value to band values that do not fall in any class. If False, the original value is kept.
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o formato do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — Int32 (Thirty two bit signed integer (qint32)) • 4 — UInt32 (Thirty two bit unsigned integer (quint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Creation options Opcional	`CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `CRE-ATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Raster reclassificado	`OUTPUT`	[raster]	Camada raster de saída com valores de faixa reclassificados

Código Python

Algorithm ID: `native:reclassifybytable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Redimensionar matriz

Redimensiona a camada matricial para um novo intervalo de valores, preservando a forma (distribuição) do histograma da matriz (valores de pixel). Os valores de entrada são mapeados usando uma interpolação linear dos valores de pixel mínimo e máximo da matricial de origem para o intervalo de pixels mínimo e mínimo de destino.

By default the algorithm preserves the original NoData value, but there is an option to override it.

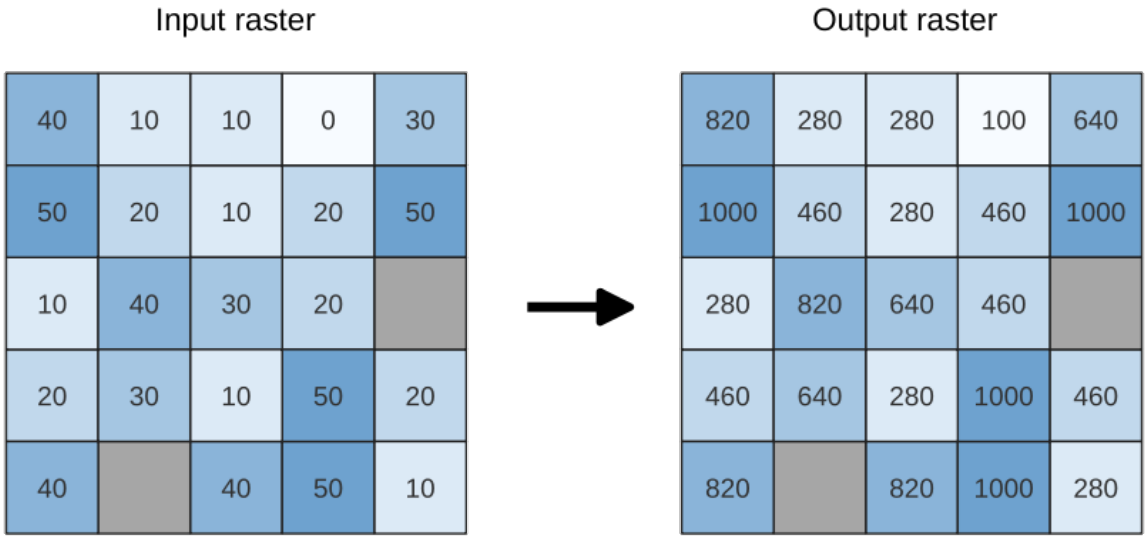


Figura24.57: Redimensionando valores de uma camada matricial de [0 - 50] para [100 - 1000]

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Entrada raster	`INPUT`	[raster]	Camada raster a ser utilizada para redimensionamento
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada de entrada	Se a matriz for multibanda, escolha uma banda.
Novo valor mínimo	`MINIMUM`	[numeric: double] Valor padrão: 0,0	Valor mínimo de pixel a ser usado na camada redimensionada
Novo valor máximo	`MAXIMUM`	[numeric: double] Valor padrão: 255,0	Valor máximo de pixel a ser usado na camada redimensionada
New NoData value Opcional	`NODATA`	[numeric: double] Valor padrão: Não definido	Value to assign to the NoData pixels. If unset, original NoData values are preserved.
Escalonado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Escalonado	`OUTPUT`	[raster]	Camada raster de saída com valores de banda redimensionados

Código Python

Algorithm ID: `native:rescaleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Rodada de raster

Rounds the cell values of a raster dataset according to the specified number of decimals.

Alternatively, a negative number of decimal places may be used to round values to powers of a base n. For example, with a Base value n of 10 and Decimal places of -1, the algorithm rounds cell values to multiples of 10, -2 rounds to multiples of 100, and so on. Arbitrary base values may be chosen, the algorithm applies the same multiplicative principle. Rounding cell values to multiples of a base n may be used to generalize raster layers.

O algoritmo preserva o tipo de dados da matriz de entrada. Portanto, matrizes de byte/inteiro só podem ser arredondadas para múltiplos de uma base n, caso contrário, um aviso é gerado e a matriz é copiada como matriz de byte/inteiro.

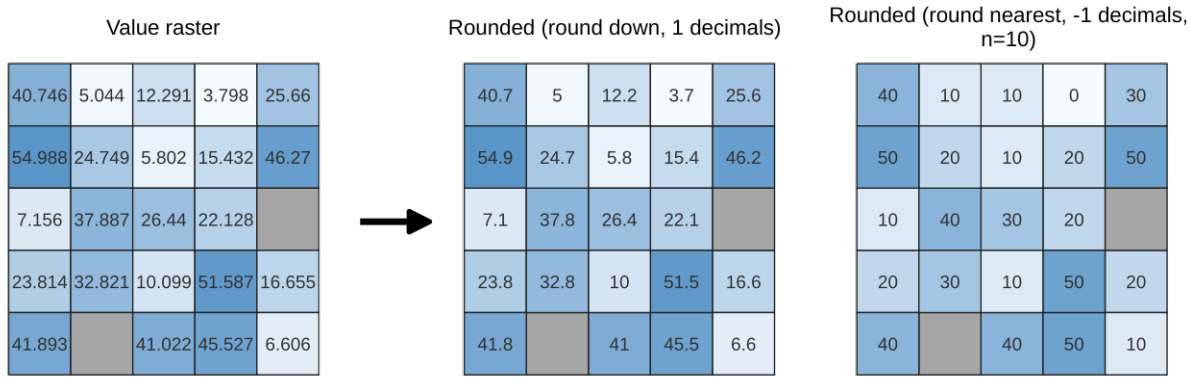


Figura24.58: Arredondamento de valores de um raster

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Entrada raster	`INPUT`	[raster]	A matriz a ser processada.
Número da banda	`BAND`	[banda raster] Padrão: 1	A banda da matriz

continua na próxima página

Tabela 24.381 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Direção de rotação	`ROUNDING_DIRECTION`	[lista] Padrão: 1	Como escolher o valor alvo arredondado. As opções são: • 0 — Arredondar para cima • 1 — Arredondamento para o mais próximo • 2 — Arredondar para baixo
Número de casas decimais	`DECIMAL_PLACES`	[numeric: integer] Padrão: 2	Número de casas decimais para arredondar. Usar valores negativos para arredondar os valores das células para um múltiplo de uma base n
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Base n para arredondamento para múltiplos de n	`BASE_N`	[numeric: integer] Padrão: 10	Quando o parâmetro `CASAS_DECIMAIS` é negativo, os valores matriciais são arredondados para múltiplos do valor base n
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	A camada matricial de saída com valores arredondados para a banda selecionada.

Código Python

Algorithm ID: `native:roundrastervalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Amostra de valores raster

Extrai valores matriciais nas localizações dos pontos. Se a camada matricial for multibanda, cada banda é amostrada.

A tabela de atributos da camada resultante terá tantas colunas novas quanto a contagem de bandas da camada raster.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Point vector layer to use for sampling
Camada Raster	`RASTERCOPY`	[raster]	Camada raster para amostragem nos locais determinados.
Output column prefix	`COLUMN_PREFIX`	[string] Padrão: 'AMOSTRA_'	Prefixo para os nomes das colunas adicionadas.
Amostrado Opcional	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output layer containing the sampled values. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Amostrado	`OUTPUT`	[vetor: ponto]	A camada de saída que contém os valores amostrados.

Código Python

Algorithm ID: `native:rastersampling`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Histograma zonal

Appends fields representing counts of each unique value from a raster layer contained within polygon features.

The output layer attribute table will have as many fields as the unique values of the raster layer that intersects the polygon(s).

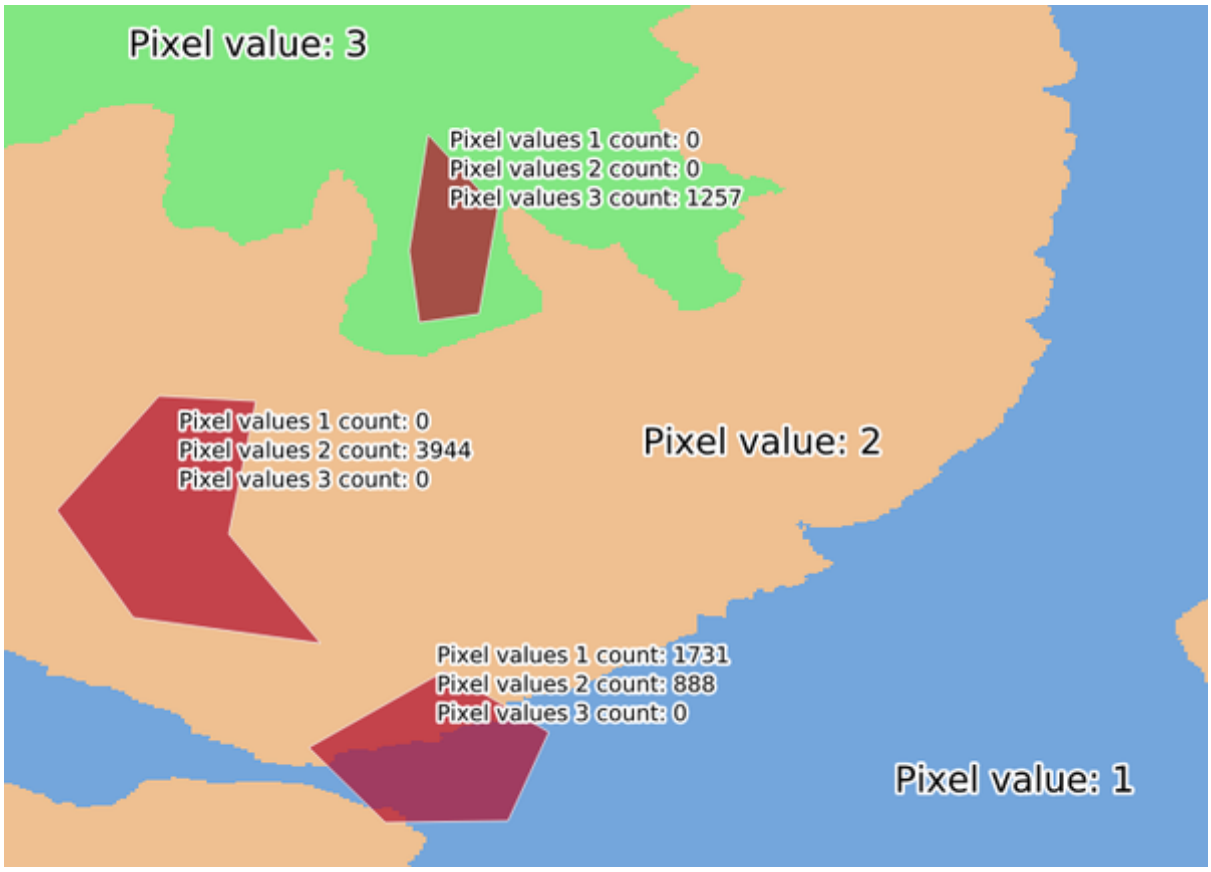


Figura24.59: Exemplo de histograma da camada raster

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada raster	`INPUT_RASTER`	[raster]	Camada raster de entrada
Número da banda	`RASTER_BAND`	[banda raster] Padrão: A primeira faixa da camada de entrada	Se a matriz for multibanda, escolha uma banda.
Camada vetorial contendo zonas	`INPUT_VECTOR`	[vetor: polígono]	Camada de polígono vetorial que define as zonas.
Output column prefix	`COLUMN_PREFIX` Opcional	[string] Default: 'HISTO_'	Prefixo para os nomes das colunas de saída.
"Áreas de saída"	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector polygon layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
"Áreas de saída"	`OUTPUT`	[vetor: polígono]	A camada de polígono vetorial de saída.

Código Python

Algorithm ID: `native:zonalhistogram`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Zonal Minimum/Maximum Point

Extracts point features corresponding to the minimum and maximum pixel values within polygon zones.

The output will contain one point feature for the minimum and one for the maximum raster value for every individual zonal feature from a polygon layer. The created point layer will be in the same spatial reference system as the selected raster layer.

> **Aviso**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	Polygon layer defining the zones.
Camada Raster	`RASTER`	[raster]	Raster layer to extract the minimum and maximum values from.
Raster Band	`BAND`	[banda raster] Padrão: A primeira faixa da camada raster	If the raster has multiple bands, select the band to process.
Zonal extrema	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specification of the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Zonal extrema	`OUTPUT`	[vetor: ponto]	Output layer containing the minimum and maximum points for each zone.

Código Python

Algorithm ID: `native:zonalminmaxpoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Estatísticas zonais

Calcula estatísticas de uma camada matricial para cada feição de uma camada vetorial de polígono sobreposta.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	Camada de polígono vetorial que contém as zonas.
Camada raster	`INPUT_RASTER`	[raster]	Camada raster de entrada
Banda Raster	`RASTER_BAND`	[banda raster] Padrão: A primeira faixa da camada de entrada	Se o raster for multibanda, escolha uma banda para as estatísticas.
Output column prefix	`COLUMN_PREFIX`	[string] Default: '_'	Prefixo para os nomes das colunas de saída.
Estatísticas para calcular	`STATISTICS`	[enumeração] [lista] Padrão: [0,1,2]	Lista de operadores estatísticos para a saída. Opções: • 0 — Contagem • 1 — Soma • 2 — Média • 3 — Mediana • 4 — St. dev. • 5 — Mínimo • 6 — Máximo • 7 — Alcance • 8 — Minoria • 9 — Maioria • 10 — Variedade • 11 — Variância
Estatísticas zonal	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector polygon layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Estatísticas zonal	`OUTPUT`	[vetor: polígono]	A camada vetorial da zona com estatísticas adicionais.

Código Python

Algorithm ID: `native:zonalstatisticsfb`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.19 Criação de Matriz

Criar camada matricial constante

Generates raster layer for given extent and cell size filled with the specified value.

Additionally an output data type can be specified. The algorithm will abort if a value has been entered that cannot be represented by the selected output raster data type.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão desejada	`EXTENT`	[extensão]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
SRC Alvo	`TARGET_CRS`	[src] Padrão: Projeto SRC	CRS for the output raster layer
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Pixel size (X=Y) in map units.

continua na próxima página

Tabela 24.389 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Valor constante	`NUMBER`	[numeric: double] Padrão: 1,0	Constant pixel value for the output raster layer.
Constante	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output raster data type	`OUTPUT_TYPE` Padrão: 5	[enumeração]	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte • 1 — Integer16 • 2 — Unsigned Integer16 • 3 — Integer32 • 4 — Unsigned Integer32 • 5 — Float32 • 6 — Float64
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Constante	`OUTPUT`	[raster]	Raster covering the desired extent with the specified pixel size and value.

Código Python

Algorithm ID: `native:createconstantrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar camada raster aleatória (distribuição binomial)

Generates a raster layer for given extent and cell size filled with binomially distributed random values.

By default, the values will be chosen given an N of 10 and a probability of 0.5. This can be overridden by using the advanced parameter for N and probability. The raster data type is set to Integer types (Integer16 by default). The binomial distribution random values are defined as positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão desejada	`EXTENT`	[extensão]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
SRC Alvo	`TARGET_CRS`	[src] Padrão: Projeto SRC	CRS for the output raster layer
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Pixel size (X=Y) in map units.
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output raster data type	`OUTPUT_TYPE` Padrão: 0	[enumeração]	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Integer16 • 1 — Unsigned Integer16 • 2 — Integer32 • 3 — Unsigned Integer32 • 4 — Float32 • 5 — Float64
N	`N`	[numeric: integer] Padrão: 10	
Probabilidade	`PROBABILITY`	[numeric: double] Padrão: 0,5	
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	Raster covering the desired extent with the cell size filled with random values

Código Python

Algorithm ID: `native:createrandombinomialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar camada raster aleatória (distribuição exponencial)

Generates a raster layer for given extent and cell size filled with exponentially distributed random values.

By default, the values will be chosen given a lambda of 1.0. This can be overridden by using the advanced parameter for lambda. The raster data type is set to Float32 by default as the exponential distribution random values are floating point numbers.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão desejada	`EXTENT`	[extensão]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
SRC Alvo	`TARGET_CRS`	[src] Padrão: Projeto SRC	CRS for the output raster layer
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Pixel size (X=Y) in map units.
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output raster data type	`OUTPUT_TYPE` Padrão: 0	[enumeração]	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Float32 • 1 — Float64

continua na próxima página

Tabela 24.396 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Lambda	`LAMBDA`	[numeric: double] Padrão: 1,0	
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	Raster covering the desired extent with the cell size filled with random values

Código Python

Algorithm ID: `native:createrandomexponentialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Create random raster layer (gamma distribution)

Generates a raster layer for given extent and cell size filled with gamma distributed random values.

By default, the values will be chosen given an alpha and beta value of 1.0. This can be overridden by using the advanced parameter for alpha and beta. The raster data type is set to Float32 by default as the gamma distribution random values are floating point numbers.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão desejada	`EXTENT`	[extensão]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
SRC Alvo	`TARGET_CRS`	[src] Padrão: Projeto SRC	CRS for the output raster layer
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Pixel size (X=Y) in map units.
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output raster data type	`OUTPUT_TYPE` Padrão: 0	[enumeração]	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Float32 • 1 — Float64
Alfa	`ALPHA`	[numeric: double] Padrão: 1,0	
Beta	`BETA`	[numeric: double] Padrão: 1,0	
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ‘’	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	Raster cobrindo a extensão desejada com o tamanho da célula preenchido com valores distribuídos aleatoriamente

Código Python

Algorithm ID: `native:createrandomgammarasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar camada raster aleatória (distribuição geométrica)

Generates a raster layer for given extent and cell size filled with geometrically distributed random values.

By default, the values will be chosen given a probability of 0.5. This can be overridden by using the advanced parameter for mean value. The raster data type is set to Integer types (Integer16 by default). The geometric distribution random values are defined as positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão desejada	`EXTENT`	[extensão]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
SRC Alvo	`TARGET_CRS`	[src] Padrão: Projeto SRC	CRS for the output raster layer
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Pixel size (X=Y) in map units.
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output raster data type	`OUTPUT_TYPE` Padrão: 0	[enumeração]	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Integer16 • 1 — Unsigned Integer16 • 2 — Integer32 • 3 — Unsigned Integer32 • 4 — Float32 • 5 — Float64
Probabilidade	`PROBABILITY`	[numeric: double] Padrão: 0,5	

continua na próxima página

Tabela 24.402 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	Raster cobrindo a extensão desejada com o tamanho da célula preenchido com valores distribuídos aleatoriamente

Código Python

Algorithm ID: `native:createrandomgeometricrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Crie uma camada raster aleatória (distribuição binomial negativa)

Generates a raster layer for given extent and cell size filled with negative binomially distributed random values.

By default, the values will be chosen given a distribution parameter k of 10.0 and a probability of 0.5. This can be overridden by using the advanced parameters for k and probability. The raster data type is set to Integer types (Integer16 by default). The negative binomial distribution random values are defined as positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão desejada	`EXTENT`	[extensão]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
SRC Alvo	`TARGET_CRS`	[src] Padrão: Projeto SRC	CRS for the output raster layer
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Pixel size (X=Y) in map units.
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output raster data type	`OUTPUT_TYPE` Padrão: 0	[enumeração]	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Integer16 • 1 — Unsigned Integer16 • 2 — Integer32 • 3 — Unsigned Integer32 • 4 — Float32 • 5 — Float64
Parâmetro de distribuição k	`K_PARAMETER`	[numeric: integer] Padrão: 10	
Probabilidade	`PROBABILITY`	[numeric: double] Padrão: 0,5	

continua na próxima página

Tabela 24.405 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	Raster cobrindo a extensão desejada com o tamanho da célula preenchido com valores distribuídos aleatoriamente

Código Python

Algorithm ID: `native:createrandomnegativebinomialrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar camada raster aleatória (distribuição normal)

Generates a raster layer for given extent and cell size filled with normally distributed random values.

By default, the values will be chosen given a mean of 0.0 and a standard deviation of 1.0. This can be overridden by using the advanced parameters for mean and standard deviation value. The raster data type is set to Float32 by default as the normal distribution random values are floating point numbers.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão desejada	`EXTENT`	[extensão]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
SRC Alvo	`TARGET_CRS`	[src] Padrão: Projeto SRC	CRS for the output raster layer
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Pixel size (X=Y) in map units.
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output raster data type	`OUTPUT_TYPE` Padrão: 0	[enumeração]	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Float32 • 1 — Float64
Média da distribuição normal	`MEAN`	[numeric: double] Padrão: 0.0	
Desvio padrão da distribuição normal	`STDDEV`	[numeric: double] Padrão: 1,0	
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ”	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	Raster cobrindo a extensão desejada com o tamanho da célula preenchido com valores distribuídos aleatoriamente

Código Python

Algorithm ID: `native:createrandomnormalrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Create random raster layer (poisson distribution)

Generates a raster layer for given extent and cell size filled with poisson distributed random values.

By default, the values will be chosen given a mean of 1.0. This can be overridden by using the advanced parameter for mean value. The raster data type is set to Integer types (Integer16 by default). The poisson distribution random values are positive integer numbers. A floating point raster will represent a cast of integer values to floating point.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão desejada	`EXTENT`	[extensão]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
SRC Alvo	`TARGET_CRS`	[src] Padrão: Projeto SRC	CRS for the output raster layer
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Pixel size (X=Y) in map units.
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output raster data type	`OUTPUT_TYPE` Padrão: 0	[enumeração]	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Integer16 • 1 — Unsigned Integer16 • 2 — Integer32 • 3 — Unsigned Integer32 • 4 — Float32 • 5 — Float64
Média	`MEAN`	[numeric: double] Padrão: 1,0	

continua na próxima página

Tabela 24.411 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	Raster cobrindo a extensão desejada com o tamanho da célula preenchido com valores distribuídos aleatoriamente

Código Python

Algorithm ID: `native:createrandompoissonrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar camada raster aleatória (distribuição uniforme)

Generates a raster layer for given extent and cell size filled with random values.

By default, the values will range between the minimum and maximum value of the specified output raster type. This can be overridden by using the advanced parameters for lower and upper bound value. If the bounds have the same value or both are zero (default) the algorithm will create random values in the full value range of the chosen raster data type. Choosing bounds outside the acceptable range of the output raster type will abort the algorithm.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão desejada	`EXTENT`	[extensão]	Specify the spatial extent of the output raster layer. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
SRC Alvo	`TARGET_CRS`	[src] Padrão: Projeto SRC	CRS for the output raster layer
Tamanho dos pixels	`PIXEL_SIZE`	[numeric: double] Padrão: 1,0	Pixel size (X=Y) in map units.
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Output raster data type	`OUTPUT_TYPE` Padrão: 5	[enumeração]	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte • 1 — Integer16 • 2 — Unsigned Integer16 • 3 — Integer32 • 4 — Unsigned Integer32 • 5 — Float32 • 6 — Float64
Lower bound for random number range	`LOWER_BOUND`	[numeric: double] Padrão: 0.0	
Upper bound for random number range	`UPPER_BOUND`	[numeric: double] Padrão: 0.0	

continua na próxima página

Tabela 24.414 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	Raster cobrindo a extensão desejada com o tamanho da célula preenchido com valores distribuídos aleatoriamente

Código Python

Algorithm ID: `native:createrandomuniformrasterlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.20 Raster terrain analysis

Aspecto

Calcula o aspecto do Modelo Digital de Terreno na entrada. A camada raster de aspecto final contém valores de 0 a 360 que expressam a direção da inclinação, começando do norte (0°) e continuando no sentido horário.

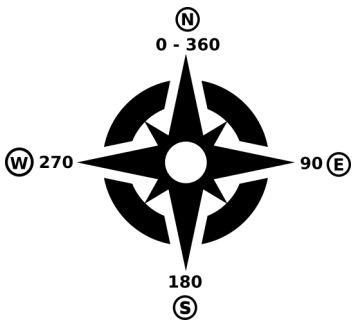


Figura24.60: Aspect values

The following picture shows the aspect layer reclassified with a color ramp:

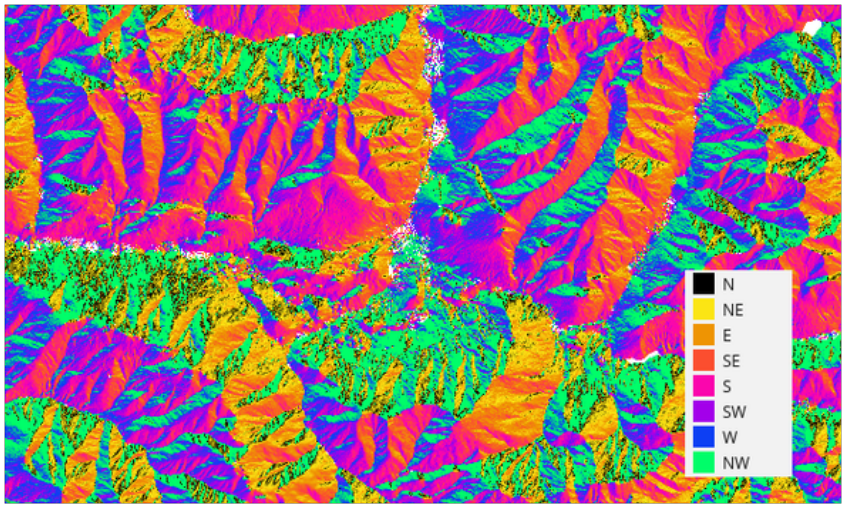


Figura24.61: Camada de aspecto reclassificado

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de elevação	`INPUT`	[raster]	Camada raster do Modelo Digital de Terreno
Fator Z	`Z_FACTOR`	[numeric: double] Padrão: 1,0	Exagero vertical. Este parâmetro é útil quando as unidades Z diferem das unidades X e Y, por exemplo, pés e metros. Você pode usar este parâmetro para ajustar isso. O padrão é 1 (sem exagero).
Aspecto	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output aspect raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Aspecto	`OUTPUT`	[raster]	The output aspect raster layer

Código Python

Algorithm ID: `native:aspect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

DTM filter (slope-based)

Can be used to filter a digital elevation model in order to classify its cells into ground and object (non-ground) cells.

The tool uses concepts as described by Vosselman (2000) and is based on the assumption that a large height difference between two nearby cells is unlikely to be caused by a steep slope in the terrain. The probability that the higher cell might be non-ground increases when the distance between the two cells decreases. Therefore the filter defines a maximum height difference (`dz_max`) between two cells as a function of the distance (`d`) between the cells (`dz_max( d ) = d`). A cell is classified as terrain if there is no cell within the kernel radius to which the height difference is larger than the allowed maximum height difference at the distance between these two cells.

The approximate terrain slope (`s`) parameter is used to modify the filter function to match the overall slope in the study area (`dz_max( d ) = d * s`). A 5 % confidence interval (`ci = 1.65 * sqrt( 2 * stddev )`) may be used to modify the filter function even further by either relaxing (`dz_max( d ) = d * s + ci`) or amplifying (`dz_max( d ) = d * s - ci`) the filter criterium.

References: Vosselman, G. (2000): Slope based filtering of laser altimetry data. IAPRS, Vol. XXXIII, Part B3, Amsterdam, The Netherlands, 935-942

Ver também

This tool is a port of the SAGA DTM Filter (slope-based)

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster do Modelo Digital de Terreno
Número da banda	`BAND`	[banda raster]	The band of the DEM to consider
Kernel radius (pixels)	`RADIUS`	[numeric: integer] Padrão: 5	The radius of the filter kernel (in pixels). Must be large enough to reach ground cells next to non-ground objects.
Terrain slope (%, pixel size/vertical units)	`TERRAIN_SLOPE`	[numeric: double] Default: 30.0	The approximate terrain slope in `%`. The terrain slope must be adjusted to account for the ratio of height units vs raster pixel dimensions. Used to relax the filter criterium in steeper terrain.

continua na próxima página

Tabela 24.418 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Filter modification	`FILTER_MODIFICATI`	[lista] Padrão: 0	Choose whether to apply the filter kernel without modification or to use a confidence interval to relax or amplify the height criterium. • 0 - None • 1 - Relax filter • 2 - Amplify
Desvio padrão	`STANDARD_DEVIATION`	[numeric: double] Padrão: 0.1	The standard deviation used to calculate a 5% confidence interval applied to the height threshold.
Output layer (ground) Opcional	`OUTPUT_GROUND`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the filtered DEM containing only cells classified as ground. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…
Output layer (non-ground objects) Opcional	`OUTPUT_NONGROUND`	[raster] Default: `[Pular saída]`	Specify the non-ground objects removed by the filter. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Output layer (ground)	`OUTPUT_GROUND`	[raster]	The filtered DEM containing only cells classified as ground.
Output layer (non-ground objects)	`OUTPUT_NONGROUND`	[raster]	The non-ground objects removed by the filter.

Código Python

Algorithm ID: `native:dtmslopebasedfilter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Fill sinks (Wang & Liu)

`NEW in 3.44`

Uses a method proposed by Wang & Liu to identify and fill surface depressions in digital elevation models.

The method was enhanced to allow the creation of hydrologically sound elevation models, i.e. not only to fill the depression(s) but also to preserve a downward slope along the flow path. If desired, this is accomplished by preserving a minimum slope gradient (and thus elevation difference) between cells.

References: Wang, L. & H. Liu (2006): An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modelling. International Journal of Geographical Information Science, Vol. 20, No. 2: 193-213.

Ver também

This tool is a port of the SAGA Fill Sinks (Wang & Liu) tool.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster do Modelo Digital de Terreno
Número da banda	`BAND`	[banda raster] Default: 1	The band of the DEM to consider
Minimum slope (degrees)	`MIN_SLOPE`	[numeric: double] Default: 0.10	Minimum slope gradient to preserve from cell to cell; with a value of zero, sinks are filled up to the spill elevation (which results in flat areas).
Output layer (filled DEM) Opcional	`OUTPUT_FILLED_DEM`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the ouput raster corresponding to the depression-free DEM. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

continua na próxima página

Tabela 24.421 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Output layer (flow directions) Opcional	`OUTPUT_FLOW_DIREC`	[raster] Default: `[Pular saída]`	Specify the output raster with computed flow directions. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…
Output layer (watershed basins) Opcional	`OUTPUT_WATERSHED_`	[raster] Default: `[Pular saída]`	Specify the output raster corresponding to the delineated watershed basins. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS`	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Output layer (filled DEM)	`OUTPUT_FILLED_DEM`	[raster]	Ouput raster corresponding to the depression-free digital elevation model.
Output layer (flow directions)	`OUTPUT_FLOW_DIREC`	[raster]	Output raster with computed flow directions; 0=N, 1=NE, 2=E, … 7=NW.
Output layer (watershed basins)	`OUTPUT_WATERSHED_`	[raster]	Output raster corresponding to the delineated watershed basins.

Código Python

Algorithm ID: `native:fillsinkswangliu`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Sombreamento

Calculates the hillshade raster layer given an input Digital Terrain Model.

O sombreamento da camada é calculado de acordo com a posição do sol: você tem a opção de alterar tanto o ângulo horizontal (azimute) quanto o ângulo vertical (elevação do sol) do sol.

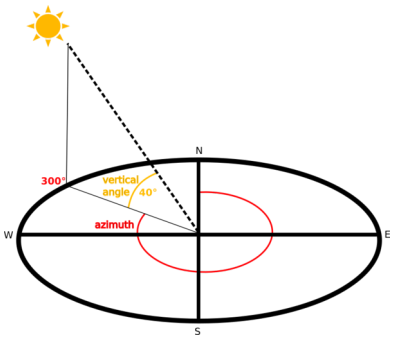


Figura24.62: Azimute e ângulo vertical

The hillshade layer contains values from 0 (complete shadow) to 255 (complete sun). Hillshade is used usually to better understand the relief of the area.

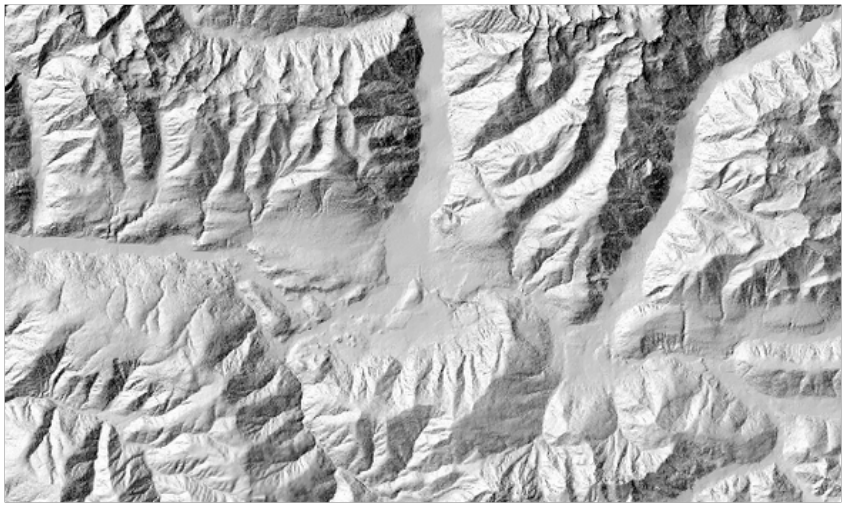


Figura24.63: Hillshade layer with azimuth 300 and vertical angle 45

Particularly interesting is to give the hillshade layer a transparency value and overlap it with the elevation raster:

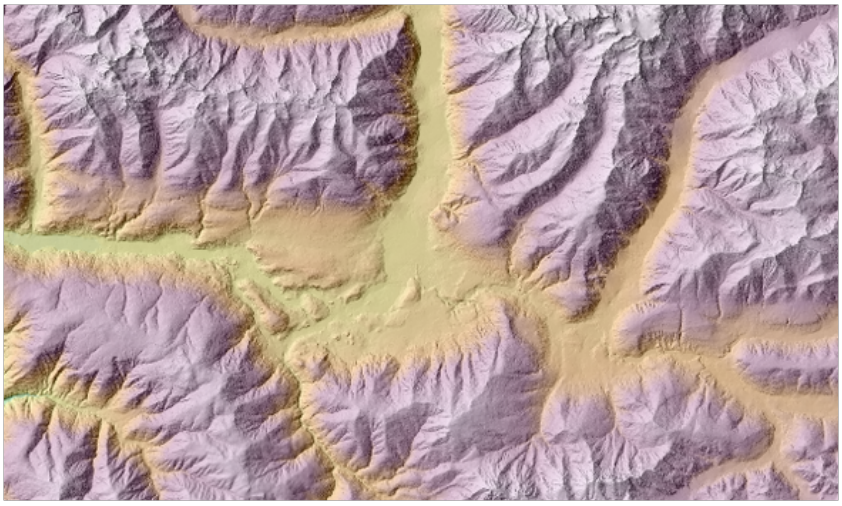


Figura24.64: Overlapping the hillshade with the elevation layer

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de elevação	`INPUT`	[raster]	Camada raster do Modelo Digital de Terreno
Fator Z	`Z_FACTOR`	[numeric: double] Padrão: 1,0	Vertical exaggeration. This parameter is useful when the Z units differ from the X and Y units, for example feet and meters. You can use this parameter to adjust for this. Increasing the value of this parameter will exaggerate the final result (making it look more “hilly”). The default is 1 (no exaggeration).
Azimute (ângulo horizontal)	`AZIMUTH`	[numeric: double] Padrão: 300.0	Defina o ângulo horizontal (em graus) do sol (sentido horário). Faixa: 0 a 360. 0 é o norte.
Ângulo vertical	`V_ANGLE`	[numeric: double] Padrão: 40,0	Defina o ângulo vertical (em graus) do sol, que é a altura do sol. Os valores podem ir de 0 (elevação mínima) a 90 (elevação máxima).
Sombreamento	`OUTPUT`	[raster] Default: `Save to temporary file`	Specify the output hillshade raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Sombreamento	`OUTPUT`	[raster]	The output hillshade raster layer

Código Python

Algorithm ID: `native:hillshade`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Curvas hipsométricas

Calculates hypsometric curves for an input Digital Elevation Model. Curves are produced as CSV files in an output folder specified by the user.

Uma curva hipsométrica é um histograma cumulativo de valores de elevação em uma área geográfica.

Você pode usar curvas hipsométricas para detectar diferenças na paisagem devido à geomorfologia do território.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
DEM to analyze	`INPUT_DEM`	[raster]	Camada raster do Modelo Digital de Terreno a ser usada para calcular altitudes
Boundary layer	`BOUNDARY_LAYER`	[vetor: polígono]	Camada vetorial de polígono com limites de áreas usadas para calcular curvas hipsométricas
Step	`STEP`	[numeric: double] Padrão: 100.0	Distância vertical entre curvas
Usar % da área em vez do valor absoluto	`USE_PERCENTAGE`	[boleano] Padrão: Falso	Escreva a porcentagem da área no campo "Área" do arquivo CSV em vez da área absoluta
Curvas Hipsométricas	`OUTPUT_DIRECTORY`	[pasta]	Specify the output folder for the hypsometric curves. *One of*: • Salvar como Diretório Temporário • Salvar no Diretório

Saídas

Etiqueta	Nome	Tipo	Descrição
Curvas Hipsométricas	`OUTPUT_DIRECTORY`	[pasta]	Directory containing the files with the hypsometric curves. For each feature from the input vector layer, a CSV file with area and altitude values will be created. The file names start with `histogram_`, followed by layer name and feature ID.

	A	B
1	Area	Elevation
2	177475194.383	307
3	233206029.24	407
4	295553735.793	507
5	394718815.615	607
6	501801102.615	707
7	624399019.792	807
8	828877274.39	907
9	1042693465.68	1007
10	1277373021.81	1107
11	1556443975.41	1207
12	1888617494.27	1307
13	2248520437.31	1407
14	2627916813.17	1507
15	3010880212.04	1607
16	3411087555.34	1707

Código Python

Algorithm ID: `qgis:hypsometriccurves`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Relevo

Creates a shaded relief layer from digital elevation data. You can specify the relief color manually, or you can let the algorithm choose automatically all the relief classes.

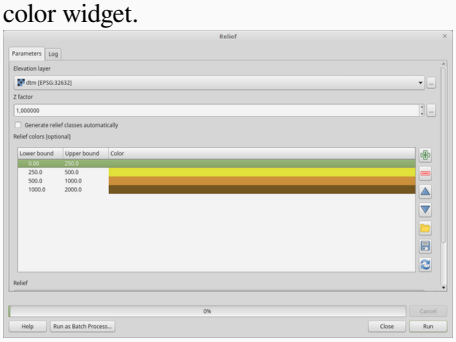
Figura24.65: Camada de relevo

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de elevação	`INPUT`	[raster]	Camada raster do Modelo Digital de Terreno
Fator Z	`Z_FACTOR`	[numeric: double] Padrão: 1,0	Vertical exaggeration. This parameter is useful when the Z units differ from the X and Y units, for example feet and meters. You can use this parameter to adjust for this. Increasing the value of this parameter will exaggerate the final result (making it look more “hilly”). The default is 1 (no exaggeration).
Gerar classes de relevo automaticamente	`AUTO_COLORS`	[boleano] Padrão: Falso	Se você marcar esta opção o algoritmo irá criar todas as classes de cores de relevo automaticamente

continua na próxima página

Tabela 24.428 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Cores de relevo Opcional	`COLORS`	[table widget]	Use the table widget if you want to choose the relief colors manually. You can add as many color classes as you want: for each class you can choose the lower and upper bound and finally by clicking on the color row you can choose the color thanks to the color widget.  Figura24.66: Ajuste manual das classes de cores em relevo The buttons in the right side panel give you the chance to: add or remove color classes, change the order of the color classes already defined, open an existing file with color classes and save the current classes as file.
Relevo	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output relief raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…
Distribuição de frequência Opcional	`FREQUENCY_DISTRIB`	[vector: table] Default: `[Pular saída]`	Specify the CSV table for the output frequency distribution. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Relevo	`OUTPUT`	[raster]	A camada raster de relevo de saída
Distribuição de frequência	`OUTPUT`	[vector: table]	A distribuição da freqüência de saída

Código Python

Algorithm ID: `qgis:relief`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Ruggedness index

Calculates the quantitative measurement of terrain heterogeneity described by Riley et al. (1999). It is calculated for every location, by summarizing the change in elevation within the 3x3 pixel grid.

Cada célula contém a somatória das diferenças absolutas de elevação entre uma célula central e as 8 que a envolvem.

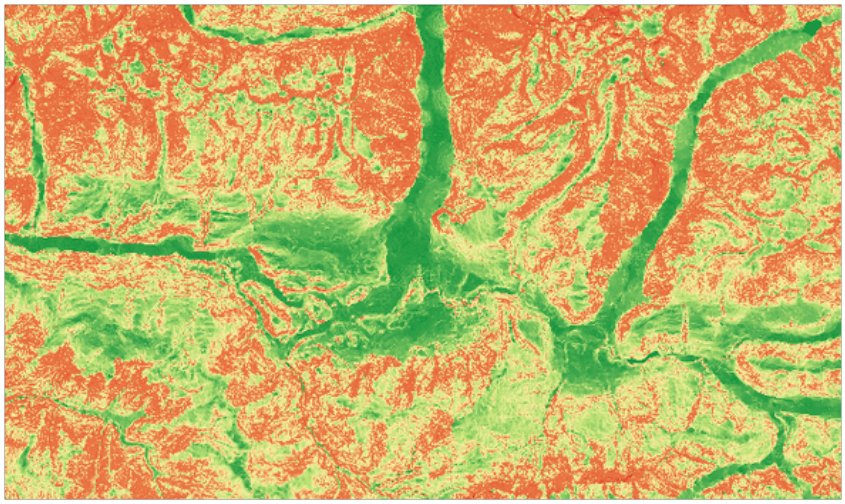


Figura24.67: Ruggedness layer from low (red) to high values (green)

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de elevação	`INPUT`	[raster]	Camada raster do Modelo Digital de Terreno
Fator Z	`Z_FACTOR`	[numeric: double] Padrão: 1,0	Vertical exaggeration. This parameter is useful when the Z units differ from the X and Y units, for example feet and meters. You can use this parameter to adjust for this. Increasing the value of this parameter will exaggerate the final result (making it look more rugged). The default is 1 (no exaggeration).

continua na próxima página

Tabela 24.430 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Ruggedness	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output ruggedness raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Ruggedness	`OUTPUT`	[raster]	The output ruggedness raster layer

Código Python

Algorithm ID: `native:ruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Inclinação

Calculates the slope from an input raster layer. The slope is the angle of inclination of the terrain and is expressed in **degrees**.

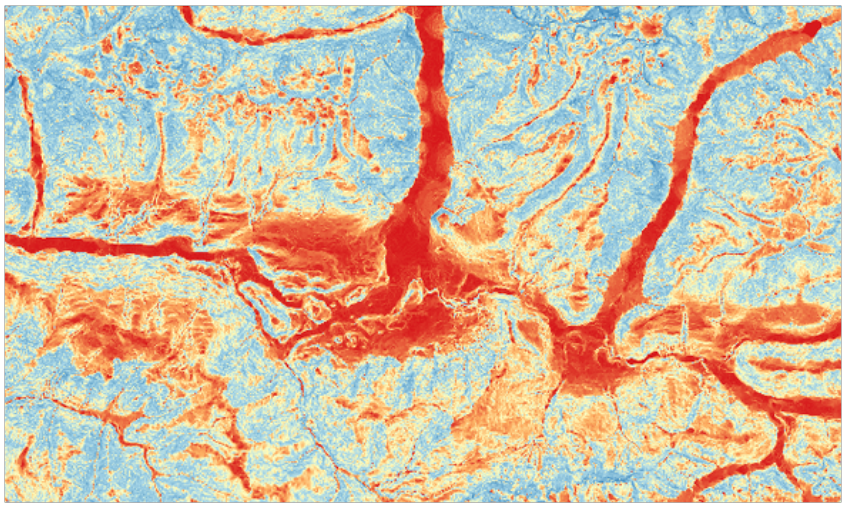


Figura24.68: Áreas planas em vermelho, áreas íngremes em azul

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de elevação	`INPUT`	[raster]	Camada raster do Modelo Digital de Terreno
Fator Z	`Z_FACTOR`	[numeric: double] Padrão: 1,0	Exagero vertical. Este parâmetro é útil quando as unidades Z diferem das unidades X e Y, por exemplo, pés e metros. Você pode usar este parâmetro para ajustar isso. Aumentar o valor deste parâmetro irá exagerar o resultado final (tornando-o mais acentuado). O padrão é 1 (sem exagero).
Declive	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output slope raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Declive	`OUTPUT`	[raster]	The output slope raster layer

Código Python

Algorithm ID: `native:slope`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.21 Ferramentas de raster

Align raster

Aligns raster by resampling it to the same cell size and reprojecting to the same CRS as a reference raster.

Aviso

This algorithm is ONLY available in the *Model Designer* context. For other contexts, use instead *Align rasters*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Input raster layer to align
Método de reamostragem	`RESAMPLING_METHOD`	[enumeração] Padrão: 0	Method to use for input layer resampling. `Available options` are: • 0 — Nearest Neighbour • 1 — Bilinear (2x2 kernel) • 2 — Cubic (4x4 kernel) • 3 — Cubic B-Spline (4x4 kernel) • 4 — Lanczos (6x6 kernel) • 5 — Média • 6 — Modo • 7 — Máximo • 8 — Mínimo • 9 — Mediana • 10 — First quartile (Q1) • 11 — Third quartile (Q3)
Rescale values according to the cell size	`RESCALE`	[boleano] Default: No	
Camada de referência	`REFERENCE_LAYER`	[raster]	A raster layer that will be used to fetch extent, cell size and CRS that will be applied to input layers.
Override reference CRS Opcional	`CRS`	[src]	CRS to be used instead of the reference layer's
Override reference cell size X Opcional	`CELL_SIZE_X`	[numeric: double]	Cell size in X direction to be used instead of the reference layer's
Override reference cell size Y Opcional	`CELL_SIZE_Y`	[numeric: double]	Cell size in Y direction to be used instead of the reference layer's
Override reference grid offset X Opcional	`GRID_OFFSET_X`	[numeric: double]	Offset in X direction to apply to cells grid
Override reference grid offset Y Opcional	`GRID_OFFSET_Y`	[numeric: double]	Offset in Y direction to apply to cells grid

continua na próxima página

Tabela 24.434 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Clip to extent Opcional	`EXTENT`	[extensão]	Especificar a extensão da camada matricial de saída. Ela será estendida internamente para um múltiplo do tamanho do bloco. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Aligned raster	`OUTPUT`	[raster]	output raster with pixels resampled

Saídas

Etiqueta	Nome	Tipo	Descrição
Aligned raster	`OUTPUT`	[raster]	output raster with pixels resampled

Código Python

Algorithm ID: `native:alignsingleraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Align rasters

Aligns rasters by resampling them to the same cell size and reprojecting to the same CRS as a reference raster.

Aviso

This algorithm is NOT available in the *Model Designer* context. Use instead *Align raster*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	List of input raster layers with resampling options associated (filled as a `QgsProcessingParameterAlignRasterLayers` item — done in GUI by pressing *Configure Raster…* button for each selected layer): **Input layer [string] (`inputFile`)** Full path of the input layer to align **Output file [string] (`outputFile`)** Full path of the corresponding aligned output layer **Resampling method [enumeration] (`resampleMethod`)** Method to use for input layer resampling. `Available options` are: • 0 — Nearest Neighbour • 1 — Bilinear (2x2 kernel) • 2 — Cubic (4x4 kernel) • 3 — Cubic B-Spline (4x4 kernel) • 4 — Lanczos (6x6 kernel) • 5 — Média • 6 — Modo • 7 — Máximo • 8 — Mínimo • 9 — Mediana • 10 — First quartile (Q1) • 11 — Third quartile (Q3) **Rescale values according to the cell size** [boolean] (`rescale`)
Camada de referência	`REFERENCE_LAYER`	[raster]	A raster layer that will be used to fetch extent, cell size and CRS that will be applied to input layers.
Override reference CRS Opcional	`CRS`	[src]	CRS to be used instead of the reference layer's
Override reference cell size X Opcional	`CELL_SIZE_X`	[numeric: double]	Cell size in X direction to be used instead of the reference layer's
Override reference cell size Y Opcional	`CELL_SIZE_Y`	[numeric: double]	Cell size in Y direction to be used instead of the reference layer's
Override reference grid offset X Opcional	`GRID_OFFSET_X`	[numeric: double]	Offset in X direction to apply to cells grid
Override reference grid offset Y Opcional	`GRID_OFFSET_Y`	[numeric: double]	Offset in Y direction to apply to cells grid

continua na próxima página

Tabela 24.436 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Clip to extent Opcional	`EXTENT`	[extensão]	Especificar a extensão da camada matricial de saída. Ela será estendida internamente para um múltiplo do tamanho do bloco. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`

Saídas

Etiqueta	Nome	Tipo	Descrição
Aligned rasters	`OUTPUT_LAYERS`	[raster] [lista]	output rasters with pixels resampled

Código Python

Algorithm ID: `native:alignrasters`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Converter mapa para raster

Cria uma imagem raster do conteúdo da tela do mapa.

A *tema do mapa* pode ser selecionado para renderizar um conjunto predeterminado de camadas com um estilo definido para cada camada.

Alternativamente, uma única camada pode ser selecionada se nenhum tema de mapa for definido.

Se nem o tema do mapa nem a camada estiverem definidos, o conteúdo do mapa atual será renderizado. A extensão mínima inserida será estendida internamente para ser um múltiplo do tamanho do bloco.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Extensão mínima para renderizar (xmin, xmax, ymin, ymax)	`EXTENT`	[extensão]	Especificar a extensão da camada matricial de saída. Ela será estendida internamente para um múltiplo do tamanho do bloco. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Tamanho do bloco	`TILE_SIZE`	[numeric: integer] Padrão: 1024	Tamanho do bloco da camada matricial de saída. Valor mínimo: 64.
Unidades de mapa por pixel*	`MAP_UNITS_PER_`	[numeric: double] Padrão: 100.0	Tamanho do pixel (em unidades de mapa). Valor mínimo: 0,0
Tornar o fundo transparente*	`MAKE_BACKGROUN`	[boleano] Padrão: Falso	Permite exportar o mapa com um fundo transparente. Produz uma imagem RGBA (ao invés de RGB) se definida como `Verdade`.
Tema de mapa para renderizar Opcional	`MAP_THEME`	[enumeração]	Utilize um *tema de mapa* existente para a renderização.
Camada única para renderizar Opcional	`LAYER`	[enumeração]	Escolher uma única camada para a renderização
Camada de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[raster]	Camada raster de saída

Código Python

Algorithm ID: `native:rasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Preencher células sem dados

Redefine os valores 'sem dado' no raster de entrada para um valor escolhido, resultando em um conjunto de dados raster pixels 'sem dado'.

O algoritmo respeita o tipo de dados raster de entrada, por exemplo, um valor de preenchimento de ponto flutuante será truncado quando aplicado a um raster inteiro.

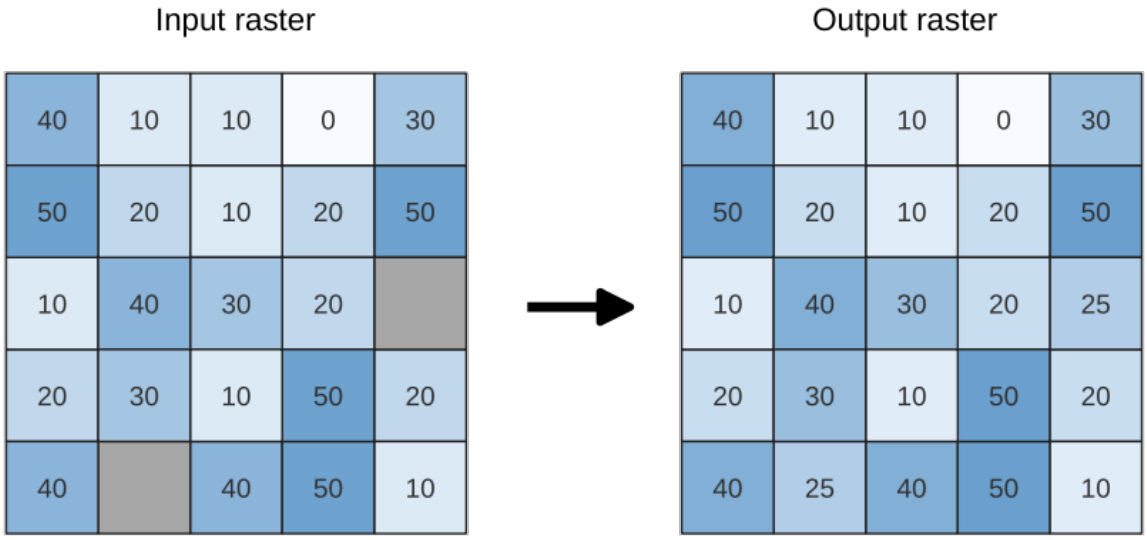


Figura24.69: Preenchimento de valores 'sem valores' (em cinza) de um raster

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Entrada raster	`INPUT`	[raster]	A matriz a ser processada.
Número da banda	`BAND`	[banda raster] Padrão: 1	A banda da matriz
Preencher valor	`FILL_VALUE`	[numeric: double] Padrão: 1,0	Defina o valor a ser usado para os pixels sem dados

continua na próxima página

Tabela 24.440 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Creation options Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `CREATE_OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Matriz de saída	`OUTPUT`	[raster]	A camada raster de saída com células de dados preenchidas.

Código Python

Algorithm ID: `native:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Gerar mosaico XYZ (Diretório)

Generates raster "XYZ" tiles using the current QGIS project as individual images to a directory structure.

Optionally, a Leaflet HTML output file using the generated tiles as a map layer could be created.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Extensão (xmin, xmax, ymin, ymax)	`EXTENT`	[extensão]	Especificar a extensão dos blocos. Ele será estendido internamente para um múltiplo do tamanho do bloco. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Zoom mínimo	`ZOOM_MIN`	[numeric: integer] Padrão: 12	Mínimo 0, máximo 25.
Zoom máximo	`ZOOM_MAX`	[numeric: integer] Padrão: 12	Mínimo 0, máximo 25.
DPI	`DPI`	[numeric: integer] Padrão: 96	Mínimo 48, máximo 600.
Cor de fundo Opcional	`BACKGROUND_COLOR`	[cor] Padrão: QCor(0, 0, 0, 0)	Escolha a cor de fundo para os blocos
Enable antialiasing	`ANTIALIAS`	[boleano] Padrão: Verdadeiro	Determines if antialiasing should be enabled
Formato de bloco	`TILE_FORMAT`	[enumeração] Padrão: 0	Um de: • 0 — PNG • 1 — JPG
Qualidade (somente JPG) Opcional	`QUALITY`	[numeric: integer] Padrão: 75	Mínimo 1, máximo 100.
Metatile size Opcional	`METATILESIZE`	[numeric: integer] Padrão: 4	Specify a custom metatile size when generating XYZ tiles. Larger values may speed up the rendering of tiles and provide better labelling (fewer gaps without labels) at the expense of using more memory. Minimum 1, maximum 20.
Largura do bloco Opcional	`TILE_WIDTH`	[numeric: integer] Padrão: 256	Mínimo 1, máximo 4096.
Altura de bloco Opcional	`TILE_HEIGHT`	[numeric: integer] Padrão: 256	Mínimo 1, máximo 4096.

continua na próxima página

Tabela 24.443 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Utilizar mosaíco invertido eixo Y (convenções TMS)	`TMS_CONVENTION`	[boleano] Padrão: Falso	
Diretório de saída	`OUTPUT_DIRECTORY`	[pasta] Padrão: `[Salvar em pasta temporária]`	Specification of the output directory (for the tiles). *One of*: • Ignorar Saída • Salvar como Diretório Temporário • Salvar no Diretório
Saída html (Folheto) Opcional	`OUTPUT_HTML`	[html] Padrão: `[Salvar em arquivo temporário]`	Specification of the output HTML file. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Leaflet HTML output title Opcional	`HTML_TITLE`	[string] Padrão: Não definido	HTML <title>-tag used for the Leaflet HTML output file.
Leaflet HTML output attribution Opcional	`HTML_ATTRIBUTI`	[string] Padrão: Não definido	Custom map attribution used for the Leaflet HTML output file. HTML links are possible.
Include OpenStreetMap basemap in Leaflet HTML output	`HTML_OSM`	[boleano] Padrão: Falso	An OpenStreetMap basemap layer (source: https://tile.openstreetmap.org) is included in the Leaflet HTML output file. Proper map attribution is added automatically.

Saídas

Etiqueta	Nome	Tipo	Descrição
Diretório de saída	`OUTPUT_DIRECTORY`	[pasta]	Diretório de saída (para os blocos)
Saída html (Folheto)	`OUTPUT_HTML`	[html]	O arquivo HTML de saída (Cartilha)

Código Python

Algorithm ID: `native:tilesxyzdirectory`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Generate XYZ tiles (MBTiles)

Generates raster "XYZ" tiles using the current QGIS project as a single file in the "MBTiles" format.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Extensão (xmin, xmax, ymin, ymax)	`EXTENT`	[extensão]	Especificar a extensão dos blocos. Ele será estendido internamente para um múltiplo do tamanho do bloco. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Zoom mínimo	`ZOOM_MIN`	[numeric: integer] Padrão: 12	Mínimo 0, máximo 25.
Zoom máximo	`ZOOM_MAX`	[numeric: integer] Padrão: 12	Mínimo 0, máximo 25.
DPI	`DPI`	[numeric: integer] Padrão: 96	Mínimo 48, máximo 600.
Cor de fundo Opcional	`BACKGROUND_COLOR`	[cor] Padrão: QCor(0, 0, 0, 0)	Escolha a cor de fundo para os blocos
Enable antialiasing	`ANTIALIAS`	[boleano] Padrão: Verdadeiro	Determines if antialiasing should be enabled
Formato de bloco	`TILE_FORMAT`	[enumeração] Padrão: 0	Um de: • 0 — PNG • 1 — JPG

continua na próxima página

Tabela 24.446 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Qualidade (somente JPG) Opcional	`QUALITY`	[numeric: integer] Padrão: 75	Mínimo 1, máximo 100.
Metatile size Opcional	`METATILESIZE`	[numeric: integer] Padrão: 4	Specify a custom metatile size when generating XYZ tiles. Larger values may speed up the rendering of tiles and provide better labelling (fewer gaps without labels) at the expense of using more memory. Minimum 1, maximum 20.
Output file (for MBTiles)	`OUTPUT_FILE`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Output file (for MBTiles)	`OUTPUT_FILE`	[arquivo]	O arquivo de saída.

Código Python

Algorithm ID: `native:tilesxyzmbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.22 Análise vetorial

Estatísticas básicas para campos

Generates basic statistics for a field of the attribute table of a vector layer. Numeric, date, time and string fields are supported. The statistics returned will depend on the field type.

Statistics can be generated as a table or an HTML file and are available in the *Processing* ► *Results viewer*.

Default menu: *Vector* ► *Analysis Tools*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Input vector	`INPUT_LAYER`	[vetor: qualquer]	Camada vetorial para calcular as estatísticas
Campo para calcular as estatísticas	`FIELD_NAME`	[campo da tabela: qualquer]	Qualquer campo de tabela suportado para calcular as estatísticas
Estatísticas Opcional	`OUTPUT`	[vector: table] Padrão: `[Criar camada temporária]`	Specify the output table for the generated statistics. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Statistics report Opcional	`OUTPUT_HTML_FILE`	[html] Padrão: `[Salvar em arquivo temporário]`	Specification of the file for the calculated statistics. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Estatísticas	`OUTPUT`	[vector: table]	Table containing the calculated statistics
Statistics report	`OUTPUT_HTML_FILE`	[html]	HTML file with the calculated statistics
Contagem	`COUNT`	[numeric: integer]	
Número de valores únicos	`UNIQUE`	[numeric: integer]	
Number of empty (null) values	`EMPTY`	[numeric: integer]	
Número de valores não vazios	`FILLED`	[numeric: integer]	
Valor mínimo	`MIN`	[o mesmo que entrada]	
Valor máximo	`MAX`	[o mesmo que entrada]	
Comprimento mínimo	`MIN_LENGTH`	[numeric: integer]	
Comprimento máximo	`MAX_LENGTH`	[numeric: integer]	
Comprimento médio	`MEAN_LENGTH`	[numeric: double]	
Coeficiente de variação	`CV`	[numeric: double]	
Soma	`SUM`	[numeric: double]	
Valor médio	`MEAN`	[numeric: double]	
Desvio padrão	`STD_DEV`	[numeric: double]	
Intervalo	`RANGE`	[numeric: double]	
Mediana	`MEDIAN`	[numeric: double]	
Minoria (valor de ocorrência mais rara)	`MINORITY`	[o mesmo que entrada]	
Maioria (valor que ocorre com mais frequência)	`MAJORITY`	[o mesmo que entrada]	

continua na próxima página

Tabela 24.449 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Primeiro quartil	`FIRSTQUARTILE`	[numeric: double]	
Terceiro quartil	`THIRDQUARTILE`	[numeric: double]	
Interquartile Range (IQR)	`IQR`	[numeric: double]	

Código Python

Algorithm ID: `native:basicstatisticsforfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Climb along line

Calculates the total climb and descent along line geometries. The input layer must have Z values present. If Z values are not available, the *Drape (configure o valor Z do raster)* algorithm may be used to add Z values from a DEM layer.

The output layer is a copy of the input layer with additional fields that contain the total climb (`climb`), total descent (`descent`), the minimum elevation (`minelev`) and the maximum elevation (`maxelev`) for each line geometry. If the input layer contains fields with the same names as these added fields, they will be renamed (field names will be altered to “name_2”, “name_3”, etc, finding the first non-duplicate name).

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de linha	`INPUT`	[vetor: linha]	Line layer to calculate the climb for. Must have Z values
Climb layer	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specification of the output (line) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Climb layer	`OUTPUT`	[vetor: linha]	Line layer containing new attributes with the results from climb calculations.
Total climb	`TOTALCLIMB`	[numeric: double]	The sum of the climb for all the line geometries in the input layer
Total descent	`TOTALDESCENT`	[numeric: double]	The sum of the descent for all the line geometries in the input layer
Elevação mínima	`MINELEVATION`	[numeric: double]	A elevação mínima para as geometrias na camada
Elevação máxima	`MAXELEVATION`	[numeric: double]	A elevação máxima para as geometrias na camada

Código Python

Algorithm ID: `native:climbalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Contar pontos em polígono

Takes a point and a polygon layer and counts the number of points from the point layer in each of the polygons of the polygon layer.

A new polygon layer is generated, with the exact same content as the input polygon layer, but containing an additional field with the points count corresponding to each polygon.

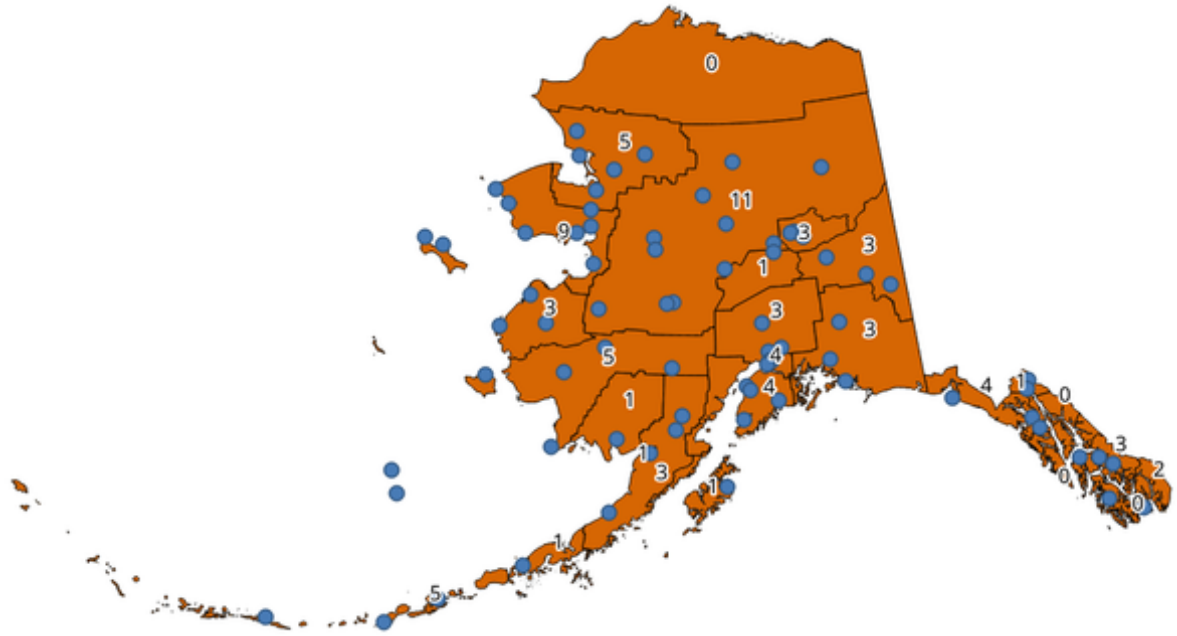


Figura24.70: Os rótulos nos polígonos mostram a contagem de pontos

An optional weight field can be used to assign weights to each point. Alternatively, a unique class field can be specified. If both options are used, the weight field will take precedence and the unique class field will be ignored.

Allows features in-place modification 1 das características dos polígonos.

`Menu padrão`: *Vetor* ► *Analisar*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Polígonos	`POLYGONS`	[vetor: polígono]	Polygon layer whose features are associated with the count of points they contain
Pontos	`POINTS`	[vetor: ponto]	Point layer with features to count
Campo de peso Opcional	`WEIGHT`	[campo da tabela: qualquer]	A field from the point layer. The count generated will be the sum of the weight field of the points contained by the polygon. If the weight field is not numeric, the count will be `0`.
Campo de classe Opcional	`CLASSFIELD`	[campo da tabela: qualquer]	Os pontos são classificados com base no atributo selecionado e se vários pontos com o mesmo valor de atributo estiverem dentro do polígono, apenas um deles será contado. A contagem final dos pontos em um polígono é, portanto, a contagem das diferentes classes que nele se encontram.
Count field name	`FIELD`	[string] Padrão: 'NUMPONTOS'	O nome do campo para armazenar a contagem de pontos

continua na próxima página

Tabela 24.452 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Contagem	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specification of the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Contagem	`OUTPUT`	[vetor: polígono]	Resulting layer with the attribute table containing the new column with the points count

Código Python

Algorithm ID: `native:countpointsinpolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

DBSCAN clustering

Clusters point features based on a 2D implementation of Density-based spatial clustering of applications with noise (DBSCAN) algorithm.

O algoritmo requer dois parâmetros, um tamanho mínimo de agrupamento e a distância máxima permitida entre os pontos agrupados.

Ver também

ST-DBSCAN clustering, *K-means clustering*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Camada a ser analisada
Tamanho mínimo do conjunto	`MIN_SIZE`	[numeric: integer] Padrão: 5	Número mínimo de feições para gerar um conjunto
Maximum distance between clustered points	`EPS`	[numeric: double] Padrão: 1,0	Distance beyond which two features can not belong to the same cluster (eps)
Clusters	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the vector layer for the result of the clustering. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Treat border points as noise (DBSCAN*)	`DBSCAN*`	[boleano] Padrão: Falso	If checked, points on the border of a cluster are themselves treated as unclustered points, and only points in the interior of a cluster are tagged as clustered.
Cluster field name	`FIELD_NAME`	[string] Default: 'CLUSTER_ID'	Name of the field where the associated cluster number shall be stored
Cluster size field name	`SIZE_FIELD_NAM`	[string] Default: 'CLUSTER_SIZE'	Nome do campo com a contagem de feições no mesmo grupo

Saídas

Etiqueta	Nome	Tipo	Descrição
Clusters	`OUTPUT`	[vetor: ponto]	Vector layer containing the original features with a field setting the cluster they belong to
Número de agrupamentos	`NUM_CLUSTERS`	[numeric: integer]	O número de agrupamentos descobertos

Código Python

Algorithm ID: `native:dbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Matriz de distância

Calculates for point features distances to their nearest features in the same layer or in another layer.

Default menu: *Vector* ► *Analysis Tools*

Ver também

Unir atributos por mais próximo

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada de ponto	`INPUT`	[vetor: ponto]	Point layer for which the distance matrix is calculated (**from** points)
Input unique ID field	`INPUT_FIELD`	[campo da tabela: qualquer]	Field to use to uniquely identify features of the input layer. Used in the output attribute table.
Target point layer	`TARGET`	[vetor: ponto]	Point layer containing the nearest point(s) to search (**to** points)
Target unique ID field	`TARGET_FIELD`	[campo da tabela: qualquer]	Field to use to uniquely identify features of the target layer. Used in the output attribute table.
Output matrix type	`MATRIX_TYPE`	[enumeração] Padrão: 0	Diferentes tipos de cálculo estão disponíveis: • 0 — Linear (N * *k* x 3) distance matrix: for each input point, reports the distance to each of the *k* nearest target points. The output matrix consists of up to *k* rows per input point, and each row has three columns: *InputID*, *TargetID* and *Distance*. • 1 — Matriz de distância padrão (N x T) • 2 — Summary distance matrix (mean, std. dev., min, max): for each input point, reports statistics on the distances to its target points.
Use apenas os pontos alvo (k) mais próximos	`NEAREST_POINTS`	[numeric: integer] Padrão: 0	You can choose to calculate the distance to all the points in the target layer (*0*) or limit to a number (*k*) of closest features.

continua na próxima página

Tabela 24.457 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Distance matrix	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specification of the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Distance matrix	`OUTPUT`	[vetor: ponto]	Point (or MultiPoint for the "Linear (N * *k* x 3)" case) vector layer containing the distance calculation for each input feature. Its features and attribute table depend on the selected output matrix type.

Código Python

Algorithm ID: `qgis:distancematrix`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Distance to nearest hub (line to hub)

Creates lines that join each feature of an input vector to the nearest feature in a destination layer. Distances are calculated based on the *center* of each feature.

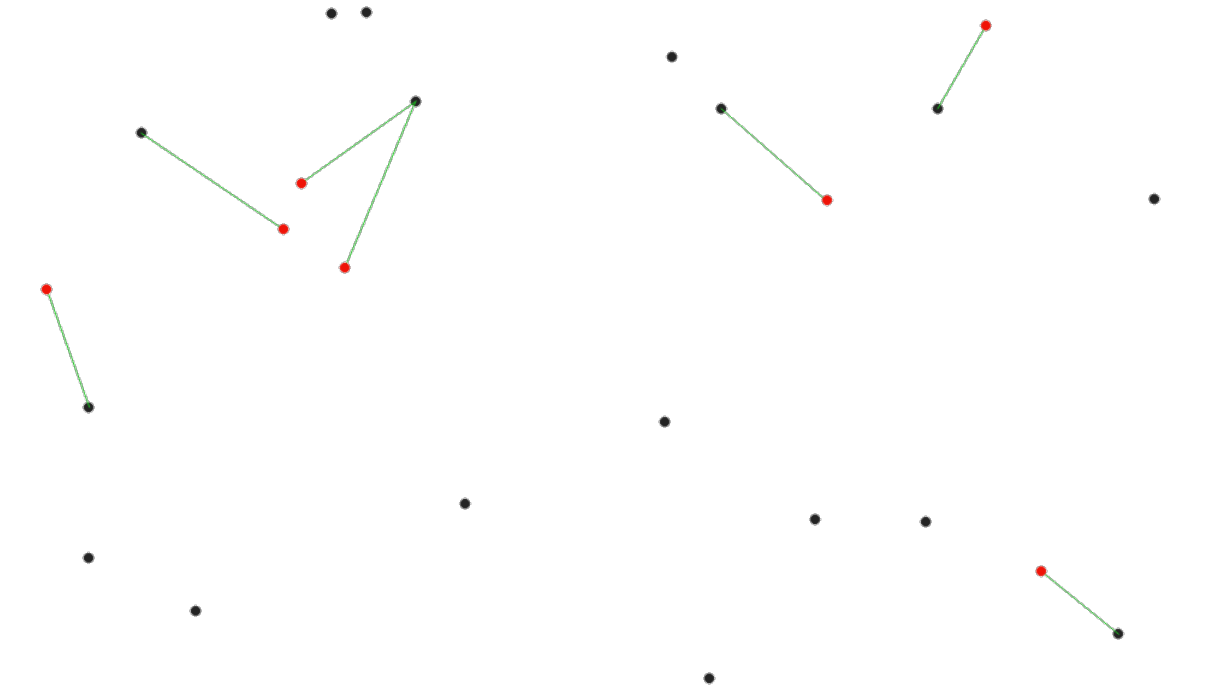


Figura24.71: Display the nearest hub for the red input features

Ver também

Distance to nearest hub (points), *Unir atributos por mais próximo*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Source points layer	`INPUT`	[vector: geometry]	Vector layer for which the nearest feature is searched
Destination hubs layer	`HUBS`	[vector: geometry]	Camada vetorial contendo as feições a serem pesquisadas
Hub layer name attribute	`FIELD`	[campo da tabela: qualquer]	Campo a ser usado para identificar exclusivamente as feições da camada de destino. Usado na tabela de atributos de saída
Unidade de medida	`UNIT`	[enumeração] Padrão: 0	Unidades nas quais relatar a distância até a feição mais próxima: • 0 — Metros • 1 — Pés • 2 — Milhas • 3 — Quilômetros • 4 — Unidades de camada

continua na próxima página

Tabela 24.459 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Hub distance	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line vector layer connecting the matching points. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Hub distance	`OUTPUT`	[vetor: linha]	Line vector layer with the attributes of the input features, the identifier of their closest feature and the calculated distance.

Código Python

Algorithm ID: `qgis:distancetonearesthublinetohub`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Distance to nearest hub (points)

Creates a point layer representing the *center* of the input features with the addition of two fields containing the identifier of the nearest feature (based on its center point) and the distance between the points.

Ver também

Distance to nearest hub (line to hub), *Unir atributos por mais próximo*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Source points layer	`INPUT`	[vetor: qualquer]	Vector layer for which the nearest feature is searched
Destination hubs layer	`HUBS`	[vetor: qualquer]	Camada vetorial contendo as feições a serem pesquisadas
Hub layer name attribute	`FIELD`	[campo da tabela: qualquer]	Campo a ser usado para identificar exclusivamente as feições da camada de destino. Usado na tabela de atributos de saída
Unidade de medida	`UNIT`	[enumeração] Padrão: 0	Unidades nas quais relatar a distância até a feição mais próxima: • 0 — Metros • 1 — Pés • 2 — Milhas • 3 — Quilômetros • 4 — Unidades de camada
Hub distance	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output point vector layer with the nearest hub. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Hub distance	`OUTPUT`	[vetor: ponto]	Point vector layer representing the center of the source features with their attributes, the identifier of their closest feature and the calculated distance.

Código Python

Algorithm ID: `qgis:distancetonearesthubpoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Join by lines (hub lines)

Creates hub and spoke diagrams by connecting lines from points on the Spoke layer to matching points in the Hub layer.

Determination of which hub goes with each point is based on a match between the Hub ID field on the hub points and the Spoke ID field on the spoke points.

Se as camadas de entrada não forem camadas de pontos, um ponto na superfície das geometrias será considerado como o local de conexão.

Opcionalmente, podem ser criadas linhas geodésicas, que representam o caminho mais curto na superfície de um elipsóide. Quando o modo geodésico é usado, é possível dividir as linhas criadas no antimeridiano (±180 graus de longitude), o que pode melhorar a renderização das linhas. Além disso, a distância entre os vértices pode ser especificada. Uma distância menor resulta em uma linha mais densa e precisa.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

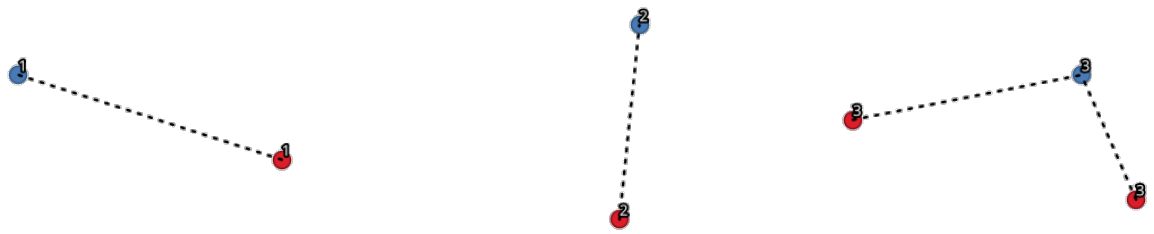


Figura24.72: Unir pontos com base em um campo/atributo comum

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Hub layer	`HUBS`	[vector: geometry]	Camada de entrada
Hub ID field	`HUB_FIELD`	[campo da tabela: qualquer]	Field of the hub layer with ID to join
Hub layer fields to copy (leave empty to copy all fields) Opcional	`HUB_FIELDS`	[campo da tabela: qualquer][list]	The field(s) of the hub layer to be copied. If no field(s) are chosen all fields are taken.
Spoke layer	`SPOKES`	[vector: geometry]	Additional spoke point layer
Spoke ID field	`SPOKE_FIELD`	[campo da tabela: qualquer]	Field of the spoke layer with ID to join
Spoke layer fields to copy (leave empty to copy all fields) Opcional	`SPOKE_FIELDS`	[campo da tabela: qualquer][list]	Field(s) of the spoke layer to be copied. If no fields are chosen all fields are taken.

continua na próxima página

Tabela 24.463 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Criar linhas geodésicas	`GEODESIC`	[boleano] Padrão: Falso	Criar linhas geodésicas (o caminho mais curto na superfície de um elipsóide)
Hub lines	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output hub line vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Distância entre vértices (somente linhas geodésicas)	`GEODESIC_DISTA`	[numeric: double] Padrão: 1000.0 (quilômetros)	Distância entre vértices consecutivos (em quilômetros). Uma distância menor resulta em uma linha mais densa e precisa
Linhas divididas em antimeridiano (±180 graus de longitude)	`ANTIMERID-IAN_SPLIT`	[boleano] Padrão: Falso	Split lines at ±180 degrees longitude (to improve rendering of the lines)

Saídas

Etiqueta	Nome	Tipo	Descrição
Hub lines	`OUTPUT`	[vetor: linha]	The resulting line layer connecting matching points in input layers

Código Python

Algorithm ID: `native:hublines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

K-means clustering

Calculates the 2D distance based k-means cluster number for each input feature.

K-means clustering aims to partition the features into k clusters in which each feature belongs to the cluster with the nearest mean. The mean point is represented by the barycenter of the clustered features.

If input geometries are lines or polygons, the clustering is based on the centroid of the feature.

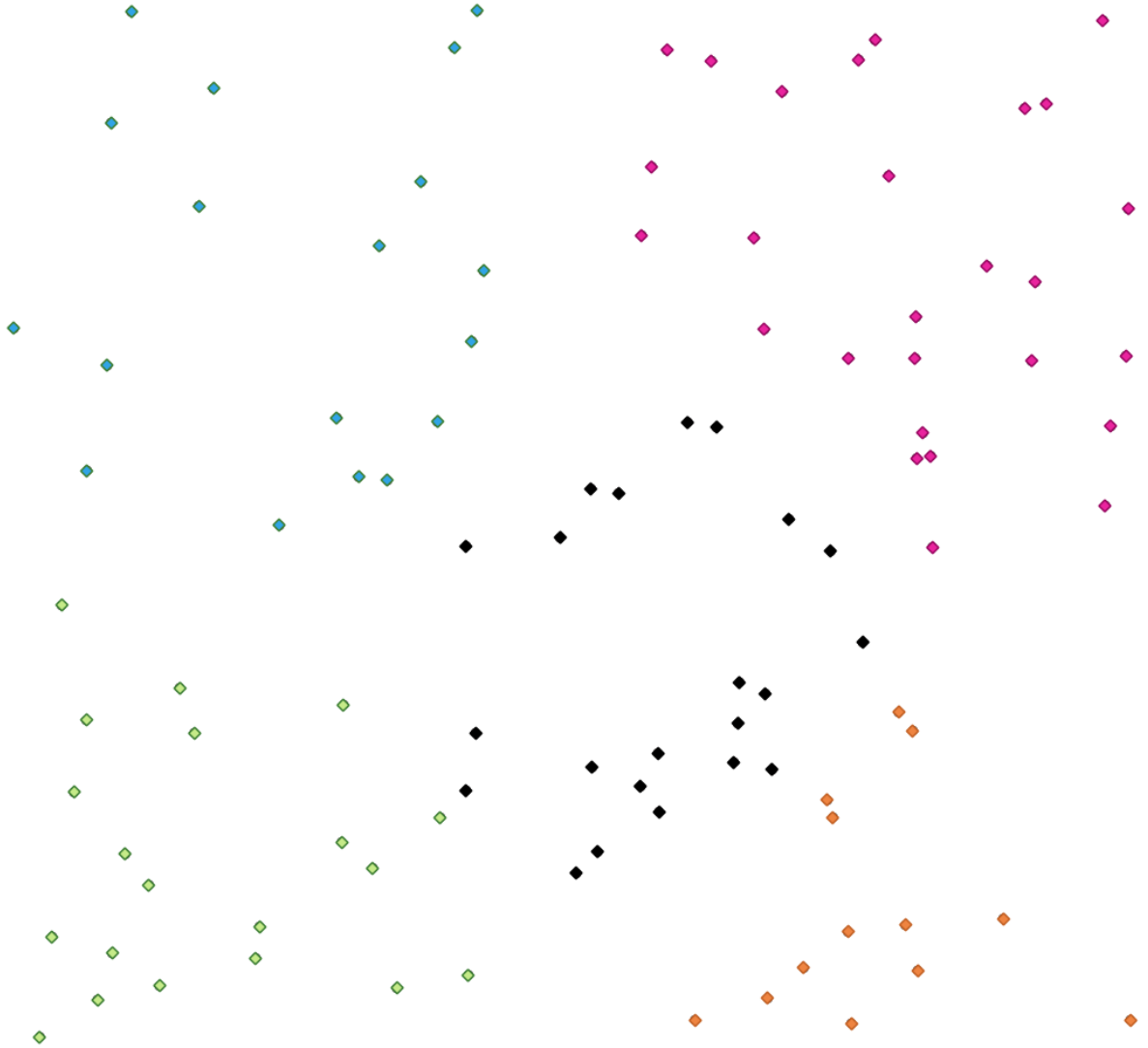


Figura24.73: A five class point clusters

Ver também

DBSCAN clustering, *ST-DBSCAN clustering*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada a ser analisada
Número de agrupamentos	`CLUSTERS`	[numeric: integer] Padrão: 5	Número de clusters a serem criados com as feições
Método	`METHOD`	[enumeração] Padrão: 0	Initial centers selection method. Possible values are: • 0 — Farthest points: the initial centers are selected as the farthest points from each other. • 1 — K-means++: The first center is selected uniformly at random from the set of input features. Subsequent centers are chosen based on a weighted probability distribution that favors features farther from existing centers. This ensures that the initial centers are well-dispersed, reducing the likelihood of poor clustering and improving the algorithm's performance. *References: Arthur, David & Vassilvitskii, Sergei. (2007). K-Means++: The Advantages of Careful Seeding. Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms. 8.*
Clusters	`OUTPUT`	[vetor: o mesmo que a entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the generated clusters. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Cluster field name	`FIELD_NAME`	[string] Default: 'CLUSTER_ID'	Name of the field where the associated cluster number shall be stored
Cluster size field name	`SIZE_FIELD_NAM`	[string] Default: 'CLUSTER_SIZE'	Nome do campo com a contagem de feições no mesmo grupo

Saídas

Etiqueta	Nome	Tipo	Descrição
Clusters	`OUTPUT`	[vetor: o mesmo que a entrada]	Vector layer containing the original features with fields specifying the cluster they belong to and their number in it

Código Python

Algorithm ID: `native:kmeansclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Listar valores únicos

Lista valores exclusivos de um campo de tabela de atributos e conta seu número.

Default menu: *Vector ► Analysis Tools*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada a ser analisada
Target field(s)	`FIELDS`	[campo da tabela: qualquer][list]	Field(s) to analyze
Valores únicos Opcional	`OUTPUT`	[vector: table] Padrão: `[Criar camada temporária]`	Specify the summary table layer with unique values. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Relatório HTML Opcional	`OUTPUT_HTML_FILE`	[html] Padrão: `[Salvar em arquivo temporário]`	HTML report of unique values in the *Processing ► Results viewer*. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Valores únicos	`OUTPUT`	[vector: table]	Summary table layer with unique values
Relatório HTML	`OUTPUT_HTML_FILE`	[html]	HTML report of unique values. Can be opened from the *Processing* ► *Results viewer*
Valores únicos totais	`TOTAL_VALUES`	[numeric: integer]	The number of unique values in the input field
Unique values concatenated	`UNIQUE_VALUES`	[string]	A string with the comma separated list of unique values found in the input field

Código Python

Algorithm ID: `native:listuniquevalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Coordenada(s) média(s)

Calcula uma camada de ponto com o centro de massa das geometrias em uma camada de entrada.

Um atributo pode ser especificado como contendo pesos a serem aplicados a cada feição ao calcular o centro de massa.

Se um atributo for selecionado no parâmetro, as feições serão agrupadas de acordo com os valores deste campo. Em vez de um único ponto com o centro de massa de toda a camada, a camada de saída conterá um centro de massa para as feições em cada categoria.

Default menu: *Vector* ► *Analysis Tools*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Campo de peso Opcional	`WEIGHT`	[campo de tabela: numérico]	Campo a ser usado se você deseja realizar uma média ponderada
*Campo de identificação único**	`UID`	[campo da tabela: qualquer]	Campo único no qual será feito o cálculo da média

continua na próxima página

Tabela 24.471 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Coordenadas médias	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the (point vector) layer for the result. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Coordenadas médias	`OUTPUT`	[vetor: ponto]	Camada de ponto(s) resultante(s)

Código Python

Algorithm ID: `native:meancoordinates`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Análise do vizinho mais próximo

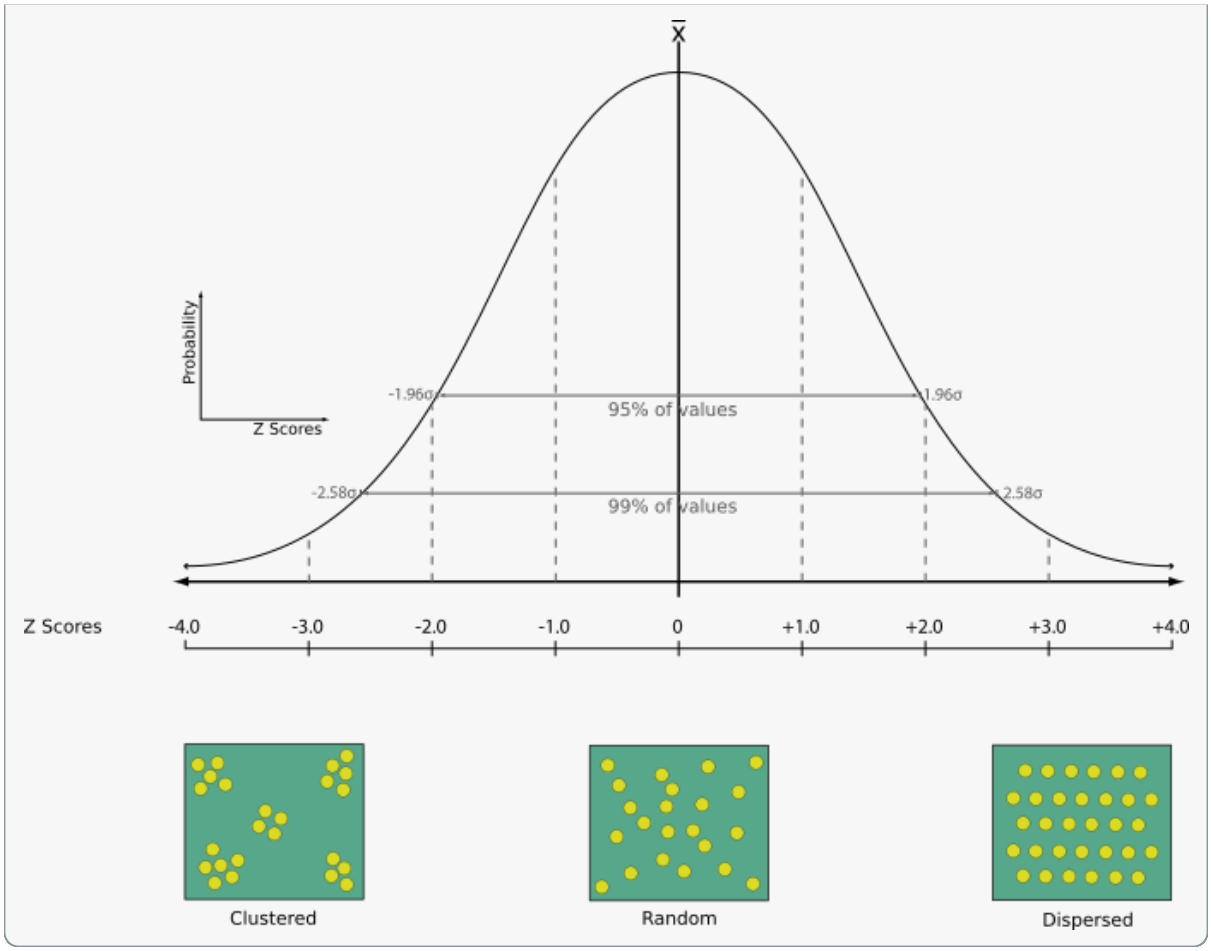
Executa a análise do vizinho mais próximo para uma camada de ponto. A saída informa como seus dados são distribuídos (agrupados, aleatoriamente ou distribuídos).

A saída é gerada como um arquivo HTML com os valores estatísticos calculados:

- Distância média observada
- Distância média esperada
- Índice do vizinho mais próximo
- Número de pontos
- Z-Score: Comparing the Z-Score with the normal distribution tells you how your data are distributed. A low Z-Score means that the data are unlikely to be the result of a spatially random process, while a high Z-Score means that your data are likely to be a result of a spatially random process.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.



Default menu: *Vector* ► *Analysis Tools*

Ver também

Unir atributos por mais próximo

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Camada vetorial de ponto para calcular as estatísticas
Vizinho mais próximo Opcional	`OUTPUT_HTML_FILE`	[html] Padrão: `[Salvar em arquivo temporário]`	Specification of the HTML file for the computed statistics. *One of*: • Ignorar Saída • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Vizinho mais próximo	`OUTPUT_HTML_FILE`	[html]	Arquivo HTML com as estatísticas computadas
Distância média observada	`OBSERVED_MD`	[numeric: double]	Distância média observada
Distância média esperada	`EXPECTED_MD`	[numeric: double]	Distância média esperada
Índice do vizinho mais próximo	`NN_INDEX`	[numeric: integer]	Índice do vizinho mais próximo
Número de pontos	`POINT_COUNT`	[numeric: integer]	Número de pontos
Z-Score	`Z_SCORE`	[numeric: double]	Z-Score

Código Python

Algorithm ID: `native:nearestneighbouranalysis`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Análise de sobreposição

Calcula a área e a cobertura percentual pela qual as feições de uma camada de entrada são sobrepostas por feições de uma seleção de camadas de sobreposição.

Novos atributos são adicionados à camada de saída relatando a área total de sobreposição e a porcentagem da feição de entrada sobreposta por cada uma das camadas de sobreposição selecionadas.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	A camada de entrada.
Overlay layers	`LAYERS`	[vector: polygon] [list]	As camadas de sobreposição.

continua na próxima página

Tabela 24.475 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Overlap	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Grid size Opcional	`GRID_SIZE`	[numeric: double] Padrão: Não definido	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Saídas

Etiqueta	Nome	Tipo	Descrição
Overlap	`OUTPUT`	[o mesmo que entrada]	The output layer with additional fields reporting the overlap (in map units and percentage) of the input feature overlapped by each of the selected layers.

Código Python

Algorithm ID: `native:calculatevectoroverlaps`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Shortest line between features

Creates a line layer as the shortest line between the source and the destination layer. By default only the first nearest feature of the destination layer is taken into account. The n-nearest neighboring features number can be specified. If a maximum distance is specified, then only features which are closer than this distance will be considered.

The output features will contain all the source layer attributes, all the attributes from the n-nearest feature and the additional field of the distance.

Importante

This algorithm uses purely Cartesian calculations for distance, and does not consider geodetic or ellipsoid properties when determining feature proximity. The measurement and output coordinate system is based on the coordinate system of the source layer.

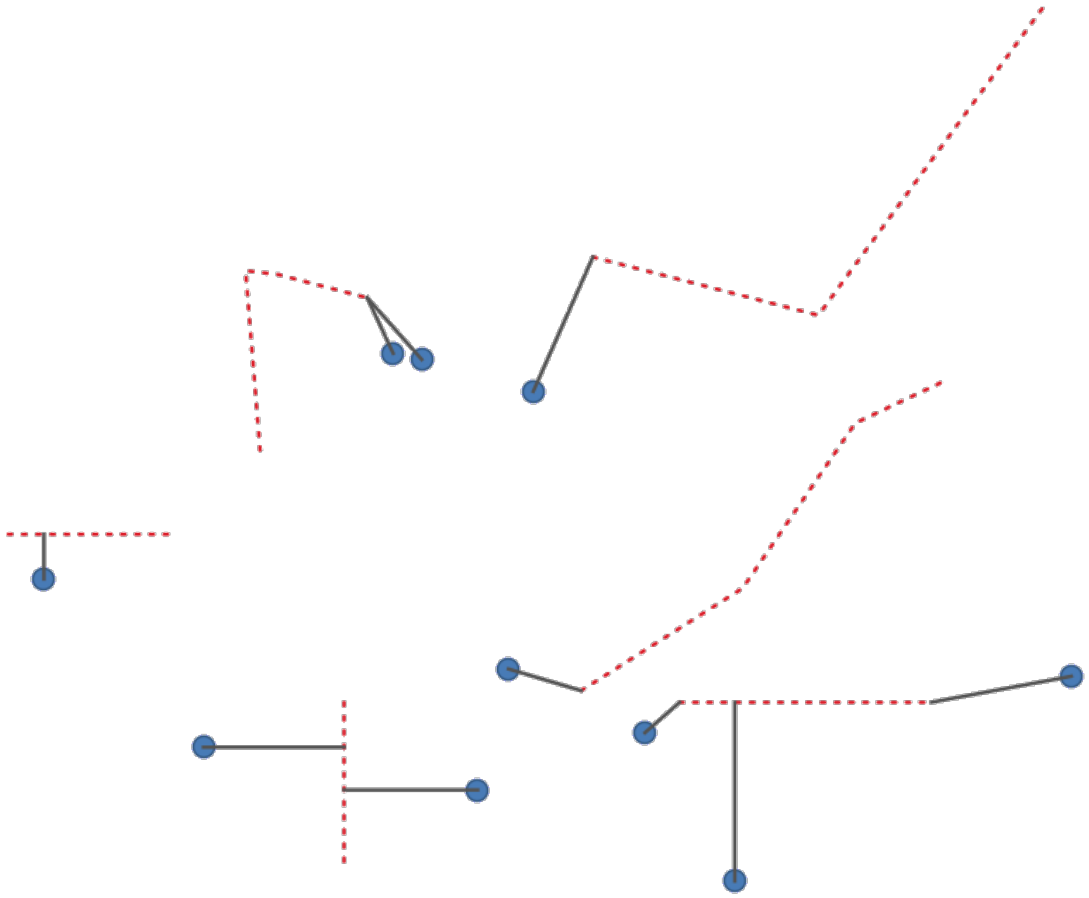


Figura24.74: Shortest line from point features to lines

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Source layer	`SOURCE`	[vector: geometry]	Origin layer for which to search for nearest neighbors
Destination layer	`DESTINATION`	[vector: geometry]	Target Layer in which to search for nearest neighbors
Método	`METHOD`	[enumeração] Padrão: 0	Shortest distance calculation method Possible values are: • 0 — Distance to nearest point on feature • 1 — Distance to feature centroid
Maximum number of neighbors	`NEIGHBORS`	[numeric: integer] Padrão: 1	Maximum number of neighbors to look for
Distância máxima Opcional	`DISTANCE`	[numeric: double]	Only destination features which are closer than this distance will be considered.
Shortest lines	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[vetor: linha]	Line vector layer joining source features to their nearest neighbor(s) in the destination layer. Contains all attributes for both source and destination features, and the computed distance.

Código Python

Algorithm ID: `native:shortestline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

ST-DBSCAN clustering

Clusters point features based on a 2D implementation of spatiotemporal Density-based clustering of applications with noise (ST-DBSCAN) algorithm.

Ver também

DBSCAN clustering, *K-means clustering*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Camada a ser analisada
Date/time field	`DATE-TIME_FIELD`	[tablefield: date]	Campo contendo as informações temporais
Tamanho mínimo do conjunto	`MIN_SIZE`	[numeric: integer] Padrão: 5	Número mínimo de feições para gerar um conjunto
Maximum distance between clustered points	`EPS`	[numeric: double] Padrão: 1,0	Distance beyond which two features can not belong to the same cluster (eps)
Duração máxima de tempo entre pontos agrupados	`EPS2`	[numeric: double] Padrão: 0.0 (dias)	Time duration beyond which two features can not belong to the same cluster (eps2). Available time units are milliseconds, seconds, minutes, hours, days and weeks.
Clusters	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the vector layer for the result of the clustering. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Treat border points as noise (DBSCAN*)	`DBSCAN*`	[boleano] Padrão: Falso	If checked, points on the border of a cluster are themselves treated as unclustered points, and only points in the interior of a cluster are tagged as clustered.
Cluster field name	`FIELD_NAME`	[string] Default: 'CLUSTER_ID'	Name of the field where the associated cluster number shall be stored

continua na próxima página

Tabela 24.481 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Cluster size field name	`SIZE_FIELD_NAM`	[string] Default: 'CLUSTER_SIZE'	Nome do campo com a contagem de feições no mesmo grupo

Saídas

Etiqueta	Nome	Tipo	Descrição
Clusters	`OUTPUT`	[vetor: ponto]	Vector layer containing the original features with a field setting the cluster they belong to
Número de agrupamentos	`NUM_CLUSTERS`	[numeric: integer]	O número de agrupamentos descobertos

Código Python

Algorithm ID: `native:stdbscanclustering`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Estatísticas por categorias

Calculates statistics of a field depending on a parent class. The parent class is a combination of values from other fields.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada vetorial de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada com classes e valores únicos
Field to calculate statistics on (if empty, only count is calculated) Opcional	`VALUES_FIELD_NAME`	[campo da tabela: qualquer]	Se estiver vazio apenas a contagem será calculada
Campo(s) com categorias	`CATEGORIES_FIELD_N`	[campo da tabela: qualquer][list]	Os campos que (combinados) definem as categorias

continua na próxima página

Tabela 24.483 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Estatísticas por categoria	`OUTPUT`	[vector: table] Padrão: `[Criar camada temporária]`	Specify the output table for the generated statistics. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Estatísticas por categoria	`OUTPUT`	[vector: table]	Tabela contendo as estatísticas

Depending on the type of the field being analyzed, the following statistics are returned for each grouped value:

Estatísticas	String	Numeric	Data
Contagem (`CONTAGEM`)	✓	✓	✓
Valores únicos (`ÚNICO`)	✓		✓
Empty (null) values (`EMPTY`)	✓		✓
Non-empty values (`FILLED`)	✓		✓
Valor mínimo (`MIN`)	✓	✓	✓
Valor máximo (`MAX`)	✓	✓	✓
Range (`RANGE`)		✓	
Soma (`SOMA`)		✓	
Valor médio (`MEDIA`)		✓	
Valor da mediana (`MEDIANA`)		✓	
Standard Deviation (`STD_DEV`)		✓	
Coeficiente de variação (`CV`)		✓	
Minority (rarest occurring value - `MINORITY`)		✓	
Majority (most frequently occurring value - `MAJORITY`)		✓	
Primeiro quartil (`PRIMEIROQUARTIL`)		✓	
Terceiro quartil (`TERCEIROQUARTIL`)		✓	
Inter Quartile Range (`IQR`)		✓	
Comprimento mínimo (`COMPRIMENTO_MIN`)	✓		
Comprimento médio (``COMPRIMENTO_MÉDIO`)	✓		
Comprimento máximo (`COMPRIMENTO_MAX`)	✓		

Código Python

Algorithm ID: `qgis:statisticsbycategories`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Soma dos comprimentos das linhas

Takes a polygon layer and a line layer and measures the total length of lines and the total number of them that cross each polygon.

The resulting layer has the same features as the input polygon layer, but with two additional attributes containing the length and count of the lines across each polygon.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Allows features in-place modification 1 das características dos polígonos.

Default menu: *Vector* ► *Analysis Tools*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Linhas	`LINES`	[vetor: linha]	Input vector line layer
Polígonos	`POLYGONS`	[vetor: polígono]	Camada de vetor de polígono
Nome do campo de comprimento das linhas	`LEN_FIELD`	[string] Padrão: 'COMPRIMENTO'	Name of the field for the lines length
Lines count field name	`COUNT_FIELD`	[string] Default: 'COUNT'	Name of the field for the lines count
Comprimento da linha	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output polygon layer with generated statistics. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Comprimento da linha	`OUTPUT`	[vetor: polígono]	Polygon output layer with fields of lines length and line count

Código Python

Algorithm ID: `native:sumlinelengths`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.23 Vector coverage

Dissolve coverage

Operates on a coverage (represented as a set of polygon features with exactly matching edge geometry) to dissolve (union) the geometries. It provides a heavily optimized approach for unioning these features compared with the standard Dissolve tools.

Ver também

Dissolver

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	insira a camada vetorial de polígonal.
Dissolvido	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Dissolvido	`OUTPUT`	[vetor: polígono]	The output polygon vector layer with dissolved geometries.

Código Python

Algorithm ID: `native:coverageunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Simplify coverage

Operates on a coverage (represented as a set of polygon features with exactly matching edge geometry) to apply a Visvalingam–Whyatt simplification to the edges, reducing complexity in proportion with the provided tolerance, while retaining a valid coverage (i.e. no edges will cross or touch after the simplification). Geometries will never be removed, but they may be simplified down to just a triangle. Also, some geometries (such as polygons which have too few non-repeated points) will be returned unchanged.

If the input dataset is not a valid coverage due to overlaps, it will still be simplified, but invalid topology such as crossing edges will still be invalid.

Atenção

Running this algorithm requires QGIS installed with GEOS >= 3.12 (see *Help* ► *About* menu).

Ver também

Simplificar, *Suave*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	insira a camada vetorial de polígonal.
Tolerância	`TOLERANCE`	[numeric: double] Padrão: 0.0	The maximum distance (in unit of choice) between two consecutive vertices to get merged.
Preserve boundary	`PRESERVE_BOUNDARY`	[boleano] Padrão: Falso	When enabled, the outside edges of the coverage will be preserved without simplification

continua na próxima página

Tabela 24.490 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Simplificado	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Simplificado	`OUTPUT`	[vetor: polígono]	The output polygon vector layer, with a lower number of vertices.

Código Python

Algorithm ID: `native:coveragesimplify`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Validate coverage

Analyzes a coverage (represented as a set of polygon features with exactly matching edge geometry) to find places where the assumption of exactly matching edges is not met. Invalidity includes polygons that overlap or that have gaps smaller than the specified gap width.

Atenção

Running this algorithm requires QGIS installed with GEOS >= 3.12 (see *Help* ► *About* menu).

Ver também

Verificar a validade, *Small gaps*, *Overlaps*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	insira a camada vetorial de polígonal.
Gap width	`GAP_WIDTH`	[numeric: double] Padrão: 0.0	The maximum width of gap to detect
Invalid edges	`INVALID_EDGES`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Invalid edges	`OUTPUT`	[vetor: linha]	The output line vector layer showing error edges for each invalid polygon.
Validity result	`IS_VALID`	[boleano]	Returns whether the coverage is valid or not.

Código Python

Algorithm ID: `native:coveragevalidate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.24 Criação de vetor

Array of offset (parallel) lines

Creates copies of line features in a layer, by creating multiple offset versions of each feature. Each new version is incrementally offset by a specified distance.

A distância positiva deslocará as linhas para a esquerda e as distâncias negativas as deslocarão para a direita.

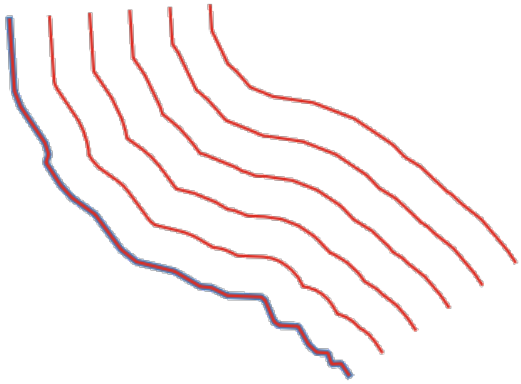


Figura24.75: Em azul a camada de origem, em vermelho a de deslocamento

Permite features in-place modification 1 da linha de características.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

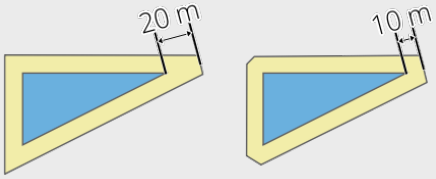
Linhas de desvio., *Array of translated features*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Input line vector layer to use for the offsets.
Número de feições a serem criadas	`COUNT`	[numeric: integer] Padrão: 10	Number of offset copies to generate for each feature
Offset step distance	`OFFSET`	[numeric: double] Padrão: 1,0	Distance between two consecutive offset copies
Offset lines	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line layer with offset features. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Segmentos	`SEGMENTS`	[numeric: double] Padrão: 8	Number of line segments to use to approximate a quarter circle when creating rounded offsets
Join style	`JOIN_STYLE`	[enumeração] Padrão: 0	Specify whether round, miter or beveled joins should be used when offsetting corners in a line. One of: • 0 — Round • 1 — Miter (mitra) • 2 — Bevel (chanfrado) Figura24.76: Estilos de junção redonda, serrada e chanfrada
Miter limit	`MITER_LIMIT`	[numeric: double] Padrão: 2.0	Configura a distância máxima da geometria de deslocamento para usar quando criando uma junção mitered como um fato da distância de deslocamento (somente aplicado para junções do estilo miter). Mínimo: 1.0  Figura24.77: Um buffer de 10m com um limite de buffer de 2 e 10m com um limite de 1.

Saídas

Etiqueta	Nome	Tipo	Descrição
Offset lines	`OUTPUT`	[vetor: linha]	Output line layer with the original and offset features. Attributes will be copied to the corresponding outputs. The following attributes related to the input parameters will be added: `instance` indexing the `COUNT` offset lines and `offset` designating the offset distance from the original line.

Código Python

Algorithm ID: `native:arrayoffsetlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Array of translated features

Creates copies of features in a layer by creating multiple translated versions of each. Each copy is incrementally displaced by a preset amount in the X, Y and/or Z axis.

Os valores M presentes na geometria também podem ser traduzidos.

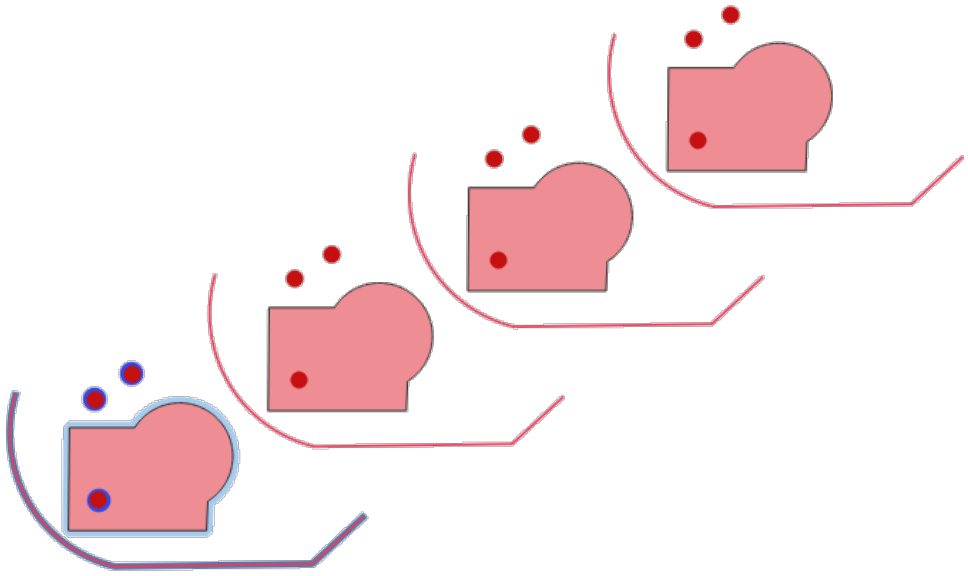


Figura24.78: Input layers in blue tones, output layers with translated features in red tones

Allows features in-place modification 1 de ponto, linha e características poligonais.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Transladar, *Array of offset (parallel) lines*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Input vector layer to translate
Número de feições a serem criadas	`COUNT`	[numeric: integer] Padrão: 10	Número de cópias a serem geradas para cada feição
Step distance (x-axis)	`DELTA_X`	[numeric: double] Padrão: 0.0	Delocamento a ser aplicado ao eixo X
Step distance (y-axis)	`DELTA_Y`	[numeric: double] Padrão: 0.0	Delocamento a ser aplicado ao eixo Y
Step distance (z-axis)	`DELTA_Z`	[numeric: double] Padrão: 0.0	Displacement to apply on the Z axis
Step distance (m values)	`DELTA_M`	[numeric: double] Padrão: 0.0	Displacement to apply on M
Transladado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Output vector layer with translated (moved) copies of the features. The original features are also copied. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Transladado	`OUTPUT`	[o mesmo que entrada]	Output vector layer with translated (moved) copies of the features. The original features are also copied.

Código Python

Algorithm ID: `native:arraytranslatedfeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Create grid

Creates a vector layer with a grid covering a given extent. Grid cells can have different shapes:

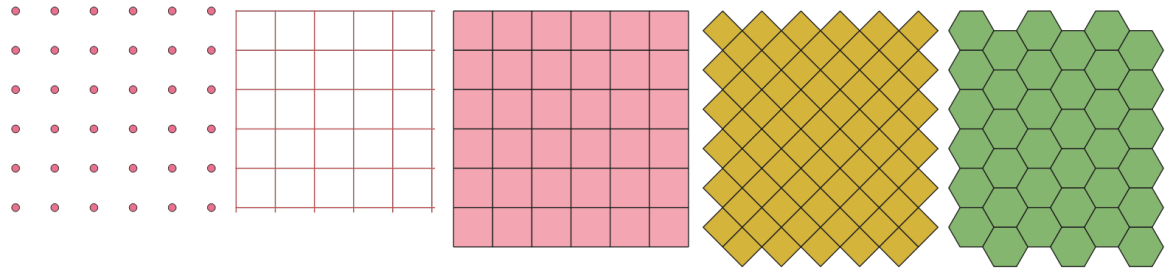


Figura24.79: Different grid cell shapes applied to the same extent, without overlaps

The size and/or placement of each element in the grid is defined using a horizontal and vertical spacing. The CRS of the output layer must be defined. The grid extent and the spacing values must be expressed in the coordinates and units of this CRS.

Menu padrão: *Vetor ► Ferramentas de pesquisa*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Grid type	`TYPE`	[enumeração] Padrão: 0	Shape of the grid. One of: • 0 — Ponto • 1 — Linha • 2 — Retângulo (polígono) • 3 — Diamante (polígono) • 4 — Hexágono (polígono)

continua na próxima página

Tabela 24.499 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Grid extent	`EXTENT`	[extensão]	Extent of the grid Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Espaçamento horizontal	`HSPACING`	[numeric: double] Padrão: 1,0	Size of a grid cell on the X-axis
Espaçamento vertical	`VSPACING`	[numeric: double] Padrão: 1,0	Size of a grid cell on the Y-axis
Horizontal overlay	`HOVERLAY`	[numeric: double] Padrão: 0.0	Overlay distance between two consecutive grid cells on the X-axis
Vertical overlay	`VOVERLAY`	[numeric: double] Padrão: 0.0	Overlay distance between two consecutive grid cells on the Y-axis
Grid CRS	`CRS`	[src] Default: *Project CRS*	Coordinate reference system to apply to the grid
Grid	`OUTPUT`	[vector: geometry] Padrão: `[Criar camada temporária]`	Resulting vector grid layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Grid	`OUTPUT`	[vector: geometry]	Resulting vector grid layer. The output geometry type (point, line or polygon) depends on the *Grid type*. Features are created from top to bottom, left to right. The attribute table is filled with: • an `id` • coordinates on the `left`, `right`, `top` and `bottom` sides • and their placement in the grid: `row_index` and `column_index` (available for point, rectangle and hexagon grid types)

Código Python

Algorithm ID: `native:creategrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Create layer from extent

Creates a new vector layer that contains a single feature with geometry matching the extent of the input layer.

It can be used in models to convert a literal extent (`xmin`, `xmax`, `ymin`, `ymax` format) into a layer which can be used for other algorithms which require a layer based input.

Ver também

Create layer from point

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Extent (xmin, xmax, ymin, ymax)	`INPUT`	[extensão]	Input extent Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Extent	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Extent	`OUTPUT`	[vetor: polígono]	The output (extent) vector layer

Código Python

Algorithm ID: `native:extenttolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Create layer from point

Creates a new vector layer that contains a single feature with geometry matching a point parameter. It can be used in models to convert a point into a point layer for algorithms which require a layer based input.

Ver também

Create layer from extent

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Point	`INPUT`	[coordinates]	Input point, including CRS info (example: `397254,6214446 [EPSG:32632]`). If the CRS is not provided, the Project CRS will be used. The point can be specified by clicking on the map canvas.
Point	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Point	`OUTPUT`	[vetor: ponto]	The output point vector layer containing the input point.

Código Python

Algorithm ID: `native:pointtolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Create points layer from table

Creates points layer from a table with columns that contain coordinates fields.

Besides X and Y coordinates you can also specify Z and M fields.

> **Aviso**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Input vector layer or a table.
Campo X	`XFIELD`	[campo da tabela: qualquer]	Campo contendo a coordenada X
Campo Y	`YFIELD`	[campo da tabela: qualquer]	Campo contendo a coordenada Y
Campo Z Opcional	`ZFIELD`	[campo da tabela: qualquer]	Campo contendo a coordenada Z
Campo M Opcional	`MFIELD`	[campo da tabela: qualquer]	Campo contendo o valor M
SRC Alvo	`TARGET_CRS`	[src] Padrão: `EPSG:4326`	Coordinate reference system to use for layer. The provided coordinates are assumed to be compliant.
Pontos da tabela	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the resulting point layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos da tabela	`OUTPUT`	[vetor: ponto]	A camada de pontos resultante

Código Python

Algorithm ID: `native:createpointslayerfromtable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Generate points (pixel centroids) along line

Generates a point vector layer from an input raster and line layer.

The points correspond to the pixel centroids that intersect the line layer.

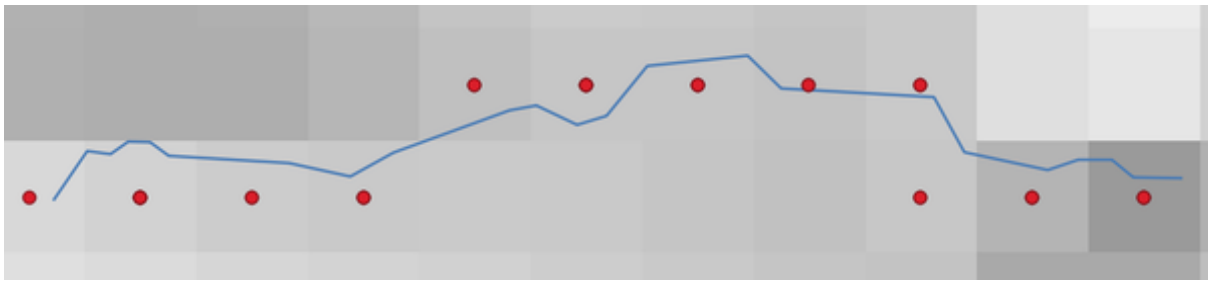


Figura24.80: Points of the pixel centroids

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada raster	`INPUT_RASTER`	[raster]	Camada raster de entrada
Camada vetorial	`INPUT_VECTOR`	[vetor: linha]	Insira a linha de camada vetorial.
Pontos ao longo da linha	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Resulting point layer with pixel centroids. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos ao longo da linha	`OUTPUT`	[vetor: ponto]	Camada de ponto resultante com centróides de pixel

Código Python

Algorithm ID: `qgis:generatepointspixelcentroidsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Gerar pontos (centróides de pixel) dentro do polígono

Gera uma camada vetorial de ponto a partir de uma camada raster e de polígono de entrada.

Os pontos correspondem aos centróides de pixel que cruzam a camada do polígono.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

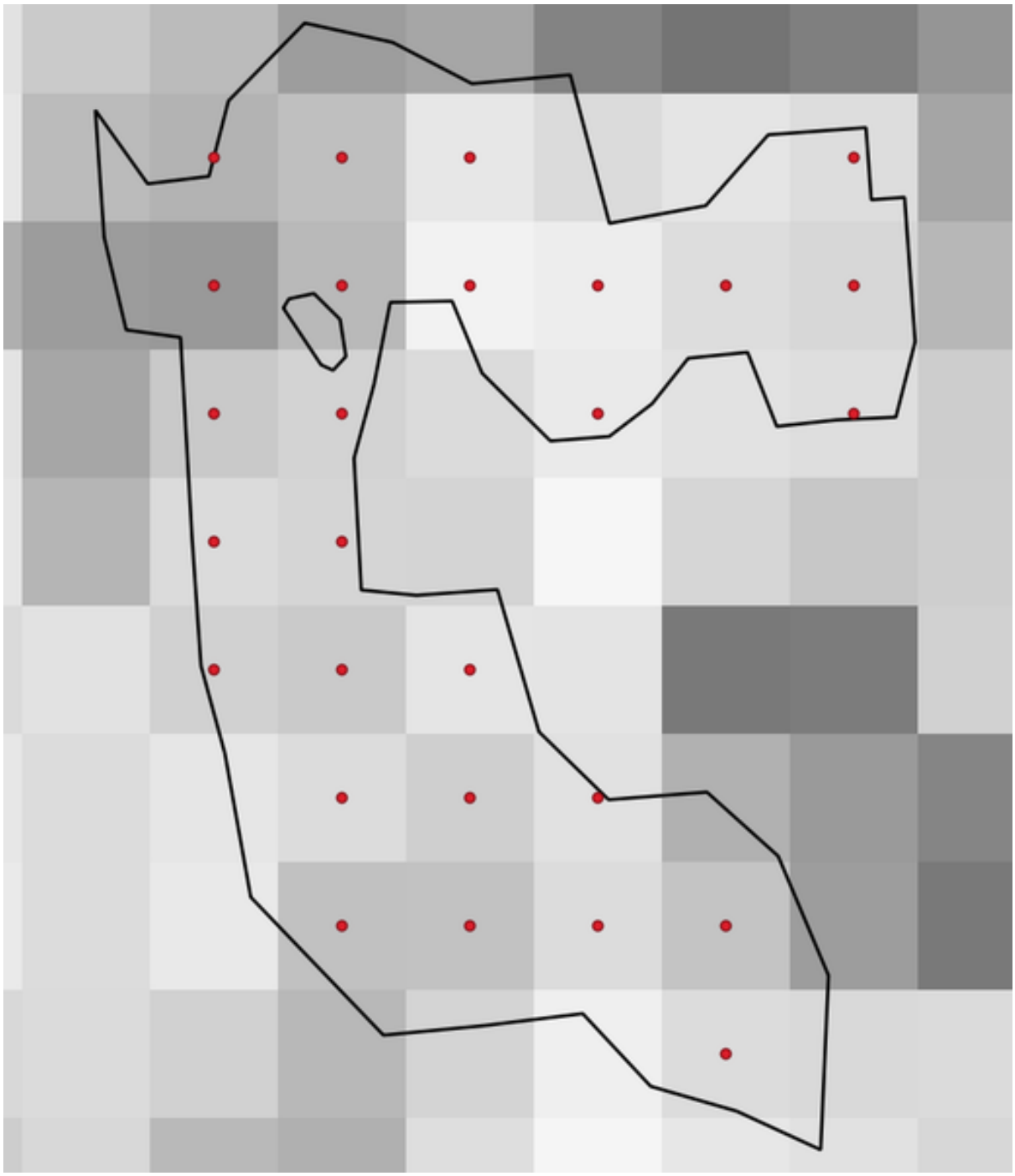


Figura24.81: Points of the pixel centroids

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada raster	`INPUT_RASTER`	[raster]	Camada raster de entrada
Camada vetorial	`INPUT_VECTOR`	[vetor: polígono]	insira a camada vetorial de polígonal.
Pontos dentro de polígonos	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Resulting point layer of pixel centroids. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos dentro de polígonos	`OUTPUT`	[vetor: ponto]	Camada de ponto resultante de centróides de pixel

Código Python

Algorithm ID: `native:generatepointspixelcentroidsinsidepolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Import geotagged photos

Creates a point layer corresponding to the geotagged locations from JPEG or HEIC/HEIF images from a source folder.

The point layer will contain a single PointZ feature per input file from which the geotags could be read. Any altitude information from the geotags will be used to set the point's Z value.

Besides longitude and latitude also altitude, direction and timestamp information, if present in the photo, will be added to the point as attributes.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Input folder	`FOLDER`	[pasta]	Path to the source folder containing the geotagged photos
Scan recursively	`RECURSIVE`	[boleano] Padrão: Falso	If checked, the folder and its subfolders will be scanned
Fotos Opcional	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the point vector layer for the geotagged photos. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Invalid photos table Opcional	`INVALID`	[vector: table] Default: `[Pular saída]`	Specify the table of unreadable or non-geotagged photos. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Fotos	`OUTPUT`	[vetor: ponto]	Point vector layer with geotagged photos. The form of the layer is automatically filled with paths and photo previews settings.
Invalid photos table Opcional	`INVALID`	[vector: table]	Table of unreadable or non-geotagged photos can also be created.

Código Python

Algorithm ID: `native:importphotos`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pontos para o caminho

Converts a point layer to a line layer, by joining points in an order defined by an expression or a field in the input point layer.

Points can be grouped by a field or an expression to distinguish line features.

In addition to the line vector layer, a text file is output that describes the resulting line as a start point and a sequence of bearings / directions (relative to azimuth) and distances.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada de ponto	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Criar caminhos fechados	`CLOSE_PATH`	[boleano] Padrão: Falso	If checked, the first and last points of the line will be connected and close the generated path
Order expression Opcional	`ORDER_EXPRESSION`	[expressão]	Campo ou expressão que fornece a ordem para conectar os pontos no caminho. Se não for definido, o ID do recurso (`$id`) é usado.
Sort text containing numbers naturally	`NATURAL_SORT`	[boleano] Padrão: Falso	Se marcado, classifica naturalmente as feições com base na expressão fornecida (ou seja, 'a9' < 'a10').
Path group expression Opcional	`GROUP_EXPRESSI`	[expressão]	As feições de ponto de mesmo valor no campo ou expressão serão agrupadas na mesma linha. Se não for definido, um único caminho é desenhado com todos os pontos de entrada.
Caminhos	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the line vector layer of the path. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Directory for text output Opcional	`OUTPUT_TEXT_DIR`	[pasta] Default: `[Pular saída]`	Specify the directory that will contain the description files of points and paths. *One of*: • Ignorar Saída • Salvar como Diretório Temporário • Salvar no Diretório

Saídas

Etiqueta	Nome	Tipo	Descrição
Caminhos	`OUTPUT`	[vetor: linha]	Line vector layer of the path
Directory for text output	`OUTPUT_TEXT_DIR`	[pasta]	Diretório contendo arquivos de descrição de pontos e caminhos

Código Python

Algorithm ID: `native:pointstopath`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pontos aleatórios ao longo da linha

Cria uma nova camada de pontos, com pontos colocados nas linhas de outra camada.

Para cada linha na camada de entrada, um determinado número de pontos é adicionado à camada resultante. O procedimento para adicionar um ponto é:

1. selecionar aleatoriamente uma feição de linha da camada de entrada
2. if the feature is multi-part, randomly select a part of it
3. selecionar aleatoriamente um segmento dessa linha
4. selecionar aleatoriamente uma posição nesse segmento.

The procedure means that curved parts of the lines (with relatively short segments) will get more points than straight parts (with relatively long segments), as demonstrated in the illustration below, where the output of the *Random points along lines* algorithm can be compared with the output of the *Random points on lines* algorithm (that produces points with an, on average, even distribution along the lines).

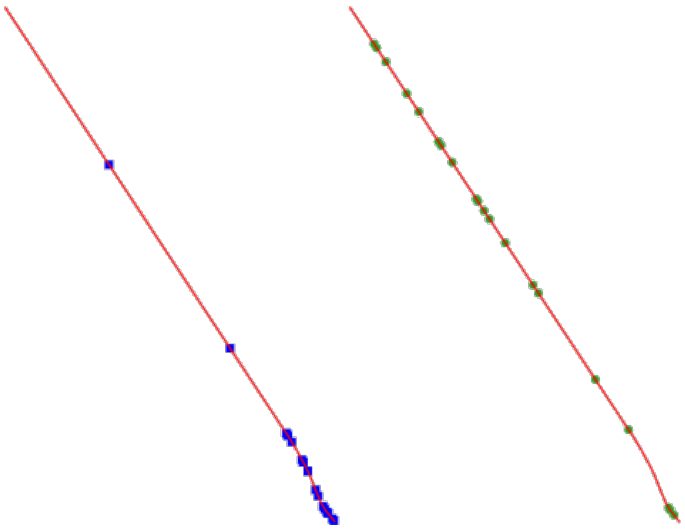


Figura24.82: Example algorithm output. Left: *Random points along line*, right: *Random points on lines*

Uma distância mínima pode ser especificada, para evitar que os pontos fiquem muito próximos uns dos outros.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Pontos aleatórios nas linhas

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada de ponto	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Número de pontos	`POINTS_NUMBER`	[numeric: integer] Padrão: 1	Número de pontos para criar
Distância mínima entre pontos	`MIN_DISTANCE`	[numeric: double] Padrão: 0.0	A distância mínima entre os pontos
Pontos aleatórios	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	The output random points. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos aleatórios	`OUTPUT`	[vetor: ponto]	The output random points layer.

Código Python

Algorithm ID: `qgis:qgisrandompointsalongline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Random points in extent

Cria uma nova camada de pontos com um determinado número de pontos aleatórios, todos dentro de uma determinada extensão.

Um fator de distância pode ser especificado, para evitar que os pontos fiquem muito próximos uns dos outros. Se a distância mínima entre os pontos impossibilita a criação de novos pontos, a distância pode ser diminuída ou o número máximo de tentativas pode ser aumentado.

Menu padrão: *Vetor ► Ferramentas de pesquisa*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Input extent	`EXTENT`	[extensão]	Map extent for the random points Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Número de pontos	`POINTS_NUMBER`	[numeric: integer] Padrão: 1	Número de pontos para criar
Distância mínima entre pontos	`MIN_DISTANCE`	[numeric: double] Padrão: 0.0	A distância mínima entre os pontos
SRC Alvo	`TARGET_CRS`	[src] Default: *Project CRS*	CRS of the random points layer
Pontos aleatórios	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	The output random points. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Maximum number of search attempts given the minimum distance	`MAX_ATTEMPTS`	[numeric: integer] Padrão: 200	Maximum number of attempts to place the points

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos aleatórios	`OUTPUT`	[vetor: ponto]	The output random points layer.

Código Python

Algorithm ID: `native:randompointsinextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Random points in layer bounds

Creates a new point layer with a given number of random points, all of them within the extent of a given layer.

Uma distância mínima pode ser especificada, para evitar que os pontos fiquem muito próximos uns dos outros.

Menu padrão: *Vetor ► Ferramentas de pesquisa*

> **Aviso**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	Input polygon layer defining the area
Número de pontos	`POINTS_NUMBER`	[numeric: integer] Padrão: 1	Número de pontos para criar
Distância mínima entre pontos	`MIN_DISTANCE`	[numeric: double] Padrão: 0.0	A distância mínima entre os pontos

continua na próxima página

Tabela 24.520 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Pontos aleatórios	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	The output random points. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos aleatórios	`OUTPUT`	[vetor: ponto]	The output random points layer.

Código Python

Algorithm ID: `qgis:randompointsinlayerbounds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pontos aleatórios em polígonos

Cria uma camada de pontos com pontos colocados dentro dos polígonos de outra camada.

For each feature (polygon / multi-polygon) geometry in the input layer, the given number of points is added to the result layer.

Per feature and global minimum distances can be specified in order to avoid points being too close in the output point layer. If a minimum distance is specified, it may not be possible to generate the specified number of points for each feature. The total number of generated points and missed points are available as output from the algorithm.

The illustration below shows the effect of per feature and global minimum distances and zero/non-zero minimum distances (generated with the same seed, so at least the first point generated will be the same).

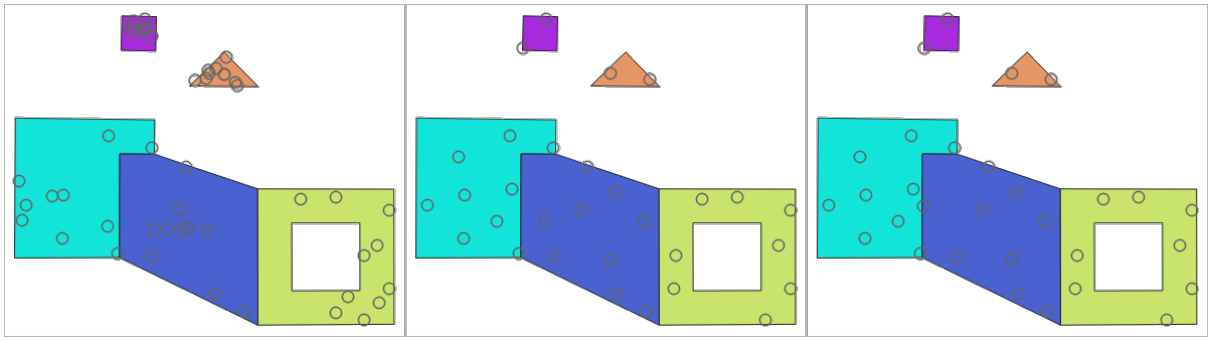


Figura24.83: Ten points per polygon feature, *left*: min. distances = 0, *middle*: min.distances = 1, *right*: min. distance = 1, global min. distance = 0

O número máximo de tentativas por ponto pode ser especificado. Isso só é relevante para distância mínima diferente de zero.

A seed for the random number generator can be provided, making it possible to get identical random number sequences for different runs of the algorithm.

The attributes of the polygon feature on which a point was generated can be included (*Include polygon attributes*).

Se você quiser aproximadamente a mesma densidade de pontos para todas as características, você pode definir o número de pontos usando a área da geometria da característica poligonal.

Ver também

Pontos aleatórios dentro de polígonos

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Inserir camada de polígono	`INPUT`	[vetor: linha]	insira a camada vetorial de polígonal.
Número de pontos para cada feição	`POINTS_NUMBER`	[numeric: integer] Padrão: 1	Número de pontos para criar
Distância mínima entre pontos Opcional	`MIN_DISTANCE`	[numeric: double] Padrão: 0.0	A distância mínima entre pontos dentro de uma feição de polígono
Pontos aleatórios em polígonos	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	The output random points. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Distância mínima global entre pontos Opcional	MIN_DISTANCE_G	[numeric: double] Padrão: 0.0	A distância mínima global entre os pontos. Deve ser menor que a *Distância mínima entre pontos (por feição)* para que esse parâmetro tenha efeito.
Maximum number of search attempts (for Min. dist. > 0) Opcional	MAX_TRIES_PER_	[numeric: integer] Padrão: 10	The maximum number of tries per point. Only relevant if the minimum distance between points is set (and greater than 0).
Random seed Opcional	SEED	[numeric: integer] Padrão: Não definido	The seed to use for the random number generator.
Incluir atributos de polígono	INCLUDE_POLYGON_	[boleano] Padrão: Verdadeiro	Se definido, um ponto obterá os atributos da linha na qual é colocado.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos aleatórios em polígonos	OUTPUT	[vetor: ponto]	The output random points layer.
Número de características com geometria vazia ou sem geometria	FEATURES_WITH_EMP	[numeric: integer]	
Número total de pontos gerados	OUTPUT_POINTS	[numeric: integer]	
Número de pontos faltantes	POINTS_MISSED	[numeric: integer]	O número de pontos que não puderam ser gerados devido à restrição de distância mínima.
Número de feições com pontos faltantes	POLYGONS_WITH_MISS	[numeric: integer]	Não incluindo feições com geometria vazia ou sem geometria

Código Python

Algorithm ID: `native:randompointsinpolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pontos aleatórios dentro de polígonos

Creates a new point layer with a given number of random points inside each polygon of the input polygon layer.

Duas estratégias de amostragem estão disponíveis:

- Contagem de pontos: número de pontos para cada feição
- Densidade de pontos: densidade de pontos para cada feição

Uma distância mínima pode ser especificada, para evitar que os pontos fiquem muito próximos uns dos outros.

Menu padrão: *Vetor* ► *Ferramentas de pesquisa*

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Pontos aleatórios em polígonos

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	insira a camada vetorial de polígonal.
Estratégia de amostragem	`STRATEGY`	[enumeração] Padrão: 0	Estratégia de amostragem para utilizar. Uma de: • 0 — Contagem de pontos: número de pontos para cada feição • 1 — Densidade de pontos: densidade de pontos para cada feição
Point count or density	`VALUE`	[numeric: double] Padrão: 1,0	The number or density of points, depending on the chosen *Sampling strategy*.
Distância mínima entre pontos	`MIN_DISTANCE`	[numeric: double] Padrão: 0.0	A distância mínima entre os pontos
Pontos aleatórios	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	The output random points. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos aleatórios	`OUTPUT`	[vetor: ponto]	The output random points layer.

Código Python

Algorithm ID: `qgis:randompointsinsidepolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pontos aleatórios nas linhas

Cria uma camada de pontos com pontos colocados nas linhas de outra camada.

Para cada geometria de feição (linha / multilinha) na camada de entrada, o número determinado de pontos é adicionado à camada de resultado.

Per feature and global minimum distances can be specified in order to avoid points being too close in the output point layer. If a minimum distance is specified, it may not be possible to generate the specified number of points for each feature. The total number of generated points and missed points are available as output from the algorithm.

The illustration below shows the effect of per feature and global minimum distances and zero/non-zero minimum distances (generated with the same seed, so at least the first point generated will be the same).

Figura24.84: Five points per line feature, *left*: min. distances = 0, *middle*: min.distances != 0, *right*: min. distance != 0, global min. distance = 0

O número máximo de tentativas por ponto pode ser especificado. Isso só é relevante para distância mínima diferente de zero.

A seed for the random number generator can be provided, making it possible to get identical random number sequences for different runs of the algorithm.

The attributes of the line feature on which a point was generated can be included (*Include line attributes*).

Se você deseja aproximadamente a mesma densidade de pontos para todos as feições de linha, você pode definir os dados do número de pontos usando o comprimento da geometria da feição de linha.

Ver também

Pontos aleatórios ao longo da linha

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada de linha	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Número de pontos para cada feição	`POINTS_NUMBER`	[numeric: integer] Padrão: 1	Número de pontos para criar

continua na próxima página

Tabela 24.527 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Distância mínima entre pontos (por feição) Opcional	MIN_DISTANCE	[numeric: double] Padrão: 0.0	A distância mínima entre pontos dentro de uma feição de linha
Pontos aleatórios nas linhas	OUTPUT	[vetor: ponto] Padrão: [Criar camada temporária]	The output random points. *One of*: • Criar Camada Temporária (TEMPORARY_OUTPUT) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Distância mínima global entre pontos Opcional	MIN_DISTANCE_G	[numeric: double] Padrão: 0.0	A distância mínima global entre os pontos. Deve ser menor que a *Distância mínima entre pontos (por feição)* para que esse parâmetro tenha efeito.
Maximum number of search attempts (for Min. dist. > 0) Opcional	MAX_TRIES_PER_	[numeric: integer] Padrão: 10	The maximum number of tries per point. Only relevant if the minimum distance between points is set (and greater than 0).
Random seed Opcional	SEED	[numeric: integer] Padrão: Não definido	The seed to use for the random number generator.
Incluir atributos de linha	INCLUDE_LINE_ATT	[boleano] Padrão: Verdadeiro	Se definido, um ponto obterá os atributos da linha na qual é colocado.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos aleatórios nas linhas	OUTPUT	[vetor: ponto]	The output random points layer.
Número de características com geometria vazia ou sem geometria	FEATURES_WITH_EMP	[numeric: integer]	
Número de feições com pontos faltantes	LINES_WITH_MIS	[numeric: integer]	Não incluindo feições com geometria vazia ou sem geometria
Número total de pontos gerados	POINTS_GENERAT	[numeric: integer]	

continua na próxima página

Tabela 24.529 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Número de pontos faltantes	`POINTS_MISSED`	[numeric: integer]	O número de pontos que não puderam ser gerados devido à restrição de distância mínima.

Código Python

Algorithm ID: `native:randompointsonlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Raster pixels to points

Cria uma camada vetorial de pontos correspondentes a cada pixel em uma camada raster.

Converts a raster layer to a vector layer, by creating point features for each individual pixel's center in the raster layer. Any NoData pixels are skipped in the output.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada raster	`INPUT_RASTER`	[raster]	Camada raster de entrada
Número da banda	`RASTER_BAND`	[banda raster]	Raster band to extract data from
Campo nome	`FIELD_NAME`	[string] Padrão: 'VALOR'	Name of the field to store the raster band value
Pontos vetoriais	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the resulting point layer of pixels centroids. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos vetoriais	`OUTPUT`	[vetor: ponto]	Resulting point layer with pixels centroids

Código Python

Algorithm ID: `native:pixelstopoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Raster pixels to polygons

Cria uma camada vetorial de polígonos correspondentes a cada pixel em uma camada raster.

Converts a raster layer to a vector layer, by creating polygon features for each individual pixel's extent in the raster layer. Any NoData pixels are skipped in the output.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada raster	`INPUT_RASTER`	[raster]	Camada raster de entrada
Número da banda	`RASTER_BAND`	[banda raster]	Raster band to extract data from
Campo nome	`FIELD_NAME`	[string] Padrão: 'VALOR'	Name of the field to store the raster band value
vetor de polígonos	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the resulting polygon layer of pixel extents. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
vetor de polígonos	`OUTPUT`	[vetor: polígono]	Camada poligonal resultante de extensões de píxeis

Código Python

Algorithm ID: `native:pixelstopolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pontos regulares

Cria uma nova camada de pontos com seus pontos colocados em uma grade regular dentro de uma determinada extensão.

The grid is specified either by the spacing between the points (same spacing for all dimensions) or by the number of points to generate. In the latter case, the spacing will be determined from the extent. In order to generate a full rectangular grid, at least the number of points specified by the user is generated for the latter case.

Random offsets to the point spacing can be applied, resulting in a non-regular point pattern.

Menu padrão: *Vetor ► Ferramentas de pesquisa*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Extensão de entrada (xmin, xmax, ymin, ymax)	`EXTENT`	[extensão]	Map extent for the random points Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`

continua na próxima página

Tabela 24.534 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Point spacing/count	`SPACING`	[numeric: integer] Padrão: 100	Spacing between the points, or the number of points, depending on whether `Use point spacing` is checked or not.
Initial inset from corner (LH side)	`INSET`	[numeric: double] Padrão: 0.0	Offsets the points relative to the upper left corner. The value is used for both the X and Y axis.
Apply random offset to point spacing	`RANDOMIZE`	[boleano] Padrão: Falso	Se selecionados, os pontos terão um espaçamento aleatório
Use espaçamento entre pontos	`IS_SPACING`	[boleano] Padrão: Verdadeiro	If unchecked the point spacing is not taken into account
Output layer CRS	`CRS`	[src] Default: *Project CRS*	CRS of the random points layer
Regular points	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output regular point layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Regular points	`OUTPUT`	[vetor: ponto]	A camada de ponto regular de saída.

Código Python

Algorithm ID: `qgis:regularpoints`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.25 Vector general

Atribuir projeção

Atribui uma nova projeção a uma camada vetorial.

It creates a new layer with the exact same features and geometries as the input one, but assigned to a new CRS. The geometries are **not** reprojected, they are just assigned to a different CRS.

Este algoritmo pode ser usado para reparar camadas que receberam uma projeção incorreta.

Os atributos não são modificados por este algoritmo.

Ver também

Define projection, *Encontrar projeção*, *Reprojetar camada*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Vector layer with wrong or missing CRS
Assigned CRS	`CRS`	[src] Padrão: `EPSG:4326 - WGS84`	Select the new CRS to assign to the vector layer
Assigned CRS Opcional	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Assigned CRS	`OUTPUT`	[o mesmo que entrada]	Vector layer with assigned projection

Código Python

Algorithm ID: `native:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Batch Nominatim geocoder

Performs batch geocoding using the Nominatim service against an input layer string field. The output layer will have a point geometry reflecting the geocoded location as well as a number of attributes associated to the geocoded location.

permite features in-place modification 1 de pontos da característica.

Nota

This algorithm is compliant with the usage policy of the Nominatim geocoding service provided by the OpenStreetMap Foundation.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Vector layer to geocode the features
Campo de Endereço	`FIELD`	[tablefield: string]	Campo contendo os endereços para geocode
Geocodificado	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output layer containing only the geocoded addresses. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Geocodificado	`OUTPUT`	[vetor: ponto]	Vector layer with point features corresponding to the geocoded addresses

Código Python

Algorithm ID: `native:batchnominatimgeocoder`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Convert layer to spatial bookmarks

Creates spatial bookmarks corresponding to the extent of features contained in a layer.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	A camada vetorial de entrada
Bookmark destination	`DESTINATION`	[enumeração] Padrão: 0	Selecione o destino para os favoritos. Um de: • 0 — Projetos favoritos • 1 — Usuários favoritos
Nome do Campo	`NAME_EXPRESSIO`	[expressão]	Field or expression that will give names to the generated bookmarks
Group field	`GROUP_EXPRESSI`	[expressão]	Field or expression that will provide groups for the generated bookmarks

Saídas

Rótulo	Nome	Tipo	Descrição
Count of bookmarks added	`COUNT`	[numeric: integer]	

Código Python

Algorithm ID: `native:layertobookmarks`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Convert spatial bookmarks to layer

Creates a new layer containing polygon features for stored spatial bookmarks. The export can be filtered to only bookmarks belonging to the current project, to all user bookmarks, or a combination of both.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Bookmark source	`SOURCE`	[enumeração] [lista] Padrão: [0,1]	Selecione a(s) fonte(s) dos favoritos. Um ou mais de: • 0 — Projetos favoritos • 1 — Usuários favoritos
SRC de Saída	`CRS`	[src] Padrão: `EPSG:4326 - WGS 84`	O SRC da camada de saída
Saída	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Saída	`OUTPUT`	[vetor: polígono]	The output (bookmarks) vector layer

Código Python

Algorithm ID: `native:bookmarkstolayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar índice de atributo

Cria um índice em um campo da tabela de atributos para acelerar as consultas. O suporte para criação de índice depende do provedor de dados da camada e do tipo de campo.

No outputs are created: the index is stored on the layer itself.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Selecione a camada vetorial para a qual deseja criar um índice de atributo
Attribute to index	`FIELD`	[campo da tabela: qualquer]	Campo da camada vetorial

Saídas

Rótulo	Nome	Tipo	Descrição
Camada indexada	`OUTPUT`	[o mesmo que entrada]	Uma cópia da camada de vetor de entrada com um índice para o campo especificado

Código Python

Algorithm ID: `native:createattributeindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Criar índice espacial

Cria um índice para acelerar o acesso às feições em uma camada com base em sua localização espacial. O suporte para a criação de índice espacial depende do provedor de dados da camada.

Nenhuma nova camada de saída é criada.

Default menu: *Vector ► Data Management Tools*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada

Saídas

Rótulo	Nome	Tipo	Descrição
Camada indexada	`OUTPUT`	[o mesmo que entrada]	Uma cópia da camada vetorial de entrada com um índice espacial

Código Python

Algorithm ID: `native:createspatialindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Define projection

Sets an existing layer's projection to the provided CRS without reprojecting features.

Contrary to the *Atribuir projeção* algorithm, it modifies the current layer and will not output a new layer.

Nota

For Shapefile datasets, the `.prj` and `.qpj` files will be overwritten - or created if missing - to match the provided CRS.

Default menu: *Vector ► Data Management Tools*

Ver também

Atribuir projeção, *Encontrar projeção*, *Reprojetar camada*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial com informações de projeção ausentes
SRC	`CRS`	[src]	Selecione o SRC para atribuir à camada vetorial

Saídas

Rótulo	Nome	Tipo	Descrição
	`INPUT`	[o mesmo que entrada]	A camada vetorial de entrada com a projeção definida

Código Python

Algorithm ID: `native:definecurrentprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Excluir geometrias duplicadas

Localiza e remove geometrias duplicadas.

Attributes are not checked, so in case two features have identical geometries but different attributes, only one of them will be added to the result layer.

Nota

This algorithm does not require valid geometries as input.

Ver também

Drop geometries, *Remover geometrias nulas*, *Excluir duplicatas por atributo*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	A camada com geometrias duplicadas que você deseja limpar
Limpo	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output layer containing the unique features. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Contagem de registros duplicados descartados	`DUPLICATE_COUNT`	[numeric: integer]	Contagem de registros duplicados descartados
Limpo	`OUTPUT`	[o mesmo que entrada]	A camada de saída sem geometrias duplicadas
Contagem de registros retidos	`RETAINED_COUNT`	[numeric: integer]	Contagem de registros únicos

Código Python

Algorithm ID: `native:deleteduplicategeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Excluir duplicatas por atributo

Deletes duplicate rows by only considering the specified field(s). The first matching row will be retained, and duplicates will be discarded.

Opcionalmente, esses registros duplicados podem ser salvos em uma saída separada para análise.

Ver também

Excluir geometrias duplicadas

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada de entrada
Campos para combinar duplicatas por	`FIELDS`	[campo da tabela: qualquer][list]	Fields defining duplicates. Features with identical values for all these fields are considered duplicates.
Filtrado (sem duplicatas)	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output layer containing the unique features. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Filtrado (duplicatas) Opcional	`DUPLICATES`	[o mesmo que entrada] Default: `[Pular saída]`	Specify the output layer containing only the duplicates. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Filtrado (duplicatas)	`DUPLICATES`	[o mesmo que entrada]	Vector layer containing the duplicate features.
Contagem de registros duplicados descartados	`DUPLICATE_COUNT`	[numeric: integer]	Contagem de registros duplicados descartados
Filtrado (sem duplicatas)	`OUTPUT`	[o mesmo que entrada]	Camada vetorial contendo as feições únicas.
Contagem de registros retidos	`RETAINED_COUNT`	[numeric: integer]	Contagem de registros únicos

Código Python

Algorithm ID: `native:removeduplicatesbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Detectar mudanças no conjunto de dados

Compara duas camadas vetoriais e determina quais recursos não foram alterados, adicionados ou excluídos entre os dois. Ele é projetado para comparar duas versões diferentes do mesmo conjunto de dados.

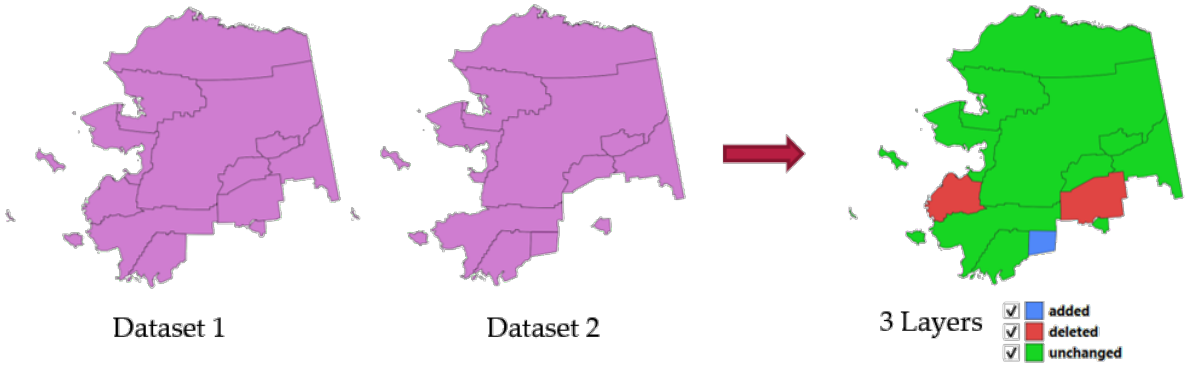


Figura24.85: Detectar exemplo de mudança de conjunto de dados

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada original	`ORIGINAL`	[vector: geometry]	A camada vetorial considerada como a versão original
Camada revisada	`REVISED`	[vector: geometry]	A camada vetorial revisada ou modificada
Atributos a considerar para a combinação Opcional	`COMPARE_ATTRIBUTE`	[campo da tabela: qualquer][list]	Atributos a serem considerados para correspondência. Por padrão, todos os atributos são comparados.
Comportamento de comparação de geometria Opcional	`MATCH_TYPE`	[enumeração] Padrão: 1	Define os critérios para comparação. Opções: • 0 — Correspondência exata: inclui a ordem e a contagem de vértices das geometrias • 1 — Combinação Tolerante (Igualdade Topológica): as geometrias são consideradas iguais

continua na próxima página

Tabela 24.554 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Feições inalteradas Opcional	UNCHANGED	[vetor: igual à Camada original]	Specify the output vector layer containing the unchanged features. *One of*: • Ignorar Saída • Criar Camada Temporária (TEMPORARY_OUTPUT) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Feições adicionadas Opcional	ADDED	[vetor: igual à Camada original]	Specify the output vector layer containing the added features. *One of*: • Ignorar Saída • Criar Camada Temporária (TEMPORARY_OUTPUT) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Feições excluídas Opcional	DELETED	[vetor: igual à Camada original]	Specify the output vector layer containing the deleted features. *One of*: • Ignorar Saída • Criar Camada Temporária (TEMPORARY_OUTPUT) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Feições inalteradas	UNCHANGED	[vetor: igual à Camada original]	Camada vetorial contendo as feições inalteradas.
Feições adicionadas	ADDED	[vetor: igual à Camada original]	Camada vetorial contendo as feições adicionadas.
Feições excluídas	DELETED	[vetor: igual à Camada original]	Camada vetorial contendo as feições excluídas.
Contagem de feições inalteradas	UNCHANGED_COUNT	[numeric: integer]	Contagem de feições inalteradas
Count of features added in revised layer	ADDED_COUNT	[numeric: integer]	Contagem de feições adicionadas na camada revisada.

continua na próxima página

Tabela 24.555 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Contagem de feições excluídas da camada original	`DELETED_COUNT`	[numeric: integer]	Contagem de feições excluídas da camada original.

Código Python

Algorithm ID: `native:detectvectorchanges`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Drop geometries

Creates a simple *geometryless* copy of the input layer attribute table. It keeps the attribute table of the source layer.

If the file is saved in a local folder, you can choose between many file formats.

Allows features in-place modification 1 de ponto, linha e características poligonais.

Ver também

Excluir geometrias duplicadas, *Remover geometrias nulas*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	A camada vetorial de entrada
Dropped geometries	`OUTPUT`	[vector: table]	Specify the output geometryless layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Dropped geometries	`OUTPUT`	[vector: table]	The output geometryless layer. A copy of the original attribute table.

Código Python

Algorithm ID: `native:dropgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Executar SQL

Runs a simple or complex query based only on SELECT with `SQL` syntax on the source layer.

Input datasources are identified with `input1`, `input2`... `inputN` and a simple query will look like `SELECT * FROM input1`.

Beside a simple query, you can add expressions or variables within the `SQL query` parameter itself. This is particulary useful if this algorithm is executed within a Processing model and you want to use a model input as a parameter of the query. An example of a query will then be `SELECT * FROM [% @table %]` where `@table` is the variable that identifies the model input.

O resultado da consulta será adicionado como uma nova camada.

Ver também

SpatiaLite execute SQL, *PostgreSQL executa SQL*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Additional input datasources (called input1, .., inputN in the query)	`INPUT_DATASOURCE`	[vetor: qualquer] [lista]	List of layers to query. In the SQL editor you can refer these layers with their **real** name or also with **input1**, **input2**, **inputN** depending on how many layers have been chosen.
Consulta SQL	`INPUT_QUERY`	[string]	Type the string of your SQL query, e.g. `SELECT * FROM input1`.
Unique identifier field Opcional	`INPUT_UID_FIELD`	[string]	Specify the column with unique ID
Campo geometria Opcional	`INPUT_GEOMETRY_F`	[string]	Especifique o campo de geometria

continua na próxima página

Tabela 24.558 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Tipo de Geometria Opcional	`INPUT_GEOMETRY_T`	[enumeração] Padrão: 0	Escolher a geometria do resultado. Por padrão, o algoritmo irá detectar automaticamente. Um de: • 0 — Detecção automática • 1 — Sem geometria • 2 — Ponto • 3 — Cadeiadelinha • 4 — Polígono • 5 — Multiponto • 6 — MultiLineString • 7 — MultiPolígono
SRC Opcional	`INPUT_GEOMETRY_C`	[src]	O SRC a ser atribuído à camada de saída
SQL Output	`OUTPUT`	[vector: geometry] Padrão: `[Criar camada temporária]`	Specify the output layer created by the query. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
SQL Output	`OUTPUT`	[vector: geometry]	Camada vetorial criada pela consulta

Código Python

Algorithm ID: `qgis:executesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exportar camadas para DXF

Exports layers to DXF file. For each layer, you can choose a field whose values are used to split features in generated destination layers in DXF output.

Ver também

Criando novos arquivos DXF

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camadas de entrada*	`LAYERS`	[vector: geometry] [list]	List of input vector layers with options associated (filled as a `QgsProcessingParameterDxfLayers` item — done in GUI by pressing *Configure Layer…* button for each selected layer): **Layer [string] (`layer`)** Full path of the input layer to export **Output layer attribute [tablefield: any] (`attributeIndex`)** Attribute index to split the input layer using unique values **Output layer name [string] (`overriddenLayerName`)** Overridden layer name to be used in the exported DXF file (the original project layer remains untouched) **Allow data defined symbol blocks** [boolean] (`buildDataDefinedBlocks`) **Maximum number of symbol blocks [numeric: integer] (`dataDefinedBlocksMaximumNumberOfClasses`)** −1 means no limitation.
Modo de simbologia	`SYMBOLOGY_MODE`	[enumeração] Padrão: 0	Tipo de simbologia a ser aplicada às camadas de saída. Você pode escolher entre: • 0 — Sem Simbologia • 1 — Simbologia das feições • 2 — Simbologia da Camada de Símbolo
Escala de simbologia	`SYMBOLOGY_SCALE`	[escala] Padrão: 1:1 000 000	Default scale of data export.
Map theme Opcional	`MAP_THEME`	[map theme]	Match layer styling to the provided map theme.
Codificando	`ENCODING`	[enumeração]	Encoding to apply to layers.
SRC	`CRS`	[src]	Escolha o SRC para a camada de saída.
Extensão Opcional	`EXTENT`	[extensão]	Limit exported features to those with geometries intersecting the provided extent.

continua na próxima página

Tabela 24.560 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Utilizar o título da camada como nome	`USE_LAYER_TITL`	[boleano] Padrão: Falso	Name the output layer with the layer title (as set in layer metadata or QGIS Server properties) instead of the layer name.
Force 2D	`FORCE_2D`	[boleano] Padrão: Falso	
Export labels as MTEXT elements	`MTEXT`	[boleano] Padrão: Verdadeiro	Exports labels as MTEXT or TEXT elements
Use only selected features	`SELECTED_FEATURE`	[boleano] Padrão: Falso	Exports only the selected features.
DXF	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Specification of the output DXF file. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
DXF	`OUTPUT`	[arquivo]	`.DXF` file containing the input layers

Código Python

Algorithm ID: `native:dxfexport`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair feições selecionadas

Salvando feições selecionadas como uma nova camada.

> **Nota**
>
> Se a camada selecionada não tiver feições selecionadas, a camada recém-criada estará vazia.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada para salvar a seleção de
Feições selecionadas	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the vector layer for the selected features. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Feições selecionadas	`OUTPUT`	[o mesmo que entrada]	Camada vetorial com apenas as feições selecionadas, ou nenhuma feição se nenhuma foi selecionada.

Código Python

Algorithm ID: `native:saveselectedfeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair a codificação do Shapefile

Extracts the attribute encoding information embedded in a Shapefile. Both the encoding specified by an optional `.cpg` file and any encoding details present in the `.dbf` LDID header block are considered.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	ESRI Shapefile (**:arquivo:`.SHP`**) Camada para extrair a informação codificada.

Saídas

Rótulo	Nome	Tipo	Descrição
Codificação do Shapefile	`ENCODING`	[string]	Informações de codificação especificadas no arquivo de entrada
CPG encoding	`CPG_ENCODING`	[string]	Encoding information specified in any optional `.CPG` file
LDID encoding	`LDID_ENCODING`	[string]	Encoding information specified in `.dbf` LDID header block

Código Python

Algorithm ID: `native:shpencodinginfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Encontrar projeção

Creates a shortlist of candidate coordinate reference systems, for instance for a layer with an unknown projection.

A área que a camada deve cobrir deve ser especificada por meio do parâmetro de área de destino. O sistema de referência de coordenadas para esta área alvo deve ser conhecido pelo QGIS.

The algorithm operates by testing the layer's extent in every known reference system and then listing any for which the bounds would be near the target area if the layer was in this projection.

Ver também

Atribuir projeção, *Define projection*, *Reprojetar camada*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada com projeção desconhecida
Target area for layer (xmin, xmax, ymin, ymax)	`TARGET_AREA`	[extensão]	A área que a camada cobre. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Inserir as coordenadas como `xmin, xmax, ymin, ymax`
CRS candidates	`OUTPUT`	[vector: table] Padrão: `[Criar camada temporária]`	Specify the table (geometryless layer) for the CRS suggestions (EPSG codes). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
CRS candidates	`OUTPUT`	[vector: table]	A table with all the CRS (EPSG codes) of the matching criteria.

Código Python

Algorithm ID: `native:findprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Flatten relationship

Flattens a *relationship* for a vector layer, exporting a single layer containing one parent feature per related child feature. This master feature contains all the attributes for the related features. This allows to have the relation as a plain table that can be e.g. exported to CSV.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

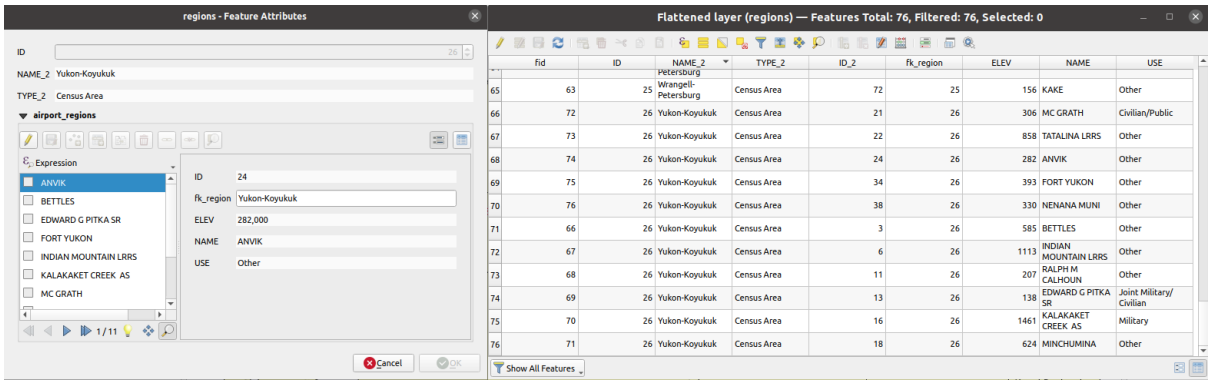


Figura24.86: Form of a region with related children (left) - A duplicate region feature for each related child, with joined attributes (right)

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Layer with the relationship that should be de-normalized
Flattened Layer Opcional	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output (flattened) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Camada achatada	`OUTPUT`	[o mesmo que entrada]	Uma camada contendo feições mestre com todos os atributos para as feições relacionadas

Código Python

Algorithm ID: `native:flattenrelationships`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Unir atributos por valor de campo

Takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. An attribute is selected in each of them to define the join criteria.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Unir atributos por mais próximo, *Unir atributos por localização*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Input vector layer. The output layer will consist of the features of this layer with attributes from matching features in the second layer.
Table field	`FIELD`	[campo da tabela: qualquer]	Campo da camada de origem a ser usado para a união
Camada de entrada 2	`INPUT_2`	[vetor: qualquer]	Camada com a tabela de atributos para unir
Table field 2	`FIELD_2`	[campo da tabela: qualquer]	Field of the second (join) layer to use for the join The type of the field must be equal to (or compatible with) the input table field type.
Campos da camada 2 para copiar Opcional	`FIELDS_TO_COPY`	[campo da tabela: qualquer][list]	Selecione os campos específicos que você deseja adicionar. Por padrão, todos os campos são adicionados.

continua na próxima página

Tabela 24.570 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Join type	`METHOD`	[enumeração] Padrão: 1	O tipo da camada final unida. Um de: • 0 — Create separate feature for each matching feature (one-to-many) • 1 — Take attributes of the first matching feature only (one-to-one)
Discard records which could not be joined	`DISCARD_NONMATCHI`	[boleano] Padrão: Verdadeiro	Check if you don't want to keep the features that could not be joined
Prefixo de campo combinado Opcional	`PREFIX`	[string]	Add a prefix to joined fields in order to easily identify them and avoid field name collision
Joined layer Opcional	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the join. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Unjoinable features from first layer Opcional	`NON_MATCHING`	[o mesmo que entrada] Default: `[Pular saída]`	Specify the output vector layer for unjoinable features from first layer. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Number of joined features from input table	`JOINED_COUNT`	[numeric: integer]	
Unjoinable features from first layer Opcional	`NON_MATCHING`	[o mesmo que entrada]	Camada vetorial com as características não correspondentes
Joined layer Opcional	`OUTPUT`	[o mesmo que entrada]	Camada vetorial de saída com atributos adicionais da junção
Number of unjoinable features from input table Opcional	`UNJOINABLE_COUNT`	[numeric: integer]	

Código Python

Algorithm ID: `native:joinattributestable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Unir atributos por localização

Takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. A spatial criteria is applied to select the values from the second layer that are added to each feature from the first layer.

Default menu: *Vector ► Data Management Tools*

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Unir atributos por mais próximo, *Unir atributos por valor de campo*, *Join attributes by location (summary)*

Explorando relações espaciais

Predicados geométricos são funções booleanas usadas para determinar a relação espacial que um recurso tem com outro, comparando se e como suas geometrias compartilham uma parte do espaço.

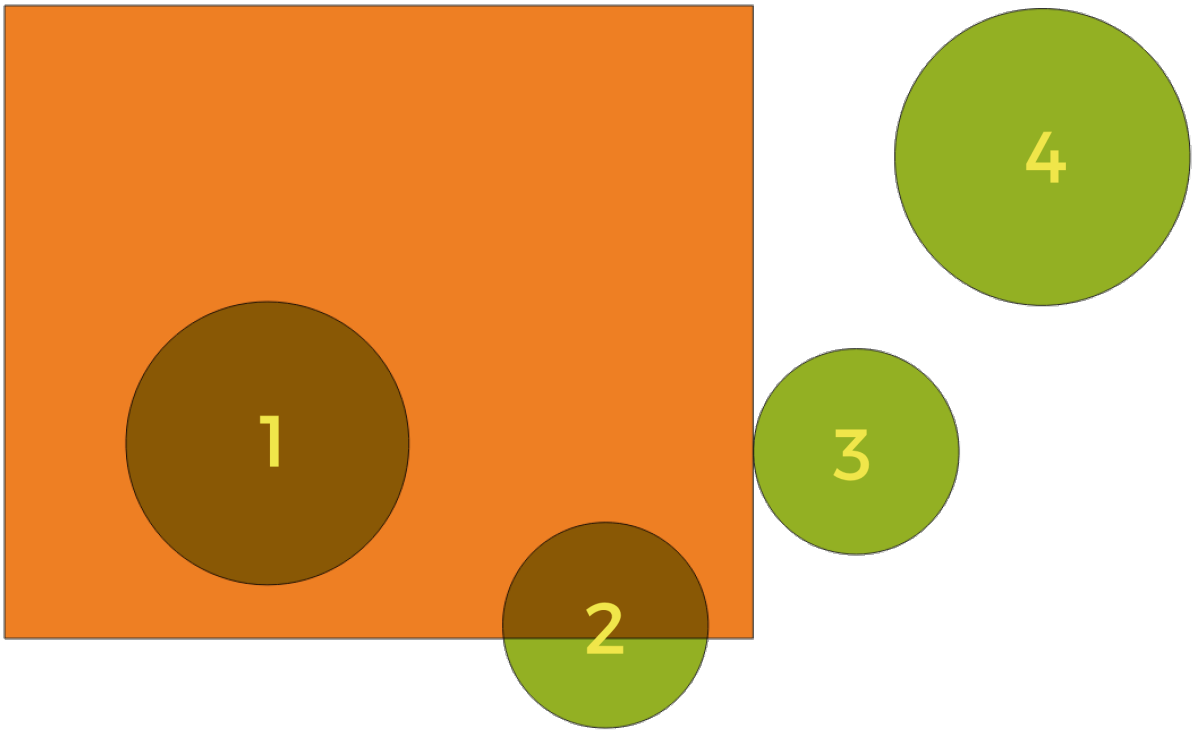


Figura24.87: Procurando relações espaciais entre camadas

Usando a figura acima, estamos procurando os círculos verdes comparando-os espacialmente com o retângulo laranja. Os predicados geométricos disponíveis são:

Intersecção
Testa se uma geometria intercepta outra. Retorna 1 (verdadeiro) se as geometrias se cruzarem espacialmente (compartilhar qualquer parte do espaço - sobreposição ou toque) e 0 se não houver. Na imagem acima, isso retornará os círculos 1, 2 e 3.

Contém
Retorna 1 (verdadeiro) se e somente se nenhum ponto de b estiver no exterior de a, e pelo menos um ponto do interior de b estiver no interior de a. Na figura, nenhum círculo é retornado, mas o retângulo seria se você o procurasse ao contrário, porque contém o círculo 1 completamente. Este é o oposto de *estão dentro*.

Disjoint
Retorna 1 (verdadeiro) se as geometrias não compartilham nenhuma parte do espaço (sem sobreposição, sem toque). Apenas o círculo 4 é retornado.

Igual
Retorna 1 (verdadeiro) se e somente se as geometrias forem exatamente iguais. Nenhum círculo será retornado.

Toca
Testa se uma geometria toca outra. Retorna 1 (verdadeiro) se as geometrias tiverem pelo menos um ponto em comum, mas seus interiores não se cruzarem. Apenas o círculo 3 é retornado.

Sobrepor
Testa se uma geometria se sobrepõe a outra. Retorna 1 (verdadeiro) se as geometrias compartilham espaço, são da mesma dimensão, mas não estão completamente contidas umas nas outras. Apenas o círculo 2 é retornado.

Estão dentro
Testa se uma geometria está dentro de outra. Retorna 1 (verdadeiro) se a geometria a estiver completamente dentro da geometria b. Apenas o círculo 1 é retornado.

Cruza
Retorna 1 (verdadeiro) se as geometrias fornecidas tiverem alguns, mas não todos, pontos internos em comum e o cruzamento real for de uma dimensão menor que a geometria fornecida mais alta. Por exemplo, uma linha

que cruza um polígono cruzará como uma linha (verdadeiro). Duas linhas que se cruzam se cruzarão como um ponto (verdadeiro). Dois polígonos se cruzam como um polígono (falso). Na imagem, nenhum círculo será devolvido.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Join to features in	`INPUT`	[vector: geometry]	Input vector layer. The output layer will consist of the features of this layer with attributes from matching features in the second layer.
Where the features	`PREDICATE`	[enumeração] [lista] Padrão: [0]	Type of spatial relation the source feature should have with the target feature so that they could be joined. One or more of: • 0 — Interseção • 1 — contém • 2 — equal • 3 — touch • 4 — overlap • 5 — are within • 6 — cross Se mais de uma condição for escolhida, pelo menos uma delas (operação OU) tem que ser atendida para que uma característica seja extraída.
By comparing to	`JOIN`	[vector: geometry]	The join layer. Features of this vector layer will **add** their attributes to the source layer attribute table if they satisfy the spatial relationship.
Campos a serem adicionados (deixe em branco para usar todos os campos) Opcional	`JOIN_FIELDS`	[campo da tabela: qualquer][list]	Select the specific fields you want to add from the join layer. By default all the fields are added.
Join type	`METHOD`	[enumeração]	O tipo da camada final unida. Um de: • 0 — Create separate feature for each matching feature (one-to-many) • 1 — Take attributes of the first matching feature only (one-to-one) • 2 — Take attributes of the feature with largest overlap only (one-to-one)
Discard records which could not be joined	`DISCARD_NONMATCHI`	[boleano] Padrão: Falso	Remove from the output the input layer's features which could not be joined
Prefixo de campo combinado Opcional	`PREFIX`	[string]	Add a prefix to joined fields in order to easily identify them and avoid field name collision

continua na próxima página

Tabela 24.572 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Joined layer Opcional	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the join. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Unjoinable features from first layer Opcional	`NON_MATCHING`	[o mesmo que entrada] Default: `[Pular saída]`	Specify the output vector layer for unjoinable features from first layer. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Number of joined features from input table	`JOINED_COUNT`	[numeric: integer]	
Unjoinable features from first layer Opcional	`NON_MATCHING`	[o mesmo que entrada]	Vector layer of the non-matched features
Joined layer	`OUTPUT`	[o mesmo que entrada]	Camada vetorial de saída com atributos adicionais da junção

Código Python

Algorithm ID: `native:joinattributesbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Join attributes by location (summary)

Takes an input vector layer and creates a new vector layer that is an extended version of the input one, with additional attributes in its attribute table.

The additional attributes and their values are taken from a second vector layer. A spatial criteria is applied to select the values from the second layer that are added to each feature from the first layer.

The algorithm calculates a statistical summary for the values from matching features in the second layer (e.g. maximum value, mean value, etc).

Ver também

Unir atributos por localização

Explorando relações espaciais

Predicados geométricos são funções booleanas usadas para determinar a relação espacial que um recurso tem com outro, comparando se e como suas geometrias compartilham uma parte do espaço.

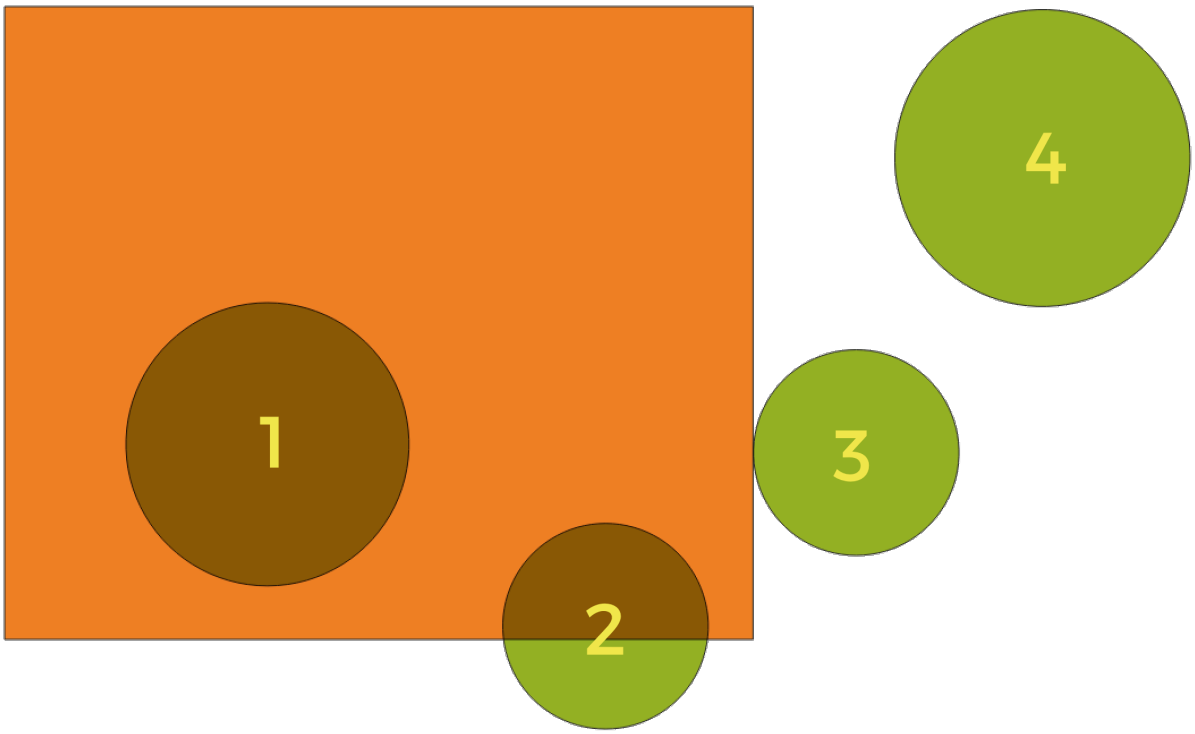


Figura24.88: Procurando relações espaciais entre camadas

Usando a figura acima, estamos procurando os círculos verdes comparando-os espacialmente com o retângulo laranja. Os predicados geométricos disponíveis são:

Intersecção
: Testa se uma geometria intercepta outra. Retorna 1 (verdadeiro) se as geometrias se cruzarem espacialmente (compartilhar qualquer parte do espaço - sobreposição ou toque) e 0 se não houver. Na imagem acima, isso retornará os círculos 1, 2 e 3.

Contém
: Retorna 1 (verdadeiro) se e somente se nenhum ponto de b estiver no exterior de a, e pelo menos um ponto do

interior de b estiver no interior de a. Na figura, nenhum círculo é retornado, mas o retângulo seria se você o procurasse ao contrário, porque contém o círculo 1 completamente. Este é o oposto de *estão dentro*.

Disjoint
Retorna 1 (verdadeiro) se as geometrias não compartilham nenhuma parte do espaço (sem sobreposição, sem toque). Apenas o círculo 4 é retornado.

Igual
Retorna 1 (verdadeiro) se e somente se as geometrias forem exatamente iguais. Nenhum círculo será retornado.

Toca
Testa se uma geometria toca outra. Retorna 1 (verdadeiro) se as geometrias tiverem pelo menos um ponto em comum, mas seus interiores não se cruzarem. Apenas o círculo 3 é retornado.

Sobrepor
Testa se uma geometria se sobrepõe a outra. Retorna 1 (verdadeiro) se as geometrias compartilham espaço, são da mesma dimensão, mas não estão completamente contidas umas nas outras. Apenas o círculo 2 é retornado.

Estão dentro
Testa se uma geometria está dentro de outra. Retorna 1 (verdadeiro) se a geometria a estiver completamente dentro da geometria b. Apenas o círculo 1 é retornado.

Cruza
Retorna 1 (verdadeiro) se as geometrias fornecidas tiverem alguns, mas não todos, pontos internos em comum e o cruzamento real for de uma dimensão menor que a geometria fornecida mais alta. Por exemplo, uma linha que cruza um polígono cruzará como uma linha (verdadeiro). Duas linhas que se cruzam se cruzarão como um ponto (verdadeiro). Dois polígonos se cruzam como um polígono (falso). Na imagem, nenhum círculo será devolvido.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Join to features in	`INPUT`	[vector: geometry]	Input vector layer. The output layer will consist of the features of this layer with attributes from matching features in the second layer.
Where the features	`PREDICATE`	[enumeração] [lista] Padrão: [0]	Type of spatial relation the source feature should have with the target feature so that they could be joined. One or more of: • 0 — Interseção • 1 — contém • 2 — equal • 3 — touch • 4 — overlap • 5 — are within • 6 — cross Se mais de uma condição for escolhida, pelo menos uma delas (operação OU) tem que ser atendida para que uma característica seja extraída.
By comparing to	`JOIN`	[vector: geometry]	The join layer. Features of this vector layer will **add** summaries of their attributes to the source layer attribute table if they satisfy the spatial relationship.

continua na próxima página

Tabela 24.574 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Campos para resumir (deixe em branco para usar todos os campos) Opcional	`JOIN_FIELDS`	[campo da tabela: qualquer][list]	Select the specific fields you want to add from the join layer. By default all the fields are added.
Resumos para calcular (deixe em branco para usar todos os campos) Opcional	`SUMMARIES`	[enumeração] [lista] Padrão: []	For each input feature, statistics are calculated on joined fields of their matching features. One or more of: • 0 — contagem • 1 — único • 2 — min • 3 — max • 4 — intervalo • 5 — soma • 6 — média • 7 — mediana • 8 — stddev • 9 — minoria • 10 — maioria • 11 — q1 • 12 — q3 • 13 — iqr • 14 — vazio • 15 — filled • 16 — comprimento_mínimo • 17 — comprimento_máximo • 17 — comprimento_médio
Discard records which could not be joined	`DISCARD_NONMATCHI`	[boleano] Padrão: Falso	Remove from the output the input layer's features which could not be joined
Joined layer	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the join. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Joined layer	`OUTPUT`	[o mesmo que entrada]	Output vector layer with summarized attributes from the join

Código Python

Algorithm ID: `native:joinbylocationsummary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Unir atributos por mais próximo

Takes an input vector layer and creates a new vector layer with additional fields in its attribute table. The additional attributes and their values are taken from a second vector layer. Features are joined by finding the closest features from each layer.

By default only the nearest feature is joined, but the join can also join to the k-nearest neighboring features.

Se uma distância máxima for especificada, somente as feições que estiverem mais próximas que essa distância serão correspondidas.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Análise do vizinho mais próximo, *Unir atributos por valor de campo*, *Unir atributos por localização*, *Matriz de distância*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	A camada de entrada.
Camada de entrada 2	`INPUT_2`	[vector: geometry]	A camada de junção.
Layer 2 fields to copy (leave empty to copy all fields)	`FIELDS_TO_COPY`	[campos]	Unir os campos da camada para copiar (se estiver vazio, todos os campos serão copiados).
Discard records which could not be joined	`DISCARD_NONMATCHI`	[boleano] Padrão: Falso	Remover da saída os registros da camada de entrada que não puderam ser unidos
Prefixo de campo combinado	`PREFIX`	[string]	Joined field prefix
Máximo de vizinhos mais próximos	`NEIGHBORS`	[numeric: integer] Padrão: 1	Número máximo de vizinhos mais próximos
Distância máxima	`MAX_DISTANCE`	[numeric: double]	Distância máxima de pesquisa

continua na próxima página

Tabela 24.576 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Joined layer Opcional	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the vector layer containing the joined features. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Unjoinable features from first layer	`NON_MATCHING`	[o mesmo que entrada] Default: `[Pular saída]`	Specify the vector layer containing the features that could not be joined. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Joined layer	`OUTPUT`	[o mesmo que entrada]	The output joined layer.
Unjoinable features from first layer	`NON_MATCHING`	[o mesmo que entrada]	Camada contendo as feições da primeira camada que não puderam ser unidas a nenhuma das feições da camada de união.
Number of joined features from input table	`JOINED_COUNT`	[numeric: integer]	Número de feições da tabela de entrada que foram unidos.
Number of unjoinable features from input table	`UNJOINABLE_COUNT`	[numeric: integer]	Número de feições da tabela de entrada que não puderam ser unidas.

Código Python

Algorithm ID: `native:joinbynearest`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

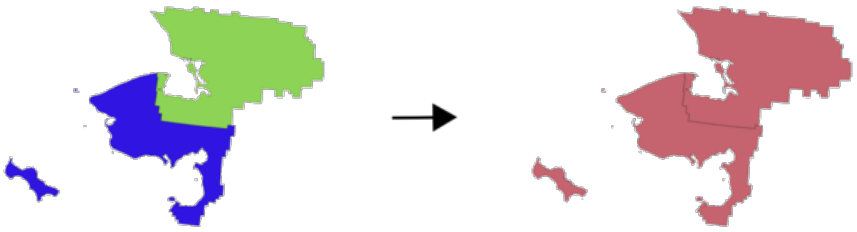
Mesclar camadas vetoriais

Combina várias camadas vetoriais do tipo **mesma geometria** em uma única.

The attribute table of the resulting layer will contain the fields from all input layers. If fields with the same name but different types are found then the exported field will be automatically converted into a string type field. Optionally, new fields storing the original layer name and source can be added.

Se alguma camada de entrada contiver valores Z ou M, a camada de saída também conterá esses valores. Da mesma forma, se qualquer uma das camadas de entrada for de várias partes, a camada de saída também será uma camada de várias partes.

Opcionalmente, o sistema de referência de coordenadas de destino (SRC) para a camada mesclada pode ser definido. Se não estiver definido, o SRC será obtido da primeira camada de entrada. Todas as camadas serão reprojetadas para corresponder a este SRC.



Default menu: *Vector ► Data Management Tools*

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Dividir camada vetorial

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camadas de Entrada	`LAYERS`	[vetor: qualquer] [lista]	As camadas que devem ser mescladas em uma única camada. As camadas devem ser do mesmo tipo de geometria.
SRC de Destino Opcional	`CRS`	[src]	Escolha o SRC para a camada de saída. Se não for especificado, o SRC da primeira camada de entrada é usado.
Add source layer information (layer name and path)	`ADD_SOURCE_FIE`	[boleano] Padrão: Verdadeiro	Add fields storing the original layer name and path

continua na próxima página

Tabela 24.578 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Mesclado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Mesclado	`OUTPUT`	[o mesmo que entrada]	Camada vetorial de saída contendo todas as feições e atributos das camadas de entrada.

Código Python

Algorithm ID: `native:mergevectorlayers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Ordenar por expressão

Classifica uma camada vetorial de acordo com uma expressão: altera o índice de feição de acordo com uma expressão.

Be careful, it might not work as expected with some providers, the order might not be kept every time.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Input vector layer to sort
Expressão	`EXPRESSION`	[expressão]	Expressão a ser usada para a classificação
Ordernar ascendente	`ASCENDING`	[boleano] Padrão: Verdadeiro	Se marcada, a camada vetorial será classificada de valores pequenos a grandes.

continua na próxima página

Tabela 24.580 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Ordenar nulos primeiro	`NULLS_FIRST`	[boleano] Padrão: Falso	Se marcado, os valores nulos são colocados primeiro
Encomendado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Encomendado	`OUTPUT`	[o mesmo que entrada]	Camada vetorial de saída (classificada)

Código Python

Algorithm ID: `native:orderbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Reparar Shapefile

Repara um conjunto de dados ESRI Shapefile quebrado (re)criando o arquivo SHX.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Inserir arquivo de forma	`INPUT`	[arquivo]	Caminho completo para o conjunto de dados ESRI Shapefile com um arquivo SHX ausente ou quebrado

Saídas

Rótulo	Nome	Tipo	Descrição
Camada reparada	`OUTPUT`	[vector: geometry]	A camada vetorial de entrada com o arquivo SHX reparado

Código Python

Algorithm ID: `native:repairshapefile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Reprojetar camada

Reprojeta uma camada vetorial em um SRC diferente. A camada reprojetada terá as mesmas feições e atributos da camada de entrada.

Allows features in-place modification 1 de ponto, linha e características poligonais.

Ver também

Atribuir projeção, *Define projection*, *Encontrar projeção*

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada para reprojetar
SRC Alvo	`TARGET_CRS`	[src] Padrão: `EPSG:4326 - WGS 84`	Destination coordinate reference system
Convert curved geometries to straight segments	`CONVERT_CURVED_GE`	[boleano] Padrão: Falso	If checked, curved geometries will be converted to straight segments in the process, avoiding potential distortion issues.

continua na próxima página

Tabela 24.584 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Reprojetado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Coordinate Operation Opcional	`OPERATION`	[string]	Specific operation to use for a particular reprojection task, instead of always forcing use of the current project's transformation settings. Useful when reprojecting a particular layer and control over the exact transformation pipeline is required. Requires proj version >= 6. Read more at *Transformações de Datum*.

Saídas

Rótulo	Nome	Tipo	Descrição
Reprojetado	`OUTPUT`	[o mesmo que entrada]	Camada vetorial de saída (reprojetada)

Código Python

Algorithm ID: `native:reprojectlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Salvar feições vetoriais em arquivo

Saves vector features to a specified file dataset.

For dataset formats supporting layers, an optional layer name parameter can be used to specify a custom string. Optional GDAL-defined dataset and layer options can be specified. For more information on this, read the online GDAL documentation on the format.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Feições de vetor	`INPUT`	[vetor: qualquer]	Input vector layer.
Feições salvas	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specify the file to save the features to. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Nome da camada Opcional	`LAYER_NAME`	[string]	Name to use for the output layer
GDAL dataset options Opcional	`DATASOURCE_OPTIONS`	[string]	GDAL dataset creation options of the output format. Separate individual options with semicolons.
GDAL layer options Opcional	`LAYER_OPTIONS`	[string]	GDAL layer creation options of the output format. Separate individual options with semicolons.
Action to take on pre-existing file	`ACTION_ON_EXISTI`	[enumeração] Padrão: 0	How to manage existing features. Valid methods are: • 0 — Create or overwrite file • 1 — Create or overwrite layer • 2 — Append features to existing layer, but do not create new fields • 3 — Append features to existing layer, and create new fields if needed

Saídas

Rótulo	Nome	Tipo	Descrição
Feições salvas	`OUTPUT`	[o mesmo que entrada]	Camada vetorial com as feições salvas.
Nome e caminho do arquivo	`FILE_PATH`	[string]	Nome e caminho do arquivo de saída.
Nome da camada	`LAYER_NAME`	[string]	Nome da camada, se houver.

Código Python

Algorithm ID: `native:savefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Definir codificação de camada

Sets the encoding used for reading a layer's attributes. No permanent changes are made to the layer, rather it affects only how the layer is read during the current session.

> **Nota**
>
> Changing the encoding is only supported for some vector layer data sources.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Feições salvas	`INPUT`	[vector: geometry]	Camada vetorial para definir a codificação.
Codificando	`ENCODING`	[string]	Text encoding to assign to the layer in the current QGIS session.

Saídas

Rótulo	Nome	Tipo	Descrição
Camada de saída	`OUTPUT`	[o mesmo que entrada]	Camada vetorial de entrada com a codificação definida.

Código Python

Algorithm ID: `native:setlayerencoding`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Split features by character

Features are split into multiple output features by splitting a field's value at a specified character. For instance, if a layer contains features with multiple comma separated values contained in a single field, this algorithm can be used to split these values up across multiple output features. Geometries and other attributes remain unchanged in the output. Optionally, the separator string can be a regular expression for added flexibility.

Allows features in-place modification 1 de ponto, linha e características poligonais.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
Dividir utilizando valores do campo	`FIELD`	[campo da tabela: qualquer]	Campo a ser utilizado para divisão
Dividir valor usando caractere	`CHAR`	[string]	Caráter a ser utilizado para divisão
Utilizar o separador de expressão regular	`REGEX`	[boleano] Padrão: Falso	
Dividir	`OUTPUT`	[o mesmo que entrada] Padrão: `Criar camada temporária`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Dividir	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída.

Código Python

Algorithm ID: `native:splitfeaturesbycharacter`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Dividir camada vetorial

Creates a set of vectors in an output folder based on an input layer and an attribute. The output folder will contain as many layers as the unique values found in the desired field.

The number of files generated is equal to the number of different values found for the specified attribute.

It is the opposite operation of *merging*.

Default menu: *Vector ► Data Management Tools*

Ver também

Mesclar camadas vetoriais

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
*Campo de identificação único**	`FIELD`	[campo da tabela: qualquer]	Campo a ser utilizado para divisão
Diretório de saída	`OUTPUT`	[pasta] Padrão: `[Salvar em pasta temporária]`	Specify the directory for the output layers. *One of*: • Salvar como Diretório Temporário • Salvar no Diretório

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Tipo de arquivo de saída Opcional	`FILE_TYPE`	[enumeração] Padrão: `gpkg` na janela de diálogo	Selecione a extensão dos arquivos de saída. Se não especificado ou inválido, o formato dos arquivos de saída será o definido na configuração de processamento "Extensão padrão da camada vetorial de saída".

Saídas

Rótulo	Nome	Tipo	Descrição
Diretório de saída	`OUTPUT`	[pasta]	O diretório para as camadas de saída
Camadas de saída	`OUTPUT_LAYERS`	[o mesmo que entrada][lista]	As camadas vetoriais de saída resultantes da divisão.

Código Python

Algorithm ID: `native:splitvectorlayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Truncate table

Truncates a layer, by deleting all features from within the layer.

> **Aviso**
>
> This algorithm modifies the layer in place, and deleted features cannot be restored!

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada

Saídas

Rótulo	Nome	Tipo	Descrição
Truncated layer	`OUTPUT`	[pasta]	The input layer, all features deleted

Código Python

Algorithm ID: `native:truncatetable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.26 Geometria vetorial

Adicionar atributos da geometria

Computa propriedades geométricas das características de uma camada de vetorial e os incluem na saída de camada.

Gera uma nova camada de vetor com o mesmo conteúdo da entrada, mas com tributos adicionais, contendo medidas geométricas baseadas em um CRS selecionado.

Os atributos adicionados à tabela dependem do tipo de geometria e dimensão da camada de entrada:

- para camadas **ponto**: coordenadas X (`coordx`), Y (`coordy`), Z (`coordz`) e/ou valor M (`valorm`)
- Das camadas **line**: `length` e, da LineString e CompoundCurve tipos geométricos, o atributo "sinuosity" e straight distance (`straightdis`)
- para camadas **polígono**: `perímetro` e `área`

> **Nota**
>
> This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Default menu: *Vector ► Geometry Tools*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada

continua na próxima página

Tabela 24.599 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Calcular usando	`CALC_METHOD`	[enumeração] Padrão: 0	Parâmetros de cálculo a utilizar para as propriedades geométricas. Um de: • 0 — Camada SRC • 1 — Projeto SRC • 2 — Elipsoidal
Informações geográficas adicionadas	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output (input copy with geometry) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Informações geográficas adicionadas	`OUTPUT`	[o mesmo que entrada]	Cópia da camada vetorial de entrada com a adição dos campos de geometria

Código Python

Algorithm ID: `native:exportaddgeometrycolumns`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Transformação afim.

Aplique uma transformação afim para a camada geometrias. Transformações afim podem incluir tradução, escala e rotação. As operações são performadas na seguinte ordem: escala, rotação e tradução.

Os valores Z e M (se presentes) podem ser convertidos e dimensionados.

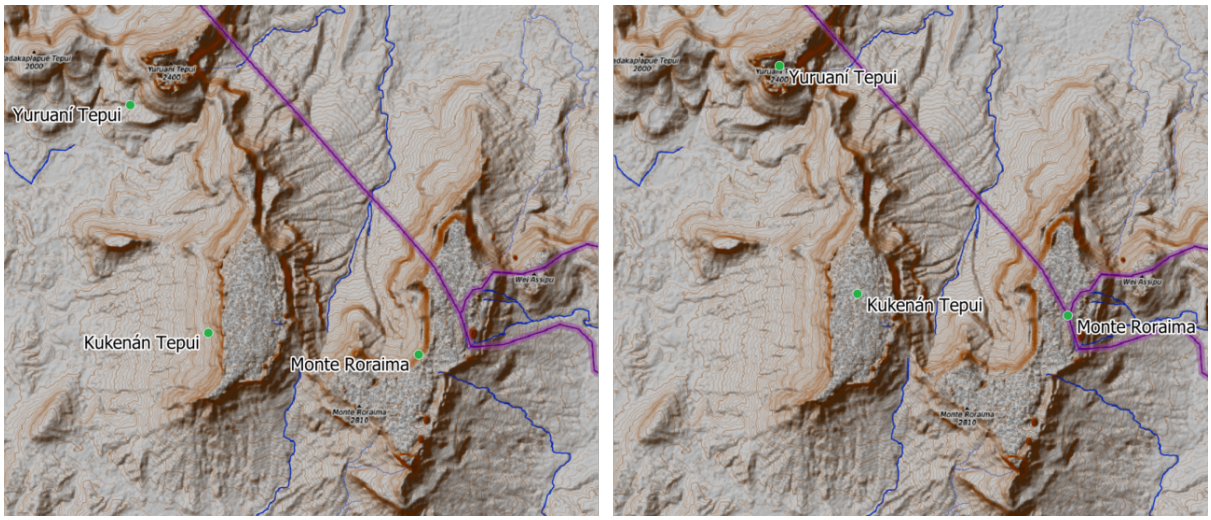


Figura24.89: Ponto da camada de vetor (pontos verdes) antes (esquerda) e depois (direita) de uma transformação afim.

Allows features in-place modification 1 de ponto, linha e características poligonais.

Ver também

Transladar

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Translation (x-axis)	`DELTA_X`	[numeric: double] Padrão: 0.0	Delocamento a aplicar ao eixo X
Translation (y-axis)	`DELTA_Y`	[numeric: double] Padrão: 0.0	Delocamento a aplicar ao eixo Y
Translation (z-axis)	`DELTA_Z`	[numeric: double] Padrão: 0.0	Delocamento a aplicar ao eixo Z
Translation (m-values)	`DELTA_M`	[numeric: double] Padrão: 0.0	Deslocamento para ser aplicado em m valores.

continua na próxima página

Tabela 24.601 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Scale factor (x-axis)	`SCALE_X`	[numeric: double] Padrão: 1,0	Valor de escala (expansão ou contração) a aplicar ao eixo X.
Scale factor (y-axis)	`SCALE_Y`	[numeric: double] Padrão: 1,0	Valor de escala (expansão ou contração) a aplicar ao eixo Y.
Scale factor (z-axis)	`SCALE_Z`	[numeric: double] Padrão: 1,0	Valor de escala (expansão ou contração) a aplicar ao eixo Z.
Scale factor (m-values)	`SCALE_M`	[numeric: double] Padrão: 1,0	Valor de escala (expansão ou contração) a aplicar em valores m.
Rotação em torno do eixo z (graus no sentido anti-horário)	`ROTATION_Z`	[numeric: double] Padrão: 0.0	Ângulo de rotação em graus.
Transformado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Transformado	`OUTPUT`	[o mesmo que entrada]	Saída(transformada) de camada vetorial.

Código Python

Algorithm ID: ``native:affinetransform`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Agregar

Pega um vetorial ou uma tabela de camada e cria uma nova camada agregando características baseada na expressão `group by`.

Características para as quais a expressão *group by*` retorna o mesmo valor são agrupadas juntas.

É possível agrupar todas as fontes de características usando a valor no parâmetro `group by`. Exemplo: NULL.

É possível também agrupar características de vários campos usando a função Array. Exemplo: Array("Field1, "Field2").

Geometrias (se presentes) são combinadas em uma geometria multipartidária para cada grupo. Atributos de saída são computados dependendo de cada definição dada.

O algoritmo permite usar o padrão aggregates functions 1 do próprio programa do QGIS.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Ver também

Coletar geometrias, *Dissolver*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
Grupo por expressão	`GROUP_BY`	[campo da tabela: qualquer] Padrão: 'NULO'	Escolha o campo de agrupamento. Se *NULL* todas as características serão agrupadas.

continua na próxima página

Tabela 24.603 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Agregados	`AGGREGATES`	[lista]	Lista de saída de definições do campo de camada. Exemplo de definição de um campo: *{'aggregate': 'sum', 'delimiter': ',', 'input': ' $area', 'length': 10, 'name': 'totarea', 'precision': 0, 'type': 6}* Por padrão, a lista contém todos os campo da entrada de camada. Na GUI (Interface gráfica do utilizador), você consegue editar todos os campos e suas definições, você também consegue: • Clique no botão para adicionar um novo campo. • Click no botão para deletar o campo selecionado. • Use e para mudar a ordem dos campos. • Click para voltar ao modo padrão (Os campos das camadas inseridas). Para cada um dos campos dos quais você deseja recuperar informações, você precisa definir o seguinte: ***Input expression*** **(`input`) [expression]** Campo ou expressão da camada de entrada ***Aggregate function*** **(`aggregate`) [enumeration]** Function to use on the input expression to return the aggregated value. These are mainly the *Funções de Agregação*, plus dedicated "first_value" and "last_value" functions for ordered list items. Padrão: *concatenar* (para tipo de dados string), *soma* (para tipo de dados numéricos) ***Delimiter*** **(`delimiter`) [string]** Texto string para separar valores agregados, por exemplo no caso de concatenação. Padrão: , ***Output field name*** **(`name`) [string]** Nome do campo agregado na camada de saída. Por padrão, o nome do campo de entrada é mantido. ***Tipo*** **(`tipo` `) [enumeração]** Data type of the output field. Available types may not be compatible with the output layer provider. One of: **Atenção** For certain field types, e.g. lists, an extra `sub_type` parameter helps refine the specific type of the data. It is automatically set in the GUI but may be needed if you're running the algorithm in Python or from the command line.

Tabela 24.603 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Carregar campos da camada	Apenas GUI	[vetor: qualquer]	Você pode carregar campos de outra camada e usá-los para a agregação
Agregado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output (aggregate) layer *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Agregado	`OUTPUT`	[o mesmo que entrada]	Camada vetorial multigeometria com os valores agregados

Código Python

Algorithm ID: `native:aggregate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Limite

Retorna o fechamento do limite combinatório das geometrias de entrada (ou seja, o limite topológico da geometria).

Somente para camadas de polígono e linha.

Para **geometrias de polígonos** , o limite consiste em todas as linhas que compõem os anéis do polígono.

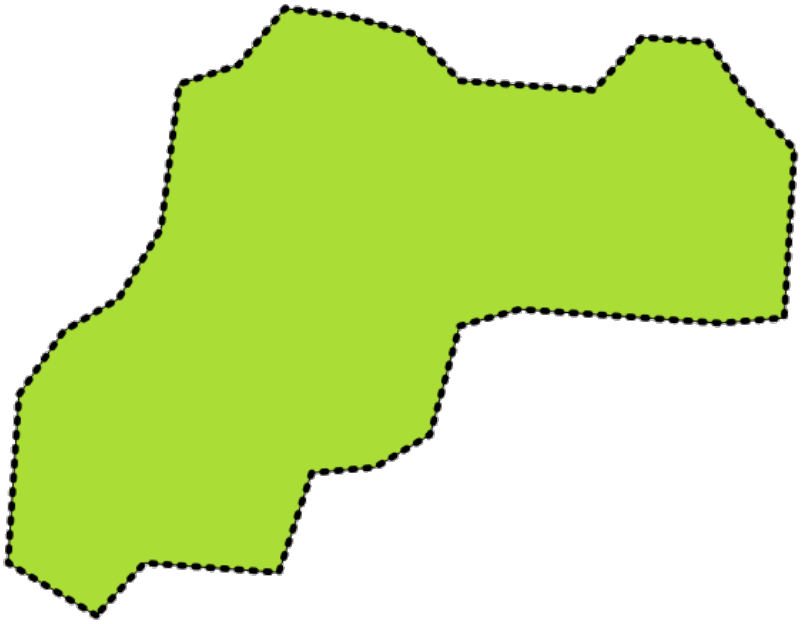


Figura24.90: Limites (linha preta tracejada) da camada do polígono de origem

Para **geometrias de linhas**, os limites são seus pontos finais.

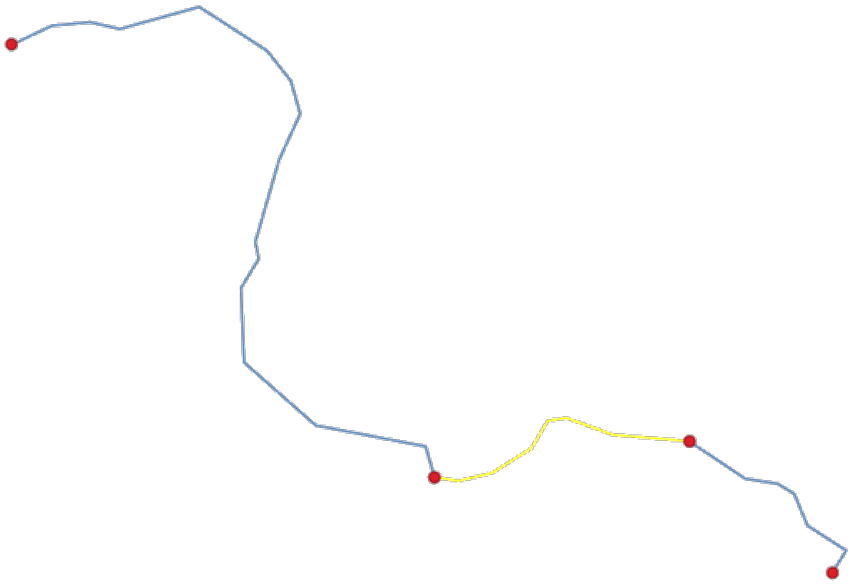


Figura24.91: Camada de limite (pontos vermelhos) para linhas. Em amarelo para uma característica selecionada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Limite	`OUTPUT`	[vetor: ponto, linha] Padrão: `[Criar camada temporária]`	Specify the output (boundary) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Limite	`OUTPUT`	[vetor: ponto, linha]	Limites da camada de entrada (ponto por linha e linha por polígono)

Código Python

Algorithm ID: `native:boundary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Caixas delimitadoras.

Calcula as caixas delimitadoras (envelope) de cada característica em uma camada de entrada. Geometrias poligonais e linha são suportadas.

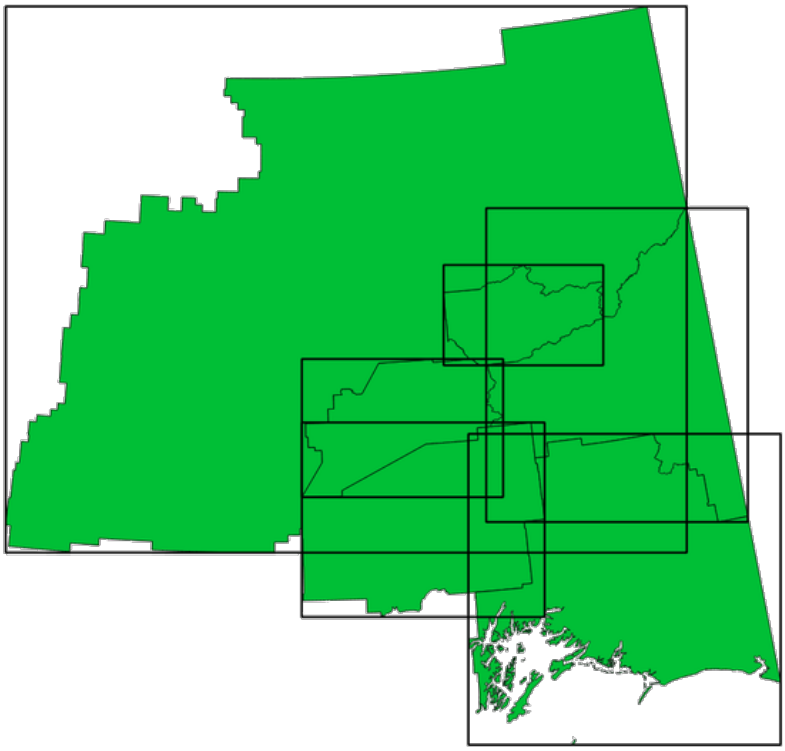


Figura24.92: As linhas pretas representam as caixas delimitadoras de cada feição do polígono

Allows features in-place modification 1 das características dos polígonos.

Ver também

Geometria minimamente limitada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.

continua na próxima página

Tabela 24.607 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Bounds	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output (bounding box) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Bounds	`OUTPUT`	[vetor: polígono]	Bounding boxes of input layer. Other than the input attributes, the output layer also contains following fields: `width`, `height`, `area` and `perimeter` of the generated polygon.

Código Python

Algorithm ID: `native:boundingboxes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Buffer

Computa uma área buffer para todas as características em um camada de entrada, usando uma distância fixa ou definida por informações

É possível usar uma distância negativa para uma entrada de camadas poligonais. Nesse caso, o tampão (buffer) irá resultar em um polígono menor (setback).

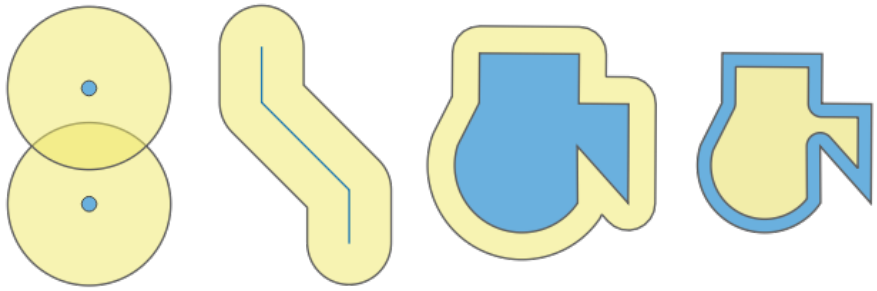


Figura24.93: Buffer (em amarelo) de pontos, linha, polígono com buffer positivo e polígono com buffer negativo

Allows features in-place modification 1 das características dos polígonos.

Default menu: *Vector* ► *Geoprocessing Tools*

Aviso

This algorithm may drop existing primary keys or FID values and regenerate them in output layers, depending on the input parameters.

Ver também

Buffer de distância variável, *Buffer de multi anéis (distância constante)*, *Buffer de largura variável (por valor M)*

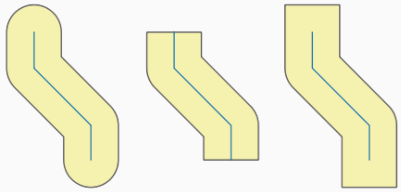
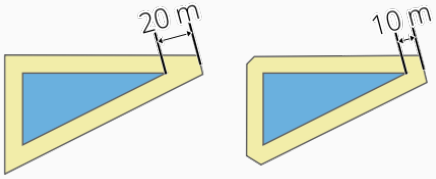
Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Distância	`DISTANCE`	[numeric: double] Padrão: 10.0	Buffer distance (from the boundary of each feature). You can use the Data Defined button on the right to choose a field from which the radius will be calculated. This way you can have different radius for each feature.
Segmentos	`SEGMENTS`	[numeric: integer] Padrão: 5	Controla o número de segmentos usadas para aproximar um quarte de círculo quando criando desvios redondos.

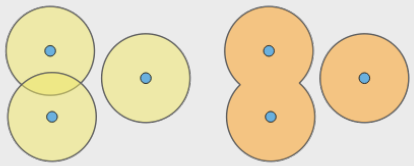
continua na próxima página

Tabela 24.609 – continuação da página anterior

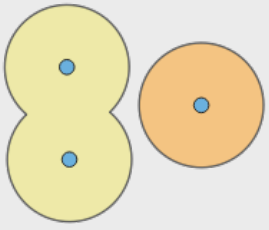
Etiqueta	Nome	Tipo	Descrição
End cap style	`END_CAP_STYLE`	[enumeração] Padrão: 0	Controla como o final das linhas são feitas no tampão. Uma do: • 0 — Round • 1 — Plano • 2 — Quadrado  Figura24.94: Estilos de cobertura redonda, plana e quadrada.
Join style	`JOIN_STYLE`	[enumeração] Padrão: 0	Especifica se redondo, mitra (estilo de corte) ou chanfrado (estilo de corte) devam ser usados quando refinando os cantos da linha. As opções são: • 0 — Round • 1 — Miter (mitra) • 2 — Bevel (chanfrado) Figura24.95: Estilos de junção redonda, serrada e chanfrada
Miter limit	`MITER_LIMIT`	[numeric: double] Padrão: 2.0	Configura a distância máxima da geometria de deslocamento para usar quando criando uma junção mitered como um fato da distância de deslocamento (somente aplicado para junções do estilo miter). Mínimo: 1.0  Figura24.96: Um buffer de 10m com um limite de buffer de 2 e 10m com um limite de 1.

continua na próxima página

Tabela 24.609 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Dissolver resultado	`DISSOLVE`	[boleano] Padrão: Falso	Dissolve o buffer final. Se ``Verdade´´ (marcado), Buffers sobrepostos serão dissolvidos (combinados) em uma característica única multipartidária.  Figura24.97: Padronizado (características de três partes únicas - esquerda), dissolvida (1 característica multipartidária com 2 partes - direita).
Buffered	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output (buffer) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Mantenha características não unidas separadas	`SEPARATE_DISJOINT`	[boleano] Padrão: Falso	If `True` (checked) and dissolved is checked, features that do not overlap or touch will be exported as separate features (instead of parts of a multipart feature).  Figura24.98: Results in 2 single part features

Saídas

Etiqueta	Nome	Tipo	Descrição
Buffered	`OUTPUT`	[vetor: polígono]	Saída (tampão) camada poligonal.

Código Python

Algorithm ID: `native:buffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Centroides

Cria uma nova camada de pontos, com pontos que representam os centróides das geometrias da camada de entrada.

O centróide é um único ponto representando o baricentro (de todas as partes) da característica, então pode ser estar de fora do setor das características. Mas também pode ser um ponto em cada parte da característica.

Os atributos dos pontos na camada de saída são os mesmos das feições originais.

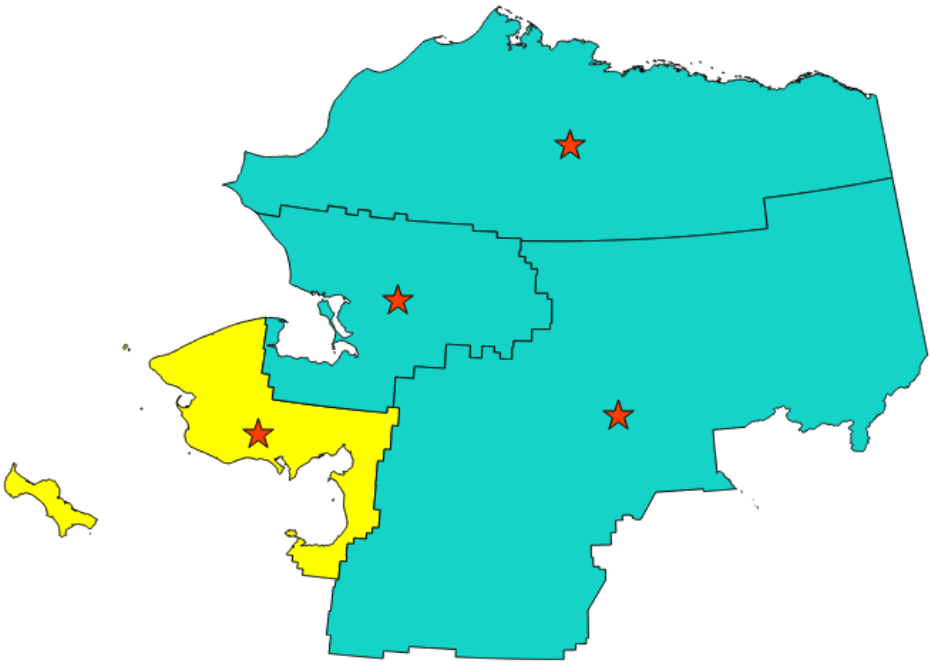


Figura24.99: As estrelas vermelhas representam os centróides das feições da camada de entrada.

permite features in-place modification 1 de pontos da característica.

Default menu: *Vector* ► *Geometry Tools*.

Aviso

This algorithm may drop existing primary keys or FID values and regenerate them in output layers, depending on the input parameters.

Ver também

Ponto na Superfície

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Criar centroide para cada parte	`ALL_PARTS`	[boolean] Padrão: Falso	Se Verdadeiro (marcado), um centroide será criado para cada parte da geometria
Centróides	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output (centroid) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Centróides	`OUTPUT`	[vetor: ponto]	Camada vetorial de ponto de saída (centróides)

Código Python

Algorithm ID: `native:centroids`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Verificar a validade

Executa uma verificação de validade nas geometrias de uma camada vetorial.

As geometrias são classificadas em três grupos (válido, inválido e erro) e para cada grupo é gerada uma camada vetorial com suas características:

- A camada**Valid output** contém somente as características válidas (sem erros topológicos).
- A camada de **Invalid output** contém todas as características encontradas pelo algoritmo.
- A camada de **Error output** é um ponto de camada que mostra onde as características inválidas foram encontradas.

O atributo mesas das camadas geradas conterão alguma informação adicional (“message” da camada **error**, FID” e “_errors” da camada **invalid** e somente “FID” da camada **valid**.

A tabela de atributos de cada camada vetorial gerada conterá algumas informações adicionais (número de erros encontrados e tipos de erro):

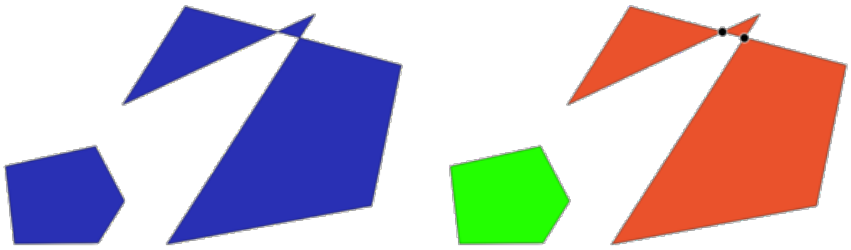


Figura24.100: Esquerda: a camada de entrada. Direita: a camada válida (verde), a camada inválida (laranja)

Default menu: *Vector ► Geometry Tools*.

Ver também

Check Geometry, *Corrigir geometrias*, *Validate coverage*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT_LAYER`	[vector: geometry]	Camada vetorial de entrada
Método	`METHOD`	[enumeração] Padrão: 2	Método a ser usado para verificar a validade. Opções: • 0: O selecionado é a configuração de digitalização. • 1: QGIS • 2: GEOS
Ignore ring self intersection	`IGNORE_RING_SELF`	[boleano] Padrão: Falso	Ignore anéis de auto intersecções quando checando a validade.

continua na próxima página

Tabela 24.614 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Valid output	`VALID_OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the vector layer to contain a copy of the valid features of the source layer. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Saída inválida	`INVALID_OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Vector layer containing copy of the invalid features of the source layer with the field `_errors` listing the summary of the error(s) found. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Error output	`ERROR_OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Point layer of the exact position of the validity problems detected with the `message` field describing the error(s) found. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Contagem de erros	`ERROR_COUNT`	[numeric: integer]	O número de geometrias que causaram erros.
Error output	`ERROR_OUTPUT`	[vetor: ponto]	Camada sinalizada da exata posição dos problemas validados detectados com o campo `message` descrevendo os erros encontrados.
Contagem de feições inválidas	`INVALID_COUNT`	[numeric: integer]	O número de geometrias inválidas.

continua na próxima página

Tabela 24.615 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Saída inválida	`IN-VALID_OUTPUT`	[o mesmo que entrada]	Camada vetorial contendo uma cópia das características inválidas da camada fonte com o campo `_errors` listando o sumário dos erros encontrados.
Contagem de feições válidas	`VALID_COUNT`	[numeric: integer]	O número de geometrias válidas.
Valid output	`VALID_OUTPUT`	[o mesmo que entrada]	Camada vetorial contendo uma cópia das feições válidas da camada de origem.

Código Python

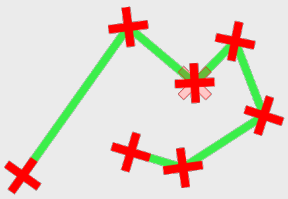
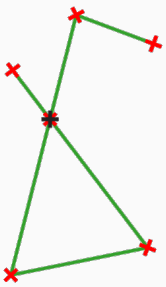
Algorithm ID: `native:checkvalidity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

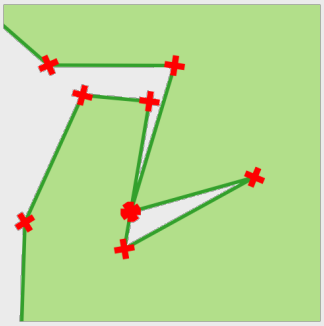
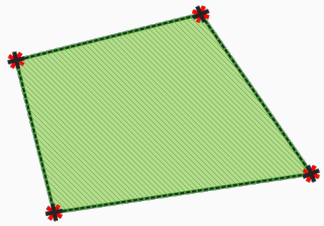
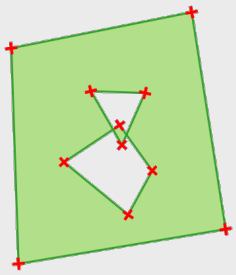
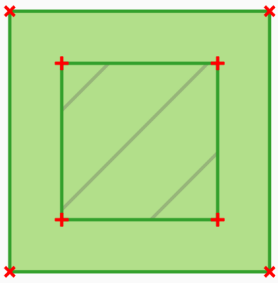
Tipos de mensagens de erro e seus significados

Tabela24.616: Some examples of geometry check failures in QGIS

Geometry context	GEOS validation and error message	QGIS validation and error message
Consecutive points on a line have the same coordinates 	✗ *Repeated point*	✗ *Line a contains x duplicate node(s) at b*
Segments of a line intersect each other 		✗ *Segments a and b of line c intersect at d*

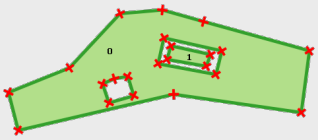
continua na próxima página

Tabela 24.616 – continuação da página anterior

Geometry context	GEOS validation and error message	QGIS validation and error message
Polygon geometry touches itself and generates a ring 	✗ *Ring self-intersection*	✗ *Ring self-intersection*
Two rings (exterior or interior) of a polygon geometry are identical 	✗ *Duplicate rings*	
Geometry touches itself 	✗ *Self-intersection*	
A polygon geometry is on top of another polygon geometry 	✗ *Nested shell*	

continua na próxima página

Tabela 24.616 – continuação da página anterior

Geometry context	GEOS validation and error message	QGIS validation and error message
Part of a MultiPolygon geometry is within a hole of a MultiPolygon geometry 		*Polygon a lies inside polygon b*
Point geometry does not have a proper coordinate pair. The coordinate pair does not contain a latitude value and a longitude value in that order.	*Invalid coordinate*	

Coletar geometrias

Pega uma camada vetorial e coleta suas geometrias em novas geometrias multipartes.

Um ou mais atributos podem ser especificados para coletar apenas geometrias pertencentes à mesma classe (tendo o mesmo valor para os atributos especificados), alternativamente todas as geometrias podem ser coletadas.

Todas as geometrias de saída serão convertidas em múltiplas geometrias, mesmo aquelas com apenas uma peça. Este algoritmo não dissolve geometrias sobrepostas - elas serão coletadas juntas sem modificar a forma de cada parte da geometria.

Veja a 'Promote to multipart' ou algoritmos 'Aggregate' para opções alternativas.

Default menu: *Vector ► Geometry Tools*.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Agregar, *Promover para multipartes*, *Dissolver*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
*Campos de id único**	`FIELD`	[campo da tabela: qualquer][list]	Escolha um ou mais atributos para coletar as geometrias
Coletado	`OUTPUT`	[o mesmo que entrada]	Camada vetorial irá coletar geometrias.

Saídas

Etiqueta	Nome	Tipo	Descrição
Coletado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the collected geometries. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Código Python

Algorithm ID: `native:collect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Concave hull

Computes the concave hull of the features from an input point layer.

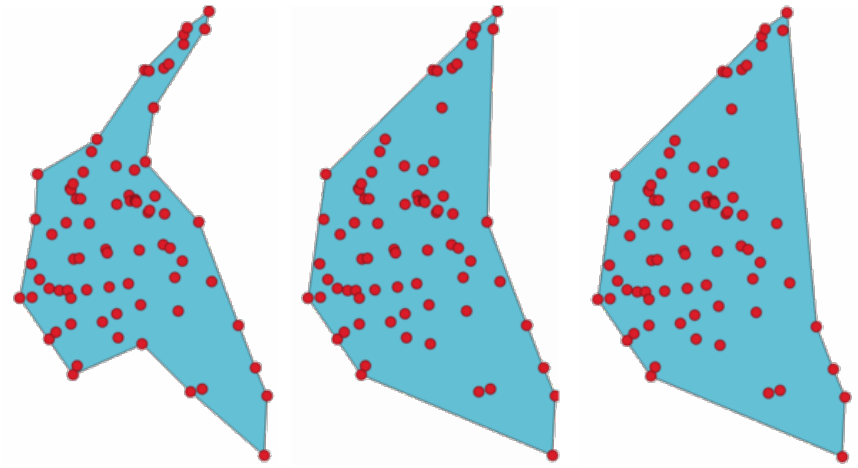


Figura24.101: Cascos côncavos com diferentes limiares (0.3, 0.6, 0.9).

Ver também

Casco convexo.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada de ponto	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Threshold.	`ALPHA`	[numeric: double] Padrão: 0.3	O número vai de 0 (Casco côncavo máximo) para 1 (casco convexo),
Allow holes.	`HOLES`	[boleano] Padrão: Verdadeiro	Escolha se você permitira buracos no casco côncavo final.
Split multipart geometry into singlepart geometries.	`NO_MULTIGEOMET`	[boleano] Padrão: Falso	Verifique se você quer ter geometrias de peça únicas ao invés de ter geometrias de várias peças.
Concave hull.	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Concave hull.	`OUTPUT`	[vetor: polígono]	A camada vetorial de saída.

Código Python

Algorithm ID: `native:concavehull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Converter tipo de geometria

Gera uma nova camada com base em uma já existente, com um tipo diferente de geometria.

A tabela de atributos da camada de saída é a mesma da camada de entrada.

Nem todas as conversões são possíveis. Por exemplo, uma linha de camada pode ser convertida para uma camada de ponto, mas a camada de ponto não pode ser convertida para uma camada de linha.

Ver também

Poligonizar, *Linhas para polígonos*, *Polígonos para linhas*, *Pontos para o caminho*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Novo tipo de geometria	`TYPE`	[enumeração] Padrão: 0	Tipo de geometria a ser aplicada às feições de saída. Um de: • 0 — Centroides • 1 — Nós • 2 — Linestrings • 3 — Multilinestrings • 4 — Polígonos

continua na próxima página

Tabela 24.621 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Convertido	`OUTPUT`	[vector: geometry] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Convertido	`OUTPUT`	[vector: geometry]	Camada vetorial de saída - o tipo depende dos parâmetros

Código Python

Algorithm ID: `native:convertgeometrytype`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Converter para geometrias curvas

Converte uma geometria em sua geometria curva equivalente.

As geometrias já curvas serão mantidas sem alterações.

permite features in-place modification 1.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha ou polígono]	Camada vetorial de entrada
Tolerância de distância máxima	`DISTANCE`	[numeric: double] Padrão: 0.000001	A distância máxima permitida entre a localização original dos vértices e onde eles cairiam nas geometrias curvas convertidas

continua na próxima página

Tabela 24.623 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tolerância máxima de ângulo	`ANGLE`	[numeric: double] Padrão: 0.000001	Segmentos são considerados adequados para substituir com um arco se todos os pontos são espaços regularmente no arco candidato. Esse parâmetro específica o desvio angular máximo (em graus) permitido quando testando espaçamento regular de pontos. Entre 0 e 45°.
Curvas	`OUTPUT`	[vector: compoundcurve or curvepolygon] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Curvas	`OUTPUT`	[vector: compoundcurve or curvepolygon]	Camada vetorial de saída com geometrias curvas

Código Python

Algorithm ID: `native:converttocurves`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Casco convexo.

Calcula o casco convexo para cada característica em uma camada de entrada.

Veja o algoritmo “Limite geométrico mínimo” para o cálculo de casco convexo que cobre toda a camada ou grupo de subconjuntos de características.

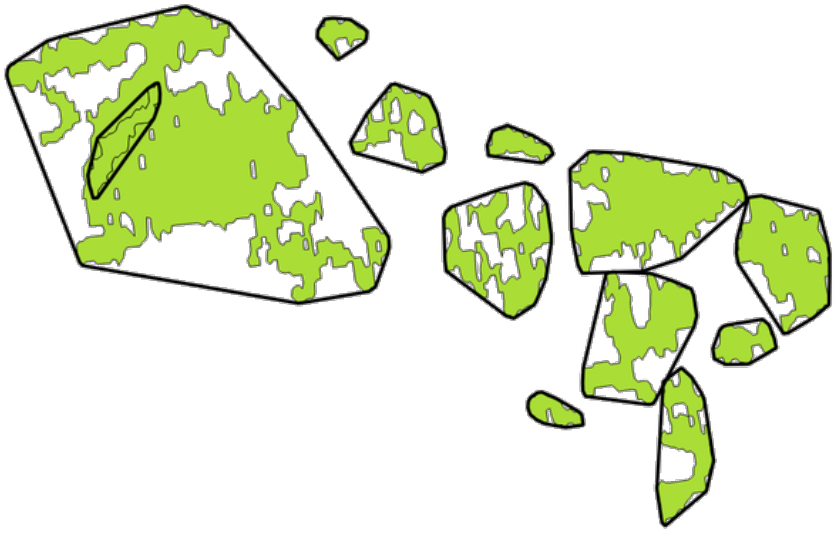


Figura24.102: Linha pretas identificam o casco convexo para cada característica da camada.

Allows features in-place modification 1 das características dos polígonos.

Default menu: *Vector* ► *Geoprocessing Tools*

Ver também

Geometria minimamente limitada., *Concave hull*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Convex hull	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Convex hull	`OUTPUT`	[vetor: polígono]	The output (convex hull) vector layer. Other than the input attributes, the output layer also contains the following fields: `area` and `perimeter` of the generated polygon.

Código Python

Algorithm ID: `native:convexhull`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Crie buffers de cunha (wedge buffer).

Cria formas de buffers de cunha de pontos de entrada.

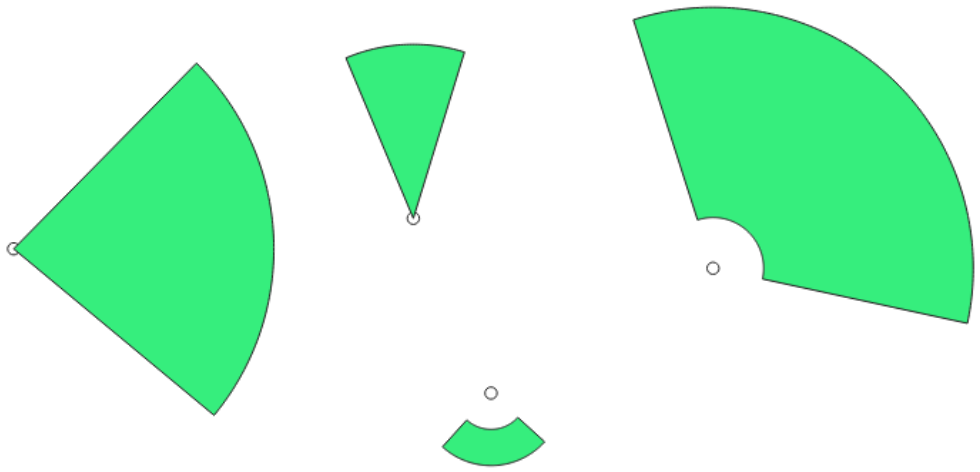


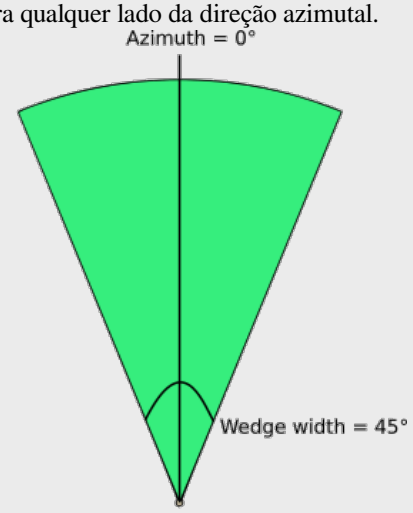
Figura24.103: Buffers de cunha.

A saída nativa desse algoritmo é geometrias CurvePolygon, mas esses podem ser automaticamente segmentadas para polígonos, dependendo no formato de saída.

Ver também

Buffer, *Buffer de largura variável (por valor M)*, *Buffers cônicos.*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Azimute (graus do Norte)	`AZIMUTH`	[numeric: double] Padrão: 0.0	Ângulo (em graus) como o valor intermediário da cunha.
Wedge width (in degrees)	`WIDTH`	[numeric: double] Padrão: 45.0	Largura (em graus) do buffer. A cunha se estenderá para a metade da largura angular para qualquer lado da direção azimutal.  Figura24.104: Valores azimutais e de largura da cunha de buffer.
Raio externo	`OUTER_RADIUS`	[numeric: double] Padrão: 1,0	O *size* (cumprimento) exterior da cunha; o tamanho vem do ponto fonte para a ponta da forma da cunha.
Raio interno Opcional	`INNER_RADIUS`	[numeric: double] Padrão: 0.0	Valor interno de raio. Se 0, a cunha começara do ponto de origem.
Buffers	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Buffers	`OUTPUT`	[vetor: polígono]	A saída (buffer de cunha) da camada vetorial.

Código Python

Algorithm ID: `native:wedgebuffers`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Triangulação de Delaunay

Cria uma camada de polígono com a triangulação de Delaunay correspondente à camada de ponto de entrada.

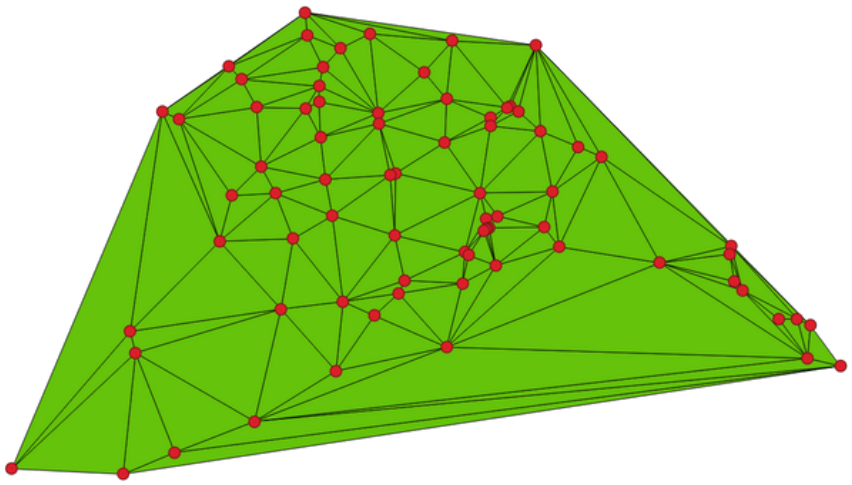


Figura24.105: Triangulação de Delaunay em pontos

Default menu: *Vector* ► *Geometry Tools*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Tolerância Opcional	`TOLERANCE`	[numeric: double] Padrão: 0.0	Specifies an optional snapping tolerance which can be used to improve the robustness of the triangulation.
Add point IDs to output	`ADD_ATTRIBUTES`	[boleano] Padrão: Verdadeiro	Specifies whether fields storing involved point features ID should be added to the output. If False, an `id` field is used to identify the polygons.
Triangulação de Delaunay	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Triangulação de Delaunay	`OUTPUT`	[vetor: polígono]	A saída (triangulação de Delaunay) de camada vetorial.

Código Python

Algorithm ID: `native:delaunaytriangulation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Excluir buracos

Pegue a camada poligonal e remova os buracos nos polígonos. Isso cria uma nova camada vetorial na qual polígonos com buracos foram substituídos por polígonos com somente seu anel externo. Atributos não são modificados.

Um parâmetro de área mínima opcional permite remover somente buracos que são menores que uma área especifica limitada. Deixando esse parâmetro em `0.0` resulta em todos os buracos serem removidos.

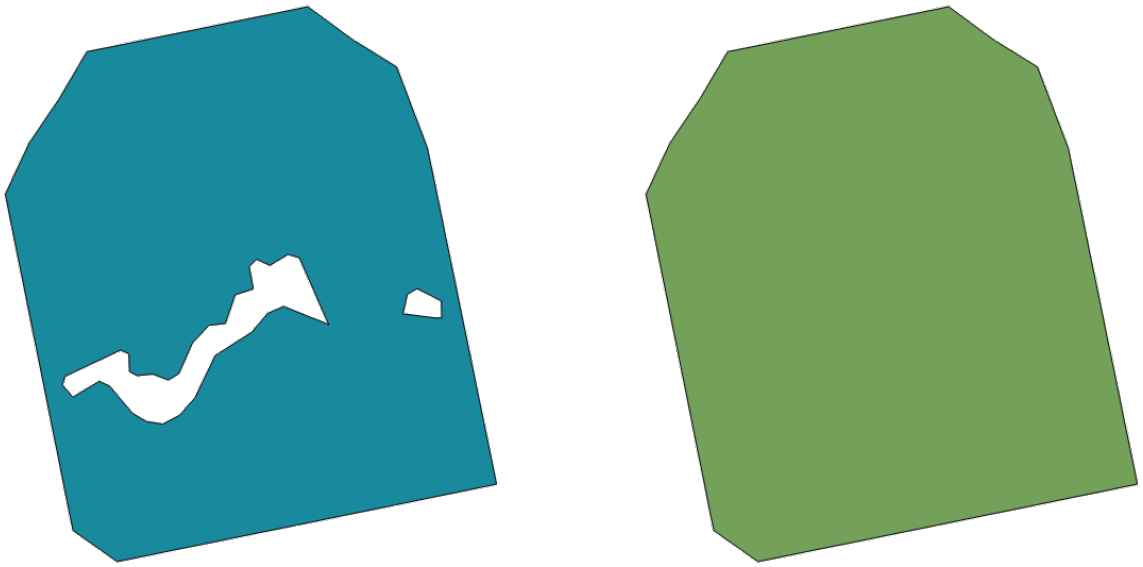


Figura24.106: Antes e depois da limpeza

Allows features in-place modification 1 das características dos polígonos.

Ver também

qgisfixgeometryhole

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	insira a camada vetorial de polígonal.
Remover buracos com área menor que Opcional	`MIN_AREA`	[numeric: double] Padrão: 0.0	Somente buracos com uma área menor que esse limite será deletada. Com o valor `0.0`, **all** os buracos serão deletados.
Limpo	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Limpo	`OUTPUT`	[o mesmo que entrada]	A saída (cleaned) da camada vetorial.

Código Python

Algorithm ID: ``native:deleteholes`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Densificar por contagem.

Pega um polígono ou camada de linha e gera uma nova na qual as geometrias possuem um número de vértices maior que o original.

Se as geometrias tiverem valores Z ou M presentes, eles serão interpolados linearmente nos vértices adicionados.

O número de novos vértices a serem adicionados a cada segmento é especificado como um parâmetro de entrada.

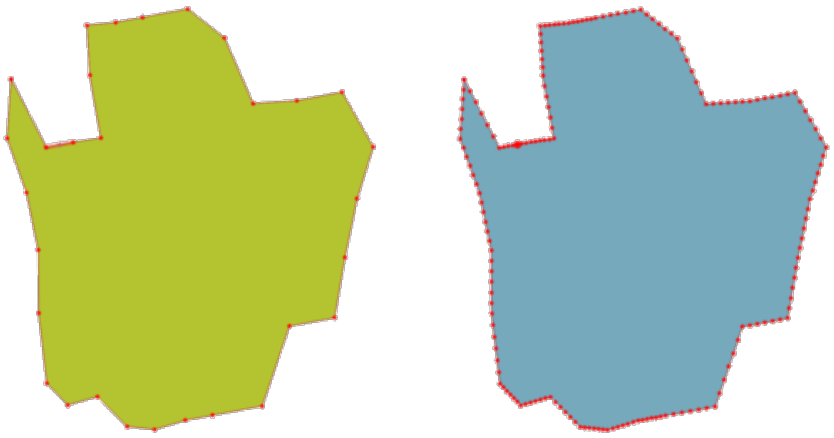


Figura24.107: Os pontos vermelhos mostram os vértices antes e depois da densificação

permite features in-place modification 1.

Default menu: *Vector ► Geometry Tools*.

Ver também

Densificar por intervalo

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Vértices a serem adicionados	`VERTICES`	[numeric: integer] Padrão: 1	Número de vértices a serem adicionados em cada segmento
Densificado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Densificado	`OUTPUT`	[o mesmo que entrada]	A saída (densificado) da camada vetorial

Código Python

Algorithm ID: `native:densifygeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Densificar por intervalo

Pega um polígono ou camada de linha e gera uma nova na qual as geometrias possuem um número de vértices maior que o original.

As geometrias são densificadas adicionando regularmente colocados vértices extras dentro de cada segmento para que assim a distâncias máxima entre dois vértices não exceda a distância específica.

Se as geometrias tiverem valores Z ou M presentes, eles serão interpolados linearmente nos vértices adicionados.

Exemplo

Especificando a distância de 3 faria com que o segmento `[0 0] -> [10 0]` fosse convertido para `[0 0] -> [2.5 0] -> [5 0] -> [7.5 0] -> [10 0]`, desde que 3 vértices extras sejam requeridos no segmento e espaçando estes em incrementos de 2.5 permitindo-os ser uniformemente espaçados pelos segmentos.

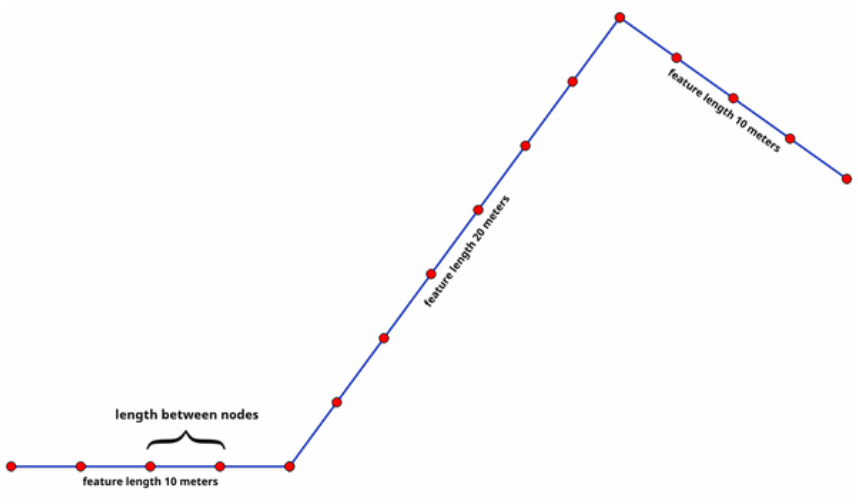


Figura24.108: Geometrias densificadas em um dado intervalo.

permite features in-place modification 1.

Ver também

Densificar por contagem.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Intervalo entre vértices a serem adicionados	`INTERVAL`	[numeric: double] Padrão: 1,0	Distância máxima entre dois vértices consecutivos
Densificado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Densificado	`OUTPUT`	[o mesmo que entrada]	A saída (densificado) da camada vetorial

Código Python

Algorithm ID: `native:densifygeometriesgivenaninterval`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Dissolver

Pegue uma camada vetorial e combine as características em novas. Um ou mais atributos podem ser especificados para dissolver características pertencendo a mesma classe (tendo o mesmo valor para os atributos especificados), alternativamente todas as características podem ser dissolvidas para uma única característica.

Todas as geometrias serão convertidas para geometrias múltiplas. No caso da entrada ser uma camada de polígono, limites comuns de polígonos adjacentes sendo dissolvidos serão apagados. Se ativado, a opção de configuração "mantenha características não juntadas separadas" ira fazer com que características e partes que não se sobrepõem ou não se tocam serem exportadas como características separadas (ao invés de partes de uma única característica multipartidária).

A tabela de tributos resultante terá os mesmos campos que a camada de entrada. Os valores de saída nos campos de camada são aqueles da primeira entra que passa a ser processado.

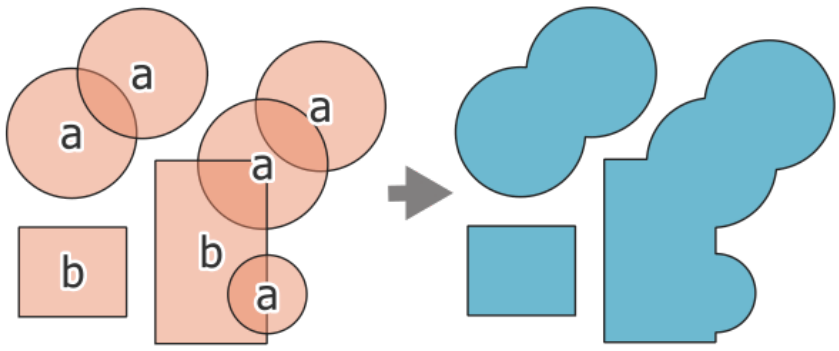


Figura24.109: Dissolvendo uma cama em uma única multipartidária.

Default menu: *Vector* ► *Geoprocessing Tools*

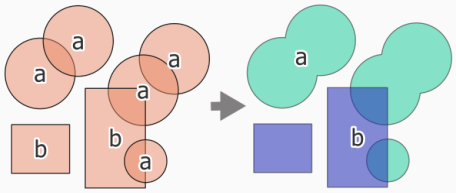
> **Aviso**
>
> This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

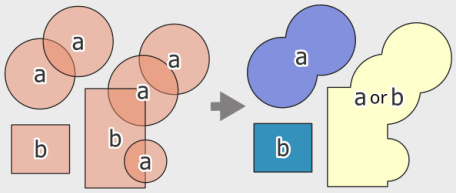
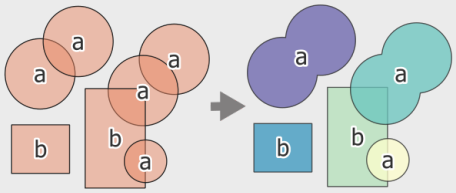
Dissolve coverage, *Agregar*, *Coletar geometrias*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Dissolver campo(s) Opcional	`FIELD`	[campo da tabela: qualquer][list] Padrão: []	As feições com o mesmo valor para o(s) campo(s) selecionado(s) serão substituídas por uma única e suas geometrias serão mescladas. Se nenhum campo for fornecido, todos os recursos serão dissolvidos, resultando em um único recurso (feição multiparte).  Figura24.110: Dissolvendo a camada de polígono em uma atributo comum (2 características multipartidárias)
Dissolvido	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Mantenha características não unidas separadas	`SEPARATE_DISJOINT`	[boleano] Padrão: Falso	Partes de características não dissolvidas são exportadas como características separadas (ao invés de partes de uma características de várias partes).  Figura24.111: Fonte (esquerda), todas dissolvidas (3 características distintas - direita)  Figura24.112: Fonte (esquerda), dissolve no campo (5 características distintas - direita) .

Saídas

Etiqueta	Nome	Tipo	Descrição
Dissolvido	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída com geometrias dissolvidas

Código Python

Algorithm ID: `native:dissolve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Drape (configure o valor Z do raster)

Use valores amostrados de uma banda dentro de uma camada de raster para configurar o valor em Z para todo vértice sobreposto na característica da geometria. Os valores do raster podem ser opcionalmente escalados para uma quantidade predefinida.

Se os valores Z já existirem na camada, eles serão substituídos pelo novo valor. Se nenhum valor Z existir, a geometria será atualizada para incluir a dimensão Z.

permite features in-place modification 1 de um ponot, linha, e característica polígonais com Z habilitado.

Ver também

Definir o valor M da matriz, *Definir valor Z*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Camada raster	`RASTER`	[raster]	Camada raster com valores Z
Número da banda	`BAND`	[banda raster] Padrão: 1	A banda raster para pegar os valores Z do.
Value for No-Data or non-intersecting vertices	`NODATA`	[numeric: double] Padrão: 0.0	Valor para ser usado em caso do vértice não faça intersecção (um pixel válido) do raster.
Scale factor	`SCALE`	[numeric: double] Padrão: 1,0	Valor de escala: os valores da banda são multiplicados por este valor.
Offset	`OFFSET`	[numeric: double] Padrão: 0.0	Valor de desvio: é adicionado algebricamente para os valores de banda depois de aplicada o "Fator escala"
Atualizado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer (with Z values from the raster layer). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Atualizado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída com valores Z da camada raster

Código Python

Algorithm ID: `native:setzfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Diminua os valores de M/Z

Remove os valores M (medida) ou Z (altitude) das geometrias de entrada.

Ver também

Definir valor M, *Definir valor Z*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Entre o valor vetorial com valores de M ou Z
Drop M Values	`DROP_M_VALUES`	[boleano] Padrão: Falso	Remove os valores M das geometrias
Drop Z Values	`DROP_Z_VALUES`	[boleano] Padrão: Falso	Remove os valores Z das geometrias
Z/M Dropped (Z/M diminuídos)	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Z/M Dropped (Z/M diminuídos)	`OUTPUT`	[o mesmo que entrada]	A saída da camada vetorial (idêntica à camada inserida, com exceção que as dimensões de M e/ou Z têm sido removidas das geometrias).

Código Python

Algorithm ID: `native:dropmzvalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Eliminar polígonos selecionados

Combina polígonos selecionados da camada de entrada com certos polígonos adjacentes apagando seu limite comum. O polígono adjacente pode ser aquele com a maior ou menor área ou aquele que compartilha a maior fronteira comum com o polígono a ser eliminado.

Eliminar é normalmente usado para se livrar de lascas poligonais, leia-se, pequenos polígonos que são resultados de intersecção poligonal processadas os os limites das entradas são similares mas não idênticas.

Default menu: *Vector ► Geoprocessing Tools*

Ver também

Corrigir geometrias

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	insira a camada vetorial de polígonal.
Merge selection with the neighboring polygon with the	`MODE`	[enumeração] Padrão: Não definido	Escolha o parâmetro que será usado para se livrar dos polígonos selecionados. • 0 — Maior Área • 1 — Menor Área • 2 — Limite comum maior

continua na próxima página

Tabela 24.644 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Eliminado	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Eliminado	`OUTPUT`	[vetor: polígono]	A saída de camada poligonal vetorizada.

Código Python

Algorithm ID: `qgis:eliminateselectedpolygons`.

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Explodir linhas

Pega uma camada de linhas e cria uma nova na qual cada camada de linha é substituída por um conjunto de linhas representando os segmentos da linha original.

Cada linha na camada resultante contém apenas um ponto inicial e um ponto final, sem vértices intermediários entre eles.

If the input layer consists of CircularStrings or CompoundCurves, the output layer will be of the same type and contain only single curve segments.

Nota

- This algorithm drops existing primary keys or FID values and regenerates them in output layers.
- This algorithm does not require valid geometries as input.

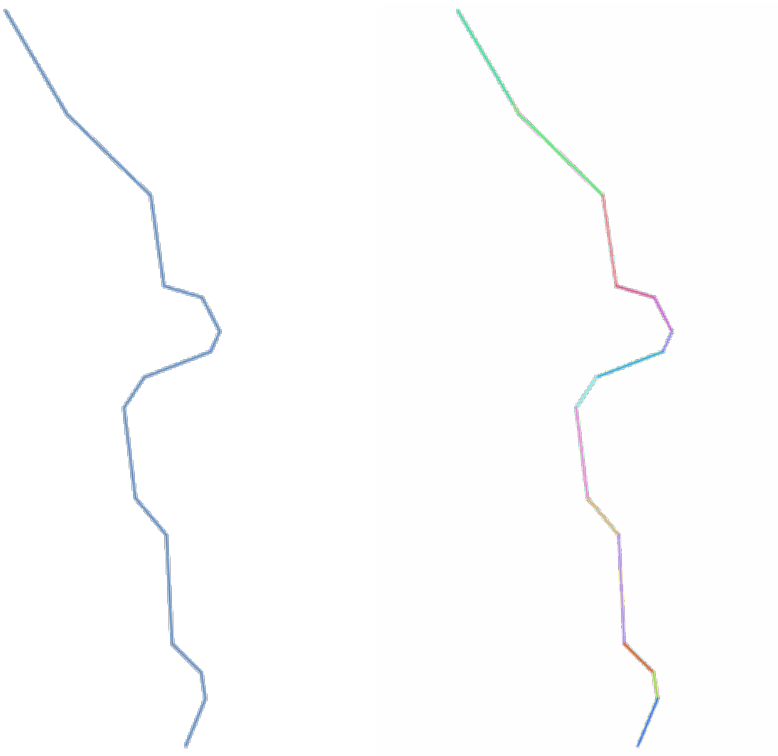


Figura24.113: A camada de linha original e a explodida

Permite features in-place modification 1 da linha de características.

Ver também

Subdividir, *Linha de substring.*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Explodido	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Explodido	`OUTPUT`	[vetor: linha]	A camada de saída de linha vetorial com as características representando cada segmento de camada inserida.

Código Python

Algorithm ID: `native:explodelines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Linhas estendidas.

Estenda linhas geométricas por meio da especificação da quantidade no começo e ao final da linha.

Linhas são estendidas usando o rolamento da primeiro e último segmento de linhas.

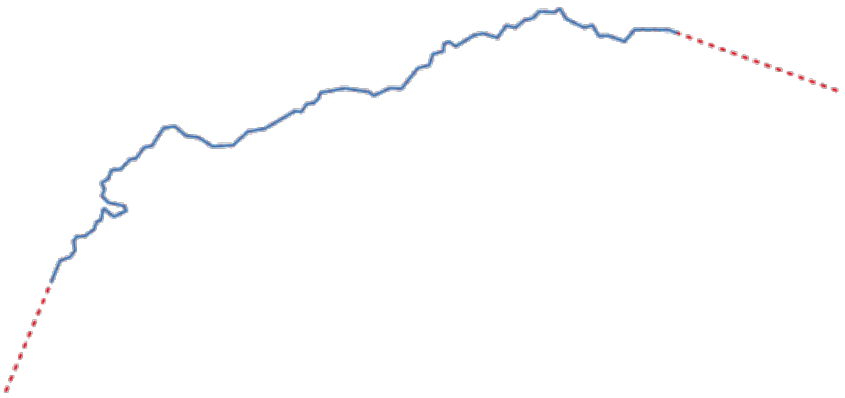


Figura24.114: Os traços vermelhos representam a extensão inicial e final da camada original

Permite features in-place modification 1 da linha de características.

Ver também

Linha de substring.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Iniciar distância	`START_DISTANCE`	[numeric: double] Padrão: 0.0	Distância para estender o primeiro segmento de linha (ponto de início).
Fim da distância	`END_DISTANCE`	[numeric: double] Padrão: 0.0	Distância para estender o último segmento de linha (ponto final).
Estendido	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Estendido	`OUTPUT`	[vetor: linha]	A saída (estendida) da camada vetorial de linha.

Código Python

Algorithm ID: `native:extendlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair valores M

Extrai valores M de geometrias em atributos de feição.

Por padrão, somente o valor de M do primeiro vértice de cada característica é extraído, de qualquer forma, o algoritmo pode opcionalmente calcular estatísticas em todos valores geométricos de M, incluindo soma, média, mínimo e máximo.

Ver também

Extrair valores Z, *Definir valor M*, *Diminua os valores de M/Z*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Summaries to calculate	`SUMMARIES`	[enumeração] Padrão: [0]	Estatísticas sobre os valores M de uma geometria. Um ou mais de: • 0 — Primeiro • 1 — Último • 2 — Contagem • 3 — Soma • 4 — Média • 5 — Mediana • 6 — St.dev (pop) • 7 — Mínimo • 8 — Máximo • 9 — Alcance • 10 — Minoria • 11 — Maioria • 12 — Variedade • 13 — Q1 • 14 — Q3 • 15 — IQR
Output column prefix	`COLUMN_PREFIX`	[string] Padrão: 'm_'	O prefixo para a saída (M) da coluna.
Extraído	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Extraído	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída (com valores M)

Código Python

Algorithm ID: `native:extractmvalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair vértices específicos

Pega uma camada vetorial e gera uma camada de pontos com pontos que representam vértices específicos nas geometrias de entrada.

Por exemplo, este algoritmo pode ser usado para extrair o primeiro ou o último vértice na geometria. Os atributos associados a cada ponto são os mesmos associados à feição à qual o vértice pertence.

O parâmetro de índice de vértice aceita uma string separada por vírgula, especificando os índices dos vértices a serem extraídos. O primeiro vértice corresponde a índice de 0, o segundo vértice corresponde ao um índice de 1, ETC. Índices negativos podem ser usados para achar vértices no final da geometria, por exemplo, um índice de -1 corresponde ao último vértice, =2 corresponde ao segundo vértice, ETC.

Campos adicionais são adicionados nos vértices indicando a posição específica do vértice (por exemplo, 0, -1, etc), o índice do vértice original, a parte do vértice e o seu índice dentro da parte (como igual ao anel de polígonos) distancia ao longo da geometria original e do ângulo bissetor do vértice da geometria original.

permite features in-place modification 1 de pontos da característica.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Extrair vértices, *Filtrar vértices por valor M*, *Filtrar vértices pelo valor Z*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Índices de vértices	`VERTICES`	[string] Padrão: '0'	Strings separadas por vírgula dos índices dos vértices para extrair.
Vértices	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Vértices	`OUTPUT`	[vetor: ponto]	The output (point) vector layer containing the specified vertices from the input layer geometries. Other than the input attributes, the output layer also contains the following fields: • `vertex_pos`: the index of the vertex in the geometry, as requested by the user. • `vertex_index`: the index of the vertex within the geometry. The first vertex is indexed as 0. • `vertex_part`: the part number of the geometry that the vertex belongs to. For single-part geometries, this is always 0. For multi-part geometries (e.g. MultiPolygon or MultiLine), this indicates which part the vertex comes from. • `vertex_part_ring`: for polygon geometries, this indicates whether the vertex belongs to the outer ring (0) or one of the inner rings (1, 2, …). • `vertex_part_index`: the index of the vertex in its specific ring (outer or inner). • `distance`: the cumulative length of line segments from the first vertex of the geometry. • `angle`: the angle (in degrees) between the previous and the current vertex.

Código Python

Algorithm ID: `native:extractspecificvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair vértices

Pega uma camada vetorial e gera uma camada de pontos com pontos representando os vértices nas geometrias de entrada.

Os atributos associados a cada ponto são os mesmos associados à feição à qual o vértice pertence.

Campos adicionais são adicionados nos vértices indicando o índice do vértice (começando por 0), a parte da característica e seu índice dentro da parte (assim como seus anéis para polígonos) distancia ao longo da geometria adicional e o bisetor angular do vértice para a geometria original.

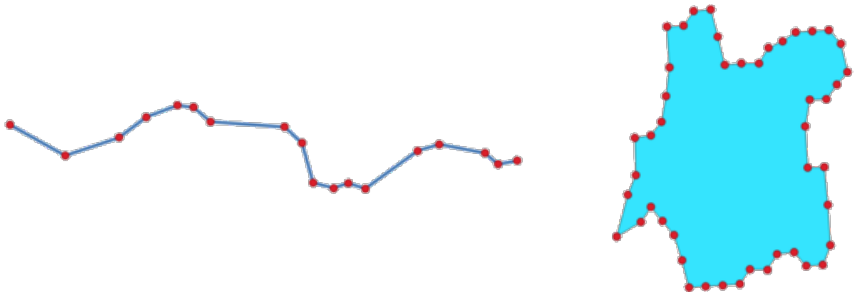


Figura24.115: Vértices extraídos para camada de linha e polígono

permite features in-place modification 1 de pontos da característica.

Default menu: *Vector* ► *Geometry Tools*.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Extrair vértices específicos, *Filtrar vértices por valor M*, *Filtrar vértices pelo valor Z*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Vértices	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Vértices	`OUTPUT`	[vetor: ponto]	The output (point) vector layer containing the vertices from the input layer geometries. Other than the input attributes, the output layer also contains the following fields: • `vertex_index`: the index of the vertex within the geometry. The first vertex is indexed as 0. • `vertex_part`: the part number of the geometry that the vertex belongs to. For single-part geometries, this is always 0. For multi-part geometries (e.g. MultiPolygon or MultiLine), this indicates which part the vertex comes from. • `vertex_part_ring`: for polygon geometries, this indicates whether the vertex belongs to the outer ring (0) or one of the inner rings (1, 2, …). • `vertex_part_index`: the index of the vertex in its specific ring (outer or inner). • `distance`: the distance from the previous vertex. • `angle`: returns approximate angle at a vertex. Usually the average angle between adjacent segments, and can be pictured as the orientation of a line following the curvature of the geometry at the specified vertex.

Código Python

Algorithm ID: `native:extractvertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair valores Z

Extraia os valores de z das geometrias dentro dos atributos de características.

Por padrão, apenas o valor Z do primeiro vértice de cada feição é extraído, no entanto, o algoritmo pode opcionalmente calcular estatísticas sobre todos os valores Z da geometria, incluindo soma, média, mínimo e máximo.

Ver também

Extrair valores M, *Definir valor Z*, *Diminua os valores de M/Z*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Summaries to calculate	`SUMMARIES`	[enumeração] Padrão: [0]	Estatísticas sobre os valores Z de uma geometria. Um ou mais de: • 0 — Primeiro • 1 — Último • 2 — Contagem • 3 — Soma • 4 — Média • 5 — Mediana • 6 — St.dev (pop) • 7 — Mínimo • 8 — Máximo • 9 — Alcance • 10 — Minoria • 11 — Maioria • 12 — Variedade • 13 — Q1 • 14 — Q3 • 15 — IQR
Output column prefix	`COLUMN_PREFIX`	[string] Padrão: 'z_'	O prefixo para a saída (Z) da coluna
Extraído	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Extraído	`OUTPUT`	[o mesmo que entrada]	A camada de saída vetorial (com valores em Z)

Código Python

Algorithm ID: `native:extractzvalues`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Filtrar vértices por valor M

Filtra os vértices com base em seu valor M, retornando geometrias apenas com pontos de vértice que tenham um valor M maior ou igual ao valor mínimo especificado e/ou menor ou igual ao valor máximo.

Se o valor mínimo não for especificado, apenas o valor máximo será testado e, da mesma forma, se o valor máximo não for especificado, apenas o valor mínimo será testado.

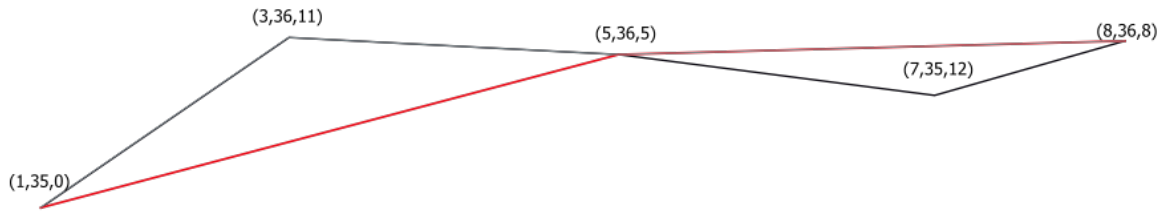


Figura24.116: A linha vermelha representa a linha preta com apenas vértices cujo valor M é <=10.

Allows *features in-place modification* of point, line and polygon features with M enabled

Aviso

Depending on the input geometry attributes and the filters used, the geometries created by this algorithm may no longer be valid.

Ver também

Filtrar vértices pelo valor Z, *Extrair vértices*, *Extrair vértices específicos*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Input vector layer to remove vertices from
Mínimo Opcional	`MIN`	[numeric: double] Padrão: Não definido	Mínimo de valores M permitidos
Máximo Opcional	`MAX`	[numeric: double] Padrão: Não definido	Máximo de valores M permitidos
Filtrado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Filtrado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de feições de saída com apenas os vértices filtrados.

Código Python

Algorithm ID: `native:filterverticesbym`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Filtrar vértices pelo valor Z

Filtra vértices baseados no valor de Z, retornando geometrias com somente pontos de vértices que tem um valor em Z maior ou igual ao valor mínimo específico e ou menor ou igual ao valor máximo.

Se o valor mínimo não for especificado, apenas o valor máximo será testado e, da mesma forma, se o valor máximo não for especificado, apenas o valor mínimo será testado.

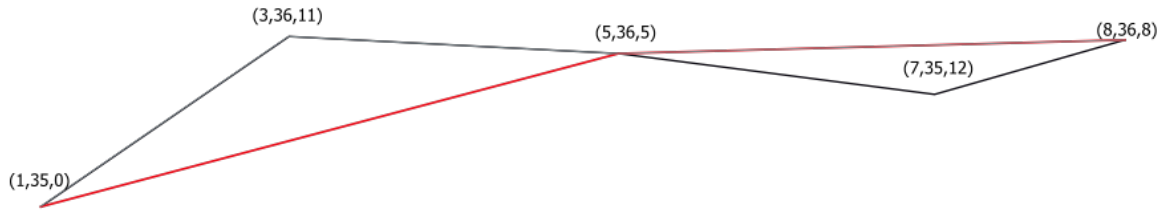


Figura24.117: A linha vermelha representa a linha preta com apenas vértices cujo valor Z é <=10.

Allows *features in-place modification* of point, line and polygon features with Z enabled

Aviso

Depending on the input geometry attributes and the filters used, the geometries created by this algorithm may no longer be valid. You may need to run the *Corrigir geometrias* algorithm to ensure their validity.

Ver também

Filtrar vértices por valor M, *Extrair vértices*, *Extrair vértices específicos*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Input vector layer to remove vertices from
Mínimo Opcional	`MIN`	[numeric: double] Padrão: Não definido	Mínimo de valores Z permitidos
Máximo Opcional	`MAX`	[numeric: double] Padrão: Não definido	Máximo de valores Z permitidos

continua na próxima página

Tabela 24.660 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Filtrado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Filtrado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de feições de saída com apenas os vértices filtrados.

Código Python

Algorithm ID: `native:filterverticesbyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Corrigir geometrias

Tenta crirar uma representação válida de uma dada geometria inválida sem perder qualquer entrada dos vértices. Geometrias já válidas são retornadas sem mais nenhuma intervenção. Sempre camadas de saídas multi geométricas.

permite :ref:`features in-place modification <processing_inplace_edit> ` características de ponto, linha e polígonos sem M estar habilitado.

Aviso

Valores de M serão descartados.

Ver também

Verificar a validade, *Fix Geometry*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Método de reparo	`METHOD`	[enumeração] Padrão: 1	Método usado para reparar geometrias. Uma do tipo: • 0 — `Linha de trabalho`: combina todos os anéis em um set de linhas de nós e então extrai polígonos válidos daquelas linhas • 1 — `Structure`: first makes all rings valid and then merges shells and subtracts holes from shells to generate valid result. Assumes that holes and shells are correctly categorized. **Atenção** Using this method requires QGIS installed with GEOS >= 3.10 (see *Help* ► *About* menu).
Geometrias corrigidas	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Geometrias corrigidas	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída com geometrias corrigidas.

Código Python

Algorithm ID: `native:fixgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Regra de força da mão direita.

Força as geometrias poligonais a respeitarem a regra da mão direita, na qual a área que é delimitada por um polígono é o limite da direita. Em particular, o anel externo é orientado no sentido horário e qualquer anel interior é orientado no sentido anti-horário.

Allows features in-place modification 1 das características dos polígonos.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	Camada vetorial de entrada
Reorientado	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Reorientado	`OUTPUT`	[vetor: polígono]	A camada vetorial de saída com geometrias reorientadas.

Código Python

Algorithm ID: `native:forcerhr`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Linha geodésica dividida no antimeridiano.

Separa a linha em múltiplos segmentos geodésicos, em qualquer momento que a linha cruzar o antimeridiano (+= 180 graus de longitude)

Dividindo no antimeridiano ajuda o visualização da linhas em algumas projeções. A geometria retornada irá sempre ser uma geometria de multipartidária.

Em qualquer momento que as linhas de segmento na entrada geométrica cruza o antimeridiano, eles irão ser separadas em dois segmentos, com a latitude do ponto de interrupção sendo determinada usando uma linha geodésica conectando os pontos em algum lado desses segmentos. O projeto atual elipsódico será usado quando calculando o ponto de interrupção.

Se a geometria de entrada contiver valores M ou Z, estes serão interpolados linearmente para os novos vértices criados no antimeridiano.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Permite features in-place modification 1 da linha de características.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Dividir	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Dividir	`OUTPUT`	[vetor: linha]	A camada vetorizada de linha de saída se divide no antimeridiano.

Código Python

Algorithm ID: `native:antimeridiansplit`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Geometria por expressão

Atualiza geometrias existentes (ou cria novas geometrias) para recursos de entrada usando uma expressão QGIS.

Isso permite modificações geométricas que podem utilizar toda a flexibilidade da expressão motora do QGIS para manipular e criar geometrias para características de saída.

Para ajudar com as funções de expressão do QGIS, veja a ajuda dentro do programa disponível no *expression builder*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
Tipo de geometria de saída	`OUTPUT_GEOMETRY`	[enumeração] Padrão: 0	A saída geométrica depende fortemente da expressão: por exemplo, se você cria um buffer, o tipo de geometria tem que ser um polígono. Um do tipo: • 0 — Polígono • 1 — Linha • 2 — Ponto
A geometria de saída tem valores z	`WITH_Z`	[boleano] Padrão: Falso	Escolha se a geometria de saída deve incluir a dimensão de .
Output geometry has m values	`WITH_M`	[boleano] Padrão: Falso	Escolha se a geometria de saída deva incluir a dimensão M.
Expressão geométrica	`EXPRESSION`	[expressão] Padrão: '$geometria'	Adicione a geometria que você queira usa. Você pode usar o botão para abrir a expressão Diálogo. O diálogo lista todas as expressão relevantes, juntas com a sua ajuda e guia.
Geometria modificada	`OUTPUT`	[vector: geometry] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Geometria modificada	`OUTPUT`	[vector: geometry]	A camada vetorial de saída.

Código Python

Algorithm ID: `native:geometrybyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Interpolar ponto na linha

Cria uma geometria de ponto interpolada a uma distância definida ao longo de geometrias de linha ou curva.

Os valores Z e M são interpolados linearmente a partir dos valores existentes.

Se uma geometria com várias partes é encontrada, somente a primeira parte é considerada quando calculando a substring.

Se a distância especificada for maior que o tamanho da entrada, o resultado da característica será uma geometria nula.

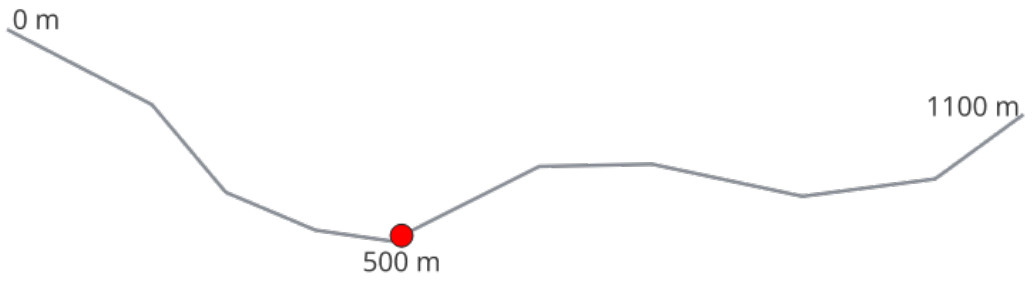


Figura24.118: Ponto interpolado a 500m do início da linha

Ver também

Pontos ao longo da geometria

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Distância	`DISTANCE`	[numeric: double] Padrão: 0.0	Distância do início da linha
Pontos interpolados	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos interpolados	`OUTPUT`	[vetor: ponto]	O ponto de camada de saída vetorizada com as características em uma certa distância junto com a linha ou o limite do polígono.

Código Python

Algorithm ID: `native:interpolatepoint`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Mantenha N partes maiores

Pegue uma camada com polígonos ou multipoligonal e retorne uma nova camada na qual somente *n* polígonos largos de cada característica multipoligonal são mantidos. Se a característica tem *n* ou menos partes, a característica somente será copiada.

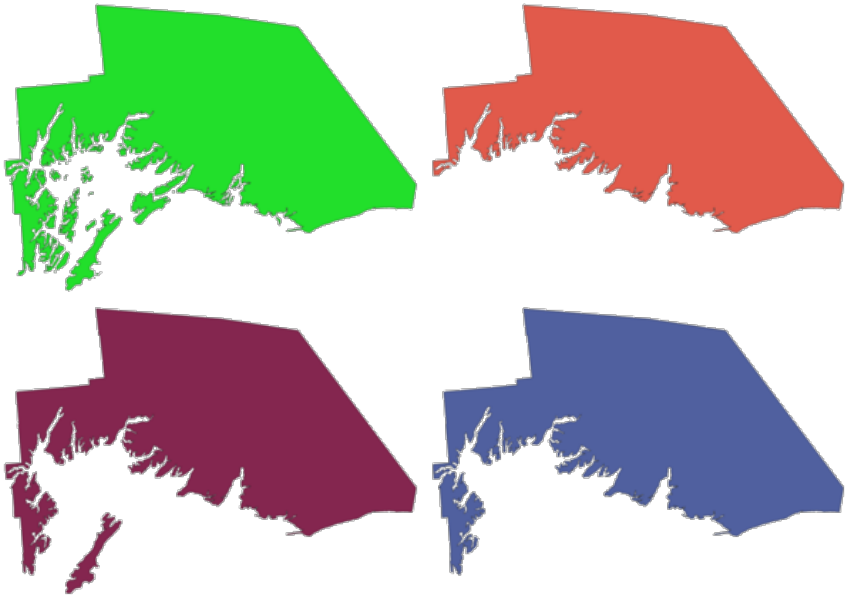


Figura24.119: Sentido horário campo superior esquerdo: Característica com várias, uma, duas ou três partes maiores mantidas.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Polígonos	`INPUT`	[vetor: polígono]	insira a camada vetorial de polígonal.
Parts to keep	`PARTS`	[numeric: integer] Padrão: 1	Número de partes para se manter. Se 1, somente a parte maior da característica será mantida.
Partes	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Partes	`OUTPUT`	[vetor: polígono]	A camada vetorial de saída poligonal com a N partes maiores de cada característica.

Código Python

Algorithm ID: `native:keepnbiggestparts`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Linha de substring.

Retorna a porção da linha (curva) que pode fica o início especificado e o ao final das distâncias (mesurado a partir do começo das linhas).

Os valores Z e M são interpolados linearmente a partir dos valores existentes.

Se uma geometria com várias partes é encontrada, somente a primeira parte é considerada quando calculando a substring.

Figura24.120: Linha de substring com começo posto em 0 metros e final 250 metros.

☑ Permite features in-place modification 1 da linha de características.

Ver também

Linhas estendidas.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Iniciar distância	`START_DISTANCE`	[numeric: double] Padrão: 0.0	Distância ao longo da linha de entrada até o ponto inicial do recurso de saída.
Fim da distância	`END_DISTANCE`	[numeric: double] Padrão: 1,0	Distância ao longo da linha de entrada até o ponto final da feição de saída
Sub-cadeia	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Sub-cadeia	`OUTPUT`	[vetor: linha]	Camada vetorial da linha de saída.

Código Python

Algorithm ID: `native:linesubstring`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Linhas para polígonos

Gera uma camada de polígono usando como anéis de polígono as linhas de uma camada de linha de entrada.

A tabela de atributos da camada de saída é a mesma da camada de entrada.

Default menu: *Vector ► Geometry Tools*.

Ver também

Polígonos para linhas, *Poligonizar*, *Converter tipo de geometria*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Polígonos	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Polígonos	`OUTPUT`	[vetor: polígono]	A saída de camada poligonal vetorizada.

Código Python

Algorithm ID: `qgis:linestopolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Mesclar linhas

Junta todas as partes conectadas das geometrias MultiLineString em somente uma do tipo de geometrias de LineString.

Se quaisquer partes da entrada das geometrias MultiLineString não são conectadas, a geometria resultante será uma MultiLineString contendendo qualquer linha que pode ser mesclada e qualquer linha não conectada.

☑ Permite features in-place modification 1 da linha de características.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Mesclado	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Mesclado	`OUTPUT`	[vetor: linha]	A saída (mesclada) de linha de camada vetorial

Código Python

Algorithm ID: `native:mergelines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Geometria minimamente limitada.

Cria geometrias que podem incluir características de uma camada inserida. As características podem ser agrupadas por campo. A camada de saída conterá uma característica por valor de grupo com a geometria (MBB) que cobre as geometrias das características com valores que coincidem.

As seguintes tipos de geometrias são suportadas:

- caixa delimitadora (envelope)
- retângulo orientado
- círculo
- Casca convexa (convex hull).

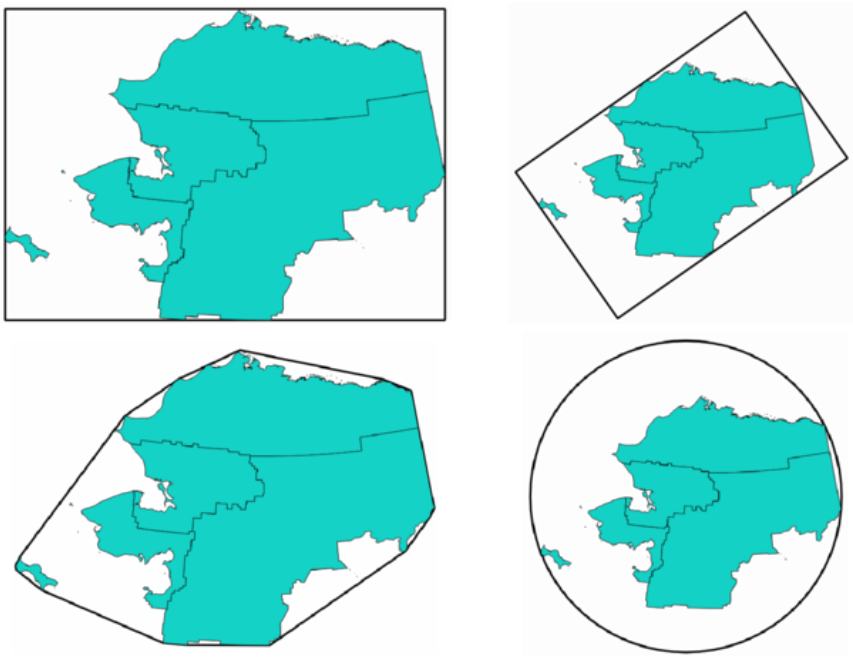


Figura24.121: Sentido horário a partir do campo superior esquerdo: envelope, retângulo orientado, círculo e casca convexa.

Ver também

Círculos mínimos fechados.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Campo Opcional	`FIELD`	[campo da tabela: qualquer]	Características podem ser agrupadas por campo. Se configurado, isso faz com que a camada de saída contenha uma característica por valor de grupo com uma mínima geometria cobrindo somente as características que se coincidem.
Tipo de Geometria	`TYPE`	[enumeração] Padrão: 0	Tipos de geometrias fechadas. Uma do tipo: • 0 — Envelope (Caixa limitante) • 1 — Retângulo Orientado Mínimo • 2 — Círculo Fechado Mínimo • 3 — Casca convexa

continua na próxima página

Tabela 24.680 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Geometria Limitante.	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Geometria Limitante.	`OUTPUT`	[vetor: polígono]	The output (bounding) polygon vector layer. It has the fields from the input layer and, depending on the selected parameter `TYPE`, the output also has the attributes: • `width`, `height`, `area` and `perimeter` of the Envelope (Bounding Box) • `width`, `height`, `area`, `angle` and `perimeter` of the Minimum Oriented Rectangle • `radius` and `area` of the Minimum Enclosing Circle • `area` and `perimeter` of the Convex Hull

Código Python

Algorithm ID: `qgis:minimumboundinggeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Círculos mínimos fechados.

Calcula o fechamento mínimo dos círculos das características na camada de entrada.

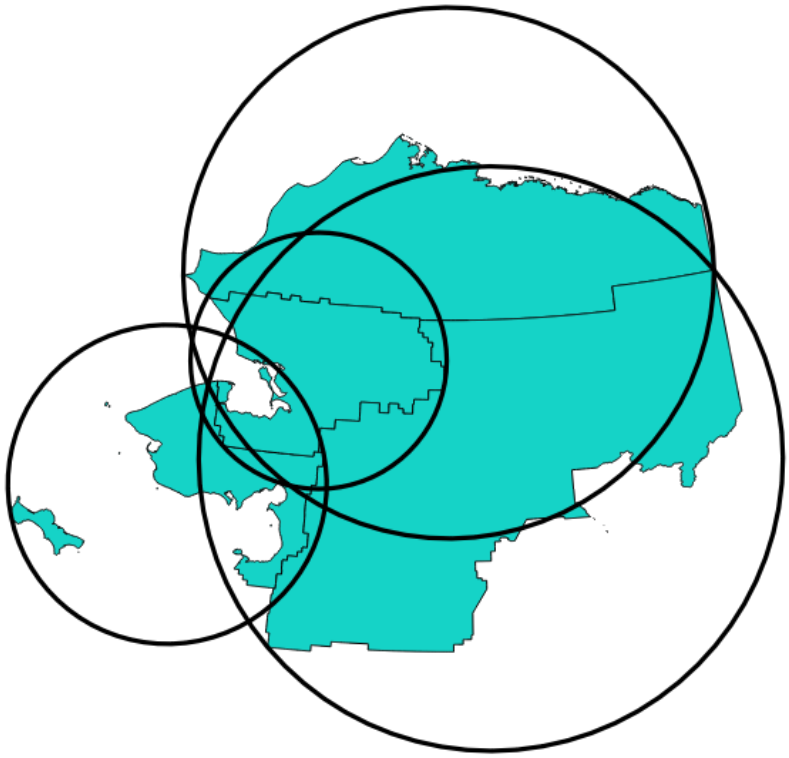


Figura24.122: Círculos fechados para cada característica.

Allows features in-place modification 1 das características dos polígonos.

Ver também

Geometria minimamente limitada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Number of segments in circles	`SEGMENTS`	[numeric: integer] Padrão: 72	The number of segments used to approximate a circle. Minimum 8, maximum 100000.

continua na próxima página

Tabela 24.682 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Minimum enclosing circles	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Minimum enclosing circles	`OUTPUT`	[vetor: polígono]	The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `area` and `radius` of the generated polygon.

Código Python

Algorithm ID: `native:minimumenclosingcircle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Buffer de multi anéis (distância constante)

Computa buffer de multi anéis (*donut*) para as características da camada de entrada, usando uma distância fixa ou dinâmica e números de anéis.

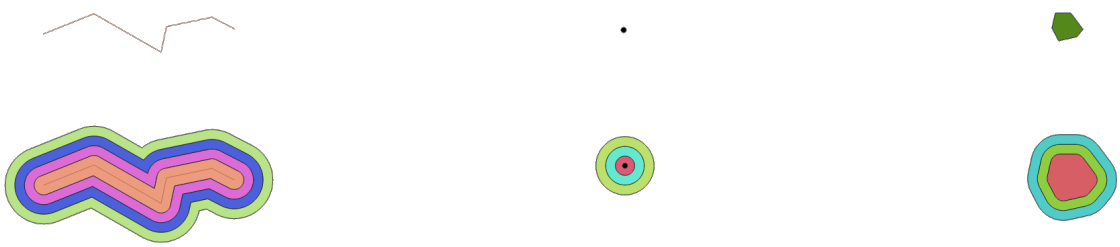


Figura24.123: Buffer de multi anéis para uma linha, ponto ou camada poligonal.

Allows features in-place modification 1 das características dos polígonos.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Buffer, *Buffer de distância variável*, *Retângulos, ovais, diamantes*, *Buffer de lado único*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Número de anéis	`RINGS`	[numeric: integer] Padrão: 1	O número de anéis. Pode ser um valor único (mesmo número de anéis para todas as características) ou pode ser pego por características de data (o número de anéis depende dos valores das características).
Distância entre anéis	`DISTANCE`	[numeric: double] Padrão: 1,0	A distância entre os anéis. Pode ser um valor único (mesma distância para todas as características) ou pode ser trazido dos dados das características (a distância depende do valores da característica).
Buffer de multi anéis (distância constante)	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Buffer de multi anéis (distância constante)	`OUTPUT`	[vetor: polígono]	The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `ringId` and `distance` of the generated buffer polygon.

Código Python

Algorithm ID: `native:multiringconstantbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Multipartes para partes únicas.

Separa características multipartidárias na camada de entrada em características únicas.

Os atributos da camada de saída são os mesmos que os originais, mas divididos em feições únicas.

Figura24.124: Esquerda da camada fonte multipartidária e o resultado de parte única de saída à direita .

Allows features in-place modification 1 de ponto, linha e características poligonais.

Default menu: *Vector* ► *Geometry Tools*.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Coletar geometrias, *Promover para multipartes*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada

continua na próxima página

Tabela 24.686 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Partes individuais	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Partes individuais	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída.

Código Python

Algorithm ID: `native:multiparttosingleparts`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Linhas de desvio.

Desloca as linhas por uma distância especificada. Distâncias positivas deslocarão as linhas à esquerda e distâncias negativas deslocarão as linhas à direita.

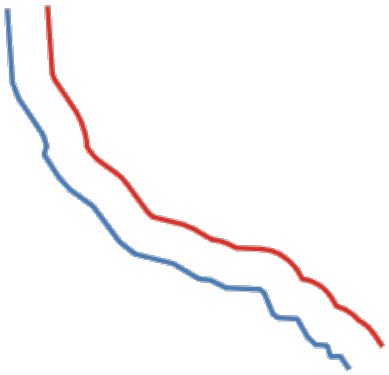


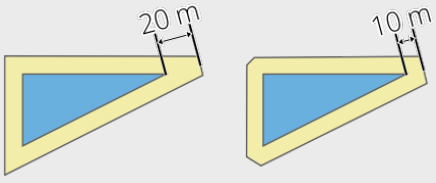
Figura24.125: Em azul a camada de origem, em vermelho a de deslocamento

Permite features in-place modification 1 da linha de características.

Ver também

Array of offset (parallel) lines, *Transladar*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Distância	`DISTANCE`	[numeric: double] Padrão: 10.0	Distância de desvio. Você pode usar o botão de definidor de data à direita para escolher o campo do qual o raio será calculado. Dessa maneira, você pode ter diferentes raios para cada característica (veja *Buffer de distância variável*).
Segmentos	`SEGMENTS`	[numeric: integer] Padrão: 8	Controla o número de segmentos usadas para aproximar um quarte de círculo quando criando desvios redondos.
Join style	`JOIN_STYLE`	[enumeração] Padrão: 0	Especifica se redondo, mitra (estilo de corte) ou chanfrado (estilo de corte) devam ser usados quando refinando os cantos da linha. As opções são: • 0 — Round • 1 — Miter (mitra) • 2 — Bevel (chanfrado) Figura24.126: Estilos de junção redonda, serrada e chanfrada
Miter limit	`MITER_LIMIT`	[numeric: double] Padrão: 2.0	Configura a distância máxima da geometria de deslocamento para usar quando criando uma junção mitered como um fato da distância de deslocamento (somente aplicado para junções do estilo miter). Mínimo: 1.0  Figura24.127: Um buffer de 10m com um limite de buffer de 2 e 10m com um limite de 1.

continua na próxima página

Tabela 24.688 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Offset	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output (offset) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Offset	`OUTPUT`	[vetor: linha]	Saída (desvio) da linha de camada

Código Python

Algorithm ID: `native:offsetline`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Orientação mínima da caixa limitante.

Calcula a área mínima do retângulo rotacionado para cada feição na camada de entrada.

Nota

For singlepart point features, the output corresponds to the bounding box of the geometry.

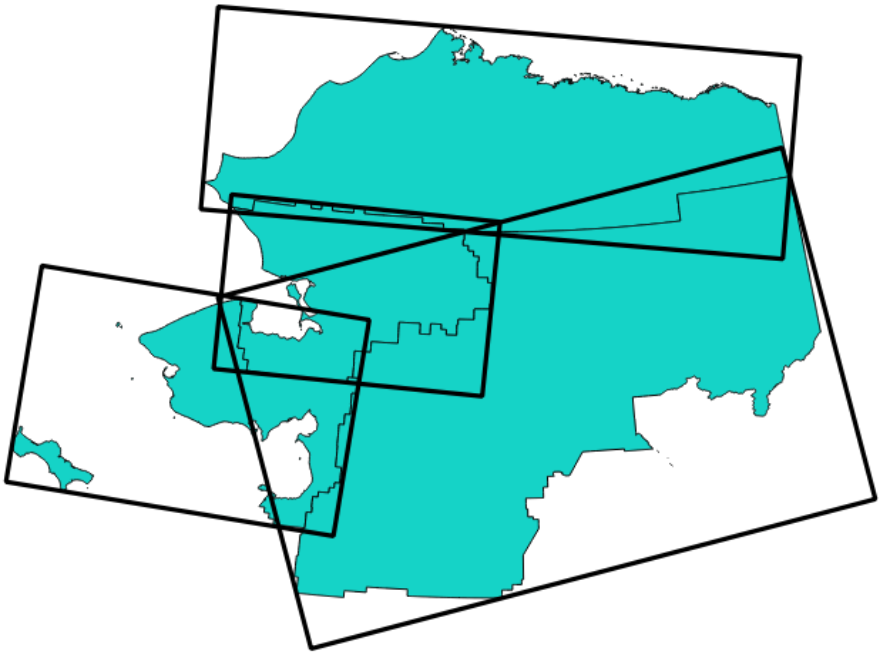


Figura24.128: Orientação mínima da caixa limitante.

Allows features in-place modification 1 das características dos polígonos.

Ver também

Geometria minimamente limitada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Bounding boxes	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Bounding boxes	`OUTPUT`	[vetor: polígono]	The output polygon vector layer. Other than the input attributes, the output layer also contains the following fields: `width`, `height`, `angle`, `area` and `perimeter` of the generated polygon.

Código Python

Algorithm ID: `native:orientedminimumboundingbox`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Ortogonalizar

Tentativas de ortoganalizar as geometrias das linhas de entrada ou camadas poligonais. Esse processo muda os vértices das geometrias para tentar fazer todo ângulo na geometria um ângulo reto ou uma linha reta.

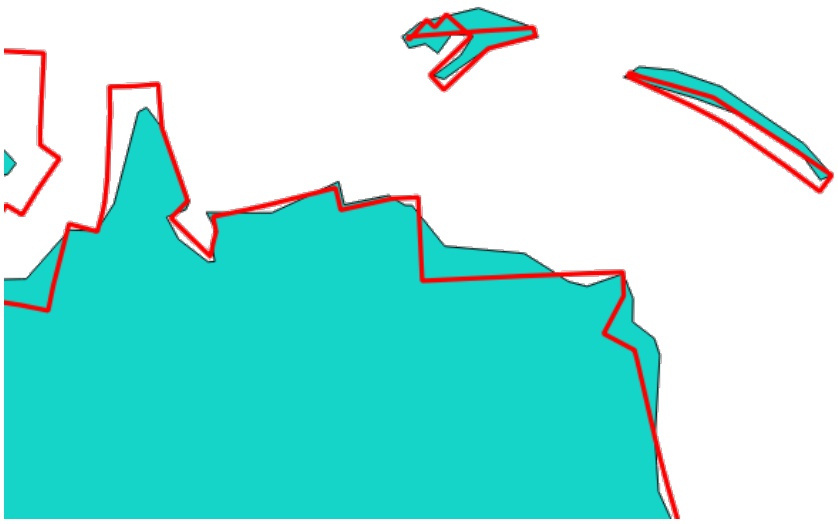


Figura24.129: Em azul a camada de origem e em vermelho o resultado ortogonalizado

permite features in-place modification 1.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Tolerância máxima do ângulo (graus)	`ANGLE_TOLERANCE`	[numeric: double] Default: 15.0	Especifique o desvio máximo de um ângulo reto ou uma linha reta que um vértice pode ter para ser ajustado. Tolerâncias menores significam que somente os vértices que já estão mais próximos dos ângulos retos serão ajustados, e tolerâncias maiores significam que vértices com desvios mais direcionados aos ângulos retos serão ajustados.
Maximum algorithm iterations	`MAX_ITERATIONS`	[numeric: integer] Padrão: 1000	Configurando um número maior para o número máximo de iterações resultará em uma geometria mais ortogonal ao custo de um maior tempo de processamento.
Ortogonalizado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Ortogonalizado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de polígono de saída com ângulos ajustados.

Código Python

Algorithm ID: `native:orthogonalize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Ponto na Superfície

Para cada feição da camada de entrada, retorna um ponto que está garantido na superfície da geometria da feição.

permite features in-place modification 1 de pontos da característica.

Aviso

This algorithm may drop existing primary keys or FID values and regenerate them in output layers, depending on the input parameters.

Ver também

Centroides

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Criar um ponto na superfície para cada parte	`ANGLE_TOLERANCE`	[boolean]	Se checado, um ponto será criado para parte da geometria.
Ponto	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output point vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Ponto	`OUTPUT`	[vetor: ponto]	A camada vetorial de ponto de saída

Código Python

Algorithm ID: `native:pointonsurface`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pontos ao longo da geometria

Cria pontos em intervalos regulares ao longo da linha ou geometrias poligonais. Pontos criados irão ter novos atributos adicionados pela distância ao longo da geometria e do angulo da linha no ponto.

Um começo opcional e fim do desvio pode especificar quais controles o quão longes do começo e o final da geometria dos pontos deverão ser criados.

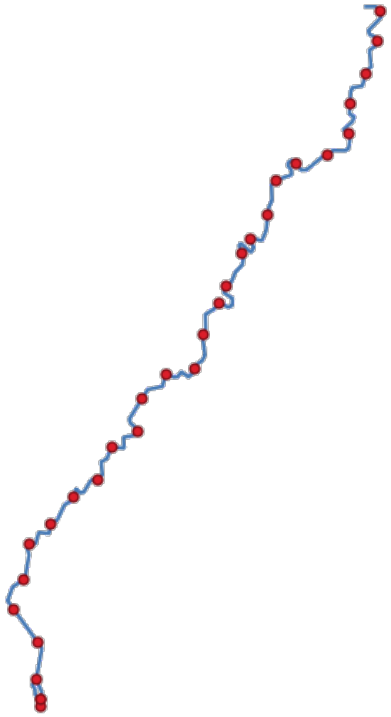


Figura24.130: Pontos criados ao longo da camada da linha de origem

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Interpolar ponto na linha

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Distância	`DISTANCE`	[numeric: double] Padrão: 1,0	Distância entre dois pontos consecutivos ao longo da linha
Start offset	`START_OFFSET`	[numeric: double] Padrão: 0.0	Distância do início da linha de entrada, representando a posição do primeiro ponto.
End offset	`END_OFFSET`	[numeric: double] Padrão: 0.0	Distância do final da linha de entrada, representando a posição além da qual nenhuma feição de ponto deve ser criada.
Pontos interpolados	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos interpolados	`OUTPUT`	[vetor: ponto]	Point vector layer with features placed along lines or polygon boundaries of the input layer. Other than the input attributes on every related point, the output layer also contains the following fields: `distance` and `angle` of the generated point.

Código Python

Algorithm ID: `native:pointsalonglines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Deslocamento de pontos

Dada uma distância de proximidade, identifica feições de pontos próximos e as distribui radialmente sobre um círculo cujo centro representa seu baricentro. Uma ferramenta conveniente para espalhar feições sobrepostas.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Distância mínima para outros pontos	`PROXIMITY`	[numeric: double] Padrão: 1,0	Distância entre cada ponto das características são considerados próximos. Características próxiumas são distribuídas completamente.
Distância de deslocamento	`DISTANCE`	[numeric: double] Padrão: 1,0	Raio do círculo no qual as feições próximas são colocadas
Distribuição horizontal para caso de dois pontos	`HORIZONTAL`	[boleano] Padrão: Falso	Quando apenas dois pontos são identificados como próximos, alinha-os horizontalmente no círculo em vez de verticalmente.
Deslocado	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Deslocado	`OUTPUT`	[vetor: ponto]	Camada vetorial de ponto de saída

Código Python

Algorithm ID: `qgis:pointsdisplacement`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pólo de inacessibilidade

Calcula o pole de inacessibilidade de uma camada poligonal, a qual é a mais distante internamente do ponto do limite da superfície.

Esse algoritmo usa o algoritmo "polylabel" (Vladimir Agafonkin, 2016) o qual é uma abordagem iterativa garantida de achar o polo verdadeiro de inacessibilidade dentro de uma tolerância especificada. Uma tolerância mais precisa (valor baixo) requere mais iterações e levara mais tempo para calcular.

Figura24.131: Pólo de inacessibilidade

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	Camada vetorial de entrada
Tolerância	`TOLERANCE`	[numeric: double] Padrão: 1,0	Estabelecer a tolerância para o cálculo
Ponto	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Ponto	`OUTPUT`	[vetor: ponto]	The output point vector layer. It will contain a new attribute with the distance `dist_pole` from the calculated pole to the polygon boundary.

Código Python

Algorithm ID: `native:poleofinaccessibility`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Poligonizar

Cria uma camada de polígono cujos limites de feições são gerados a partir de uma camada de linha de feições **fechadas**.

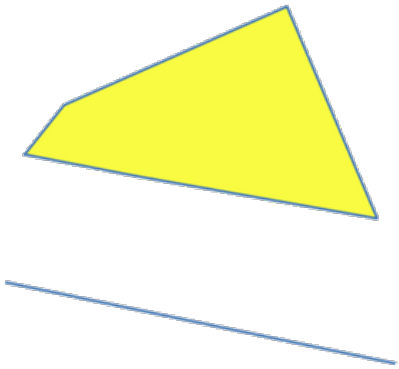


Figura24.132: Os polígonos amarelos gerados a partir das linhas fechadas

Nota

A camada de linha deve ter formas fechadas para ser transformada em um polígono.

Ver também

Polígonos para linhas, *Linhas para polígonos*, *Converter tipo de geometria*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Manter campos da camada de entrada	`KEEP_FIELDS`	[boleano] Padrão: Falso	Check para manter os campos (somente a tabela de estrutura, não os valores) da camada de entrada.
Polígonos de linhas	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output polygon vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Polígonos de linhas	`OUTPUT`	[vetor: polígono]	A camada vetorial de polígono de saída das linhas

Código Python

Algorithm ID: `native:polygonize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Polígonos para linhas

Pega uma camada de polígono e cria uma camada de linha, com linhas representando os limites dos polígonos na camada de entrada.

A tabela de atributos da camada de saída é a mesma da camada de entrada.

Figura24.133: Linhas pretas como resultado do algoritmo

Default menu: *Vector* ► *Geometry Tools*.

Ver também

Linhas para polígonos, *Poligonizar*, *Converter tipo de geometria*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	insira a camada vetorial de polígonal.
Linhas	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Linhas	`OUTPUT`	[vetor: linha]	A camada vetorial de linha de saída de polígonos

Código Python

Algorithm ID: `native:polygonstolines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Projetar pontos (Cartesiano)

Projeta geometrias de pontos por uma distância e direção especificadas (azimute).

permite features in-place modification 1 de pontos da característica.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Bearing (degrees from North)	`BEARING`	[numeric: double] Padrão: 0.0	Ângulo no sentido horário a partir do Norte, em unidade de grau (°)
Distância	`DISTANCE`	[numeric: double] Padrão: 1,0	Distância das geometrias dos desvios, in unidades por camada
Projetado	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output point vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Projetado	`OUTPUT`	[vetor: ponto]	A camada vetorial de ponto de saída (projetada)

Código Python

Algorithm ID: `native:projectpointcartesian`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Promover para multipartes

Pega uma camada vetorial com geometrias de uma única parte e gera uma nova na qual todas as geometrias são multipartes.

As feições de entrada que já são feições multipartes permanecerão inalteradas.

Este algoritmo pode ser usado para forçar geometrias para tipos multipartes para serem compatíveis com provedores de dados que requerem feições multipartes.

Allows features in-place modification 1 de ponto, linha e características poligonais.

Ver também

Agregar, *Coletar geometrias*, *Convert to strictly multipart*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Multipartes	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output multipart vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Multipartes	`OUTPUT`	[o mesmo que entrada]	A camada vetorial multipartes de saída

Código Python

Algorithm ID: `native:promotetomulti`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Retângulos, ovais, diamantes

Cria uma área de buffer com uma forma retangular, oval ou diamante para cada feição da camada de ponto de entrada.

Os parâmetros de forma podem ser corrigidos para todas as feições ou dinâmicos usando um campo ou uma expressão.

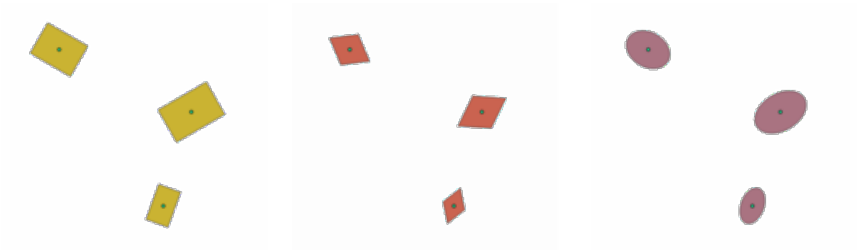


Figura24.134: Diferentes formas de buffer com parâmetros dinâmicos

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Buffer shape	`SHAPE`	[enumeração]	A forma a ser usada. Uma de: • 0 — Retângulos • 1 — Ovais • 2 — Diamantes
Largura	`WIDTH`	[numeric: double] Padrão: 1,0	Largura da forma do buffer
Altura	`HEIGHT`	[numeric: double] Padrão: 1,0	Altura da forma do buffer

continua na próxima página

Tabela 24.710 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Rotação Opcional	`ROTATION`	[numeric: double] Padrão: 0.0	Rotação da forma do buffer
Number of segments	`SEGMENTS`	[numeric: integer] Padrão: 36	Número de segmentos para um círculo completo (forma *Ovais*)
Saída	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Saída	`OUTPUT`	[vetor: polígono]	A camada vetorizada de saída (com shapes buffer)

Código Python

Algorithm ID: `native:rectanglesovalsdiamonds`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Remover vértices duplicados

Remove os vértices duplicados das feições, sempre que a remoção dos vértices não resultar em uma geometria degenerada.

O parâmetro de tolerância especifica a tolerância para coordenadas ao determinar se os vértices são idênticos.

Por padrão, valores em Z não são considerados quando detectando vértices duplicados, por exemplo, dois vértices com o mesmo X e Y de coordenada, mas diferentes valores para Z ainda serão considerados duplicados e um será removido. Se a *Use Z Value* for um parâmetro verdadeiro, então os valores para Z também são testados e os vértices com o mesmo valor para X e Y, mas diferentes para Z serão mantidos.

Allows features in-place modification 1 de ponto, linha e características poligonais.

Nota

Vértices duplicados não são testados entre diferentes partes de uma geometria multipartes, por exemplo, uma geometria multiponto com pontos sobrepostos não será alterada por este método.

Ver também

Extrair vértices, *Extrair vértices específicos*, *Excluir geometrias duplicadas*, *Delete duplicated vertices*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Tolerância	`TOLERANCE`	[numeric: double] Padrão: 0.000001	Vértices mais próximos do que a distância especificada são considerados duplicados
Usar valor Z	`USE_Z_VALUE`	[boolean] Padrão: Falso	Se o *Use Z Value* parâmetro for verdadeiro, então os valores para Z serão testados e os vértices com mesmo valor de X e Y mas diferentes para Z serão mantidos.
Limpo	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Limpo	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída (sem vértices duplicados)

Código Python

Algorithm ID: `native:removeduplicatevertices`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Remover geometrias nulas

Remove quaisquer feições que não tenham uma geometria de uma camada vetorial. Todas as outras feições serão copiadas inalteradas.

As feições com geometrias nulas podem ser salvas em uma camada separada.

Se guilabel:*Also remove empty geometries* é marcado, o algoritmo remove as características das geometrias as quais não tem atributos, leia-se, geometrias que são vazias. Nesse caso, os valores nulos de saída refletirão essa opção, contendo ambos valores nulos e vazios para as geometrias.

Ver também

Excluir geometrias duplicadas

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada (com geometrias não-NULAS).
Remover também geometrias vazias	`REMOVE_EMPTY`	[boleano]	
Geometrias não nulas	`OUTPUT` Opcional	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the non-NULL (and non-empty) geometries. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Geometrias nulas Opcional	`NULL_OUTPUT`	[o mesmo que entrada] Default: `[Pular saída]`	Specify the output vector layer for the NULL (and empty) geometries. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Geometrias nulas	`NULL_OUTPUT`	[o mesmo que entrada]	Camada vetorial de saída (para NULA e, se escolhida, geometrias vazias)
Geometrias não nulas	`OUTPUT`	[o mesmo que entrada]	Camada vetorial de saída (sem NULA e, se escolhida, geometrias vazias).

Código Python

Algorithm ID: `native:removenullgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Direção reversa da linha

Inverte a direção de uma camada de linha.

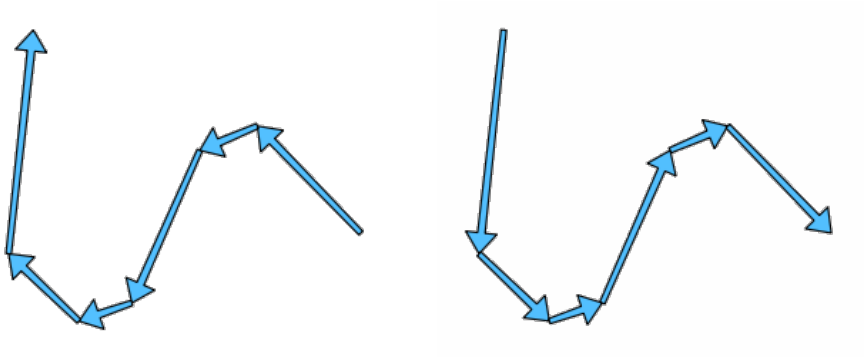


Figura24.135: Antes e depois da inversão de direção

Permite features in-place modification 1 da linha de características.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.

continua na próxima página

Tabela 24.716 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Invertido	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Invertido	`OUTPUT`	[vetor: linha]	A camada vetorial de linha de saída (com linhas invertidas)

Código Python

Algorithm ID: `native:reverselinedirection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Rotacionar

Rotaciona geometrias de recurso por meio angulo de relógio especifico. A rotação ocorre em volta de cada característica do centroide, ou opcionalmente em volta de um ponto único' pré-selecionado

Allows features in-place modification 1 de ponto, linha e características poligonais.

Ver também

Transladar, *Trocar as coordenadas X e Y*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Rotação (em graus no sentido horário)	`ANGLE`	[numeric: double] Padrão: 0.0	Ângulo de rotação em graus
Rotation anchor point (x, y) Opcional	`ANCHOR`	[coordenadas] Padrão: Não definido	Coordenadas X,Y do ponto para rotacionar as feições ao redor. Se não estiver definido, a rotação ocorre em torno do centroide de cada feição.
Rotacionado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer (with rotated geometries). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Rotacionado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída com geometrias rotacionadas

Código Python

Algorithm ID: `native:rotatefeatures`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Grau de arredondamento

Calculates the roundness of each feature and stores it as a new field `roundness`. The input vector layer must contain polygons.

O grau de arredondamento de um polígono é definido como $4\pi \times$ área do polígono / perímetro^2. O valor do grau de arredondamento vária entre 0 e 1. Um círculo perfeito tem um grau de arredondamento de 1, enquanto que um polígono completamente plano tem um grau de arredondamento de 0.

Nota

O algoritmo retorna NULL para características de polígono multipartidárias.

Allows features in-place modification 1 das características dos polígonos.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	Camada vetorial de entrada
Grau de arredondamento	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output vector layer (with roundness field). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Grau de arredondamento	`OUTPUT`	[o mesmo que entrada]	The output vector layer with `roundness` value in a field

Código Python

Algorithm ID: `nativo: grau de arredondamento`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Segmentar por ângulo máximo

Segmenta uma geometria convertendo seções curvas em seções lineares.

A segmentação é performada por meio da especificação máxima permitida do angulo do raio entre os vértices da geometria endireitada (por exemplo, o angulo do arco criado a partir do centro do arco original para saídas dos vértices consecutivos na geometria linearizada). Geometrias não curvas serão mantidas sem mudanças.

Ver também

Segmentar por distância máxima, *Simplificar*, *Suave*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Ângulo máximo entre vértices (graus)	`ANGLE`	[numeric: double] Padrão: 5.0	Ângulos máximos permitidos entre os vértices endireitados das geometrias.
Segmentado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer (with segmentized geometries). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Segmentado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída com geometrias segmentadas

Código Python

Algorithm ID: `native:segmentizebymaxangle`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Segmentar por distância máxima

Segmenta uma geometria convertendo seções curvas em seções lineares.

A segmentação é realizada especificando a distância de deslocamento máxima permitida entre a curva original e a representação segmentada. Geometrias não curvas serão mantidas sem alterações.

Ver também

Segmentar por ângulo máximo, *Simplificar*, *Suave*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Maximum offset distance	`DISTANCE`	[numeric: double] Padrão: 1,0	Distância máxima de deslocamento permitida entre a curva original e a representação segmentada, nas unidades da camada.
Segmentado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer (with segmentized geometries). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Segmentado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída com geometrias segmentadas

Código Python

Algorithm ID: `native:segmentizebymaxdistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Definir valor M

Define o valor M para geometrias em uma camada.

Se já existirem valores M na camada, eles serão substituídos pelo novo valor. Se nenhum valor M existir, a geometria será atualizada para incluir valores M e o valor especificado usado como o valor M inicial para todas as geometrias.

Permite *features in-place modification* do ponto, linha, e características poligonais com o M ativado,

Dica

Use o botão de

to para confirmar o valor de M adicionado: os resultados estarão disponíveis na caixa de diálogo *Identify Results*.

Ver também

Definir o valor M da matriz, *Definir valor Z*, *Diminua os valores de M/Z*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Valor M	`M_VALUE`	[numeric: double] Padrão: 0.0	Valor M para atribuir às geometrias da feição
M Adicionado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
M Adicionado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída (com valores M atribuídos às geometrias)

Código Python

Algorithm ID: `native:setmvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Definir o valor M da matriz

Usa valores sampleados de um banda dentro de uma camada raster para configurar o valor M para cada vértice sobreposto na geometria da característica. Os valores de raster podem opcionalmente serem escalados por uma certa quantiade de pré-sets.

Se valores de M já existem na camada, eles serão sobrescritos com um novo valor, Se valores de não M existem, a geometria será atualizada para incluir os valores de M.

Permite *features in-place modification* do ponto, linha, e características poligonais com o M ativado,

Ver também

Drape (configure o valor Z do raster), *Definir valor M*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Camada raster	`RASTER`	[raster]	Camada matricial com valores M
Número da banda	`BAND`	[banda raster] Padrão: 1	A banda raster da qual os valores M são retirados
Value for No-Data or non-intersecting vertices	`NODATA`	[numeric: double] Padrão: 0.0	Valor para ser usado em caso do vértice não faça intersecção (um pixel válido) do raster.
Scale factor	`SCALE`	[numeric: double] Padrão: 1,0	Valor de escala: os valores da banda são multiplicados por este valor.
Offset	`OFFSET`	[numeric: double] Padrão: 0.0	Valor de desvio: é adicionado algebricamente para os valores de banda depois de aplicada o “Fator escala”

continua na próxima página

Tabela 24.728 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Atualizado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer (with updated M values). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Atualizado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída (com valores M atualizados)

Código Python

Algorithm ID: `native:setmfromraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Definir valor Z

Definir o valor Z para geometrias em uma camada.

Se os valores de Z existem na camada, eles serão sobrescritos com um novo valor, Se valores de não Z existem, a geometria será atualizada para incluir os valores de z e o valor especificado usado como valor Z inicial para as geometrias.

permite features in-place modification 1 de um ponot, linha, e característica polígonais com Z habilitado.

Dica

Use o botão

para chegar o valor adicionado de Z> os resultados são disponibilizados na caixa de diálogo *Identify Results*

Ver também

Drape (configure o valor Z do raster), *Definir valor M*, *Diminua os valores de M/Z*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Valor Z	`Z_VALUE`	[numeric: double] Padrão: 0.0	Valor Z para atribuir às geometrias da feição
Z Adicionado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Z Adicionado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída (com valores Z atribuídos)

Código Python

Algorithm ID: `native:setzvalue`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Simplificar

Simplifique as geometrias em uma linha ou camada poligonal. Cria uma nova camada com as mesmas características como aquelas na camada inserida, mas todas as geometrias contendo um número menor de vértices.

O algoritmo dá a escolha de métodos de simplificação, incluindo a distância base (o algoritmo “Douglas-Peucker”) áreas baseadas (algoritmos “Visvalingam”) e geometrias encaixadas na grade.

Figura24.136: From left to right, source layer and increasing simplification tolerances

permite features in-place modification 1.

Default menu: *Vector ► Geometry Tools*.

Ver também

Simplify coverage, *Suave*, *Densificar por contagem.*, *Densificar por intervalo*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Método de simplificação	`METHOD`	[enumeração] Padrão: 0	Método de simplificação. Um de: • 0 — Distance (Douglas-Peucker) • 1 — Snap to grid • 2 — Area (Visvalingam)
Tolerância	`TOLERANCE`	[numeric: double] Padrão: 1,0	Tolerância de limite (em unidades da camada): Se a distância entre dois nós é menor que o valor de tolerância, o segmento será simplificado e os vértices serão removidos.
Simplificado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output (simplified) vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Simplificado	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída (simplificada)

Código Python

Algorithm ID: `native:simplifygeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Buffer de lado único

Calcula um buffer em linhas por uma distância especificada em apenas um lado da linha.

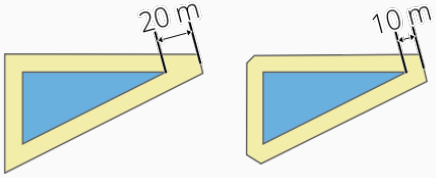
Buffer sempre resulta em uma camada de polígono.

Figura24.137: Buffer do lado esquerdo versus direito na mesma camada de linha vetorial

Ver também

Buffer

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Distância	`DISTANCE`	[numeric: double] Padrão: 10.0	Distância do buffer
Lado	`SIDE`	[enumeração] Padrão: 0	De que lado criar o buffer. Um de: • 0 – Esquerda • 1 – Direita
Segmentos	`SEGMENTS`	[numeric: integer] Padrão: 8	Controla o número de segmentos usadas para aproximar um quarte de círculo quando criando desvios redondos.
Join style	`JOIN_STYLE`	[enumeração] Padrão: 0	Especifica se redondo, mitra (estilo de corte) ou chanfrado (estilo de corte) devam ser usados quando refinando os cantos da linha. As opções são: • 0 — Round • 1 — Miter (mitra) • 2 — Bevel (chanfrado) Figura24.138: Estilos de junção redonda, serrada e chanfrada
Miter limit	`MITER_LIMIT`	[numeric: double] Padrão: 2.0	Configura a distância máxima da geometria de deslocamento para usar quando criando uma junção mitered como um fato da distância de deslocamento (somente aplicado para junções do estilo miter). Mínimo: 1.0  Figura24.139: Um buffer de 10m com um limite de buffer de 2 e 10m com um limite de 1.
Buffer	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output (buffer) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Buffer	`OUTPUT`	[vetor: polígono]	Saída (tampão) camada poligonal.

Código Python

Algorithm ID: `native:singlesidedbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Suave

Suaviza as geometrias em uma linha ou camada poligonal por meio da adição de mais **vertices and corners** (vértices e cantos) para as características das geometrias.

O parâmetro de iterações dita quantas iterações de suavização serão aplicadas para cada geometria. Um número maior de iterações resultará em geometrias mais suaves, e o número de nós crescerá na mesma medida na geometrias.

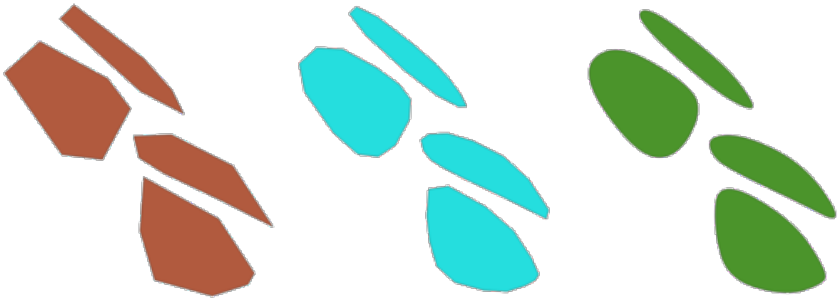


Figura24.140: Um número crescente de iterações terá como resultado geometrias mais suaves.

O parâmetro de desvio controla o quão “apertado” as geometrias suavizadas seguem as geometrias originais. Valores menores resultarão em valores de ajuste mais apertados, e valores maiores criarão um ajuste mais frouxo.

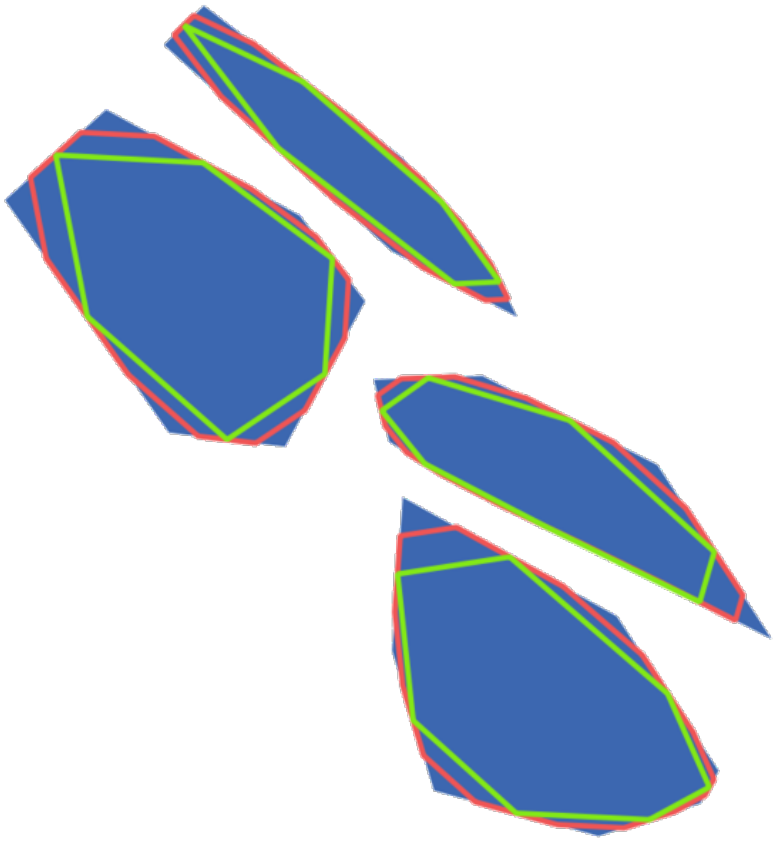


Figura24.141: Azul: A camada de entrada. Desvio de 0.25 devolve a linha vermelha, enquanto que o desvio de 0.50 devolve a linha verde.

O parâmetro de ângulo máximo pode ser usado para evitar a suavização de nós com grandes ângulos. Qualquer nó em que o ângulo dos segmentos para qualquer um dos lados for maior do que isso não será suavizado. Por exemplo, definir o ângulo máximo para 90 graus ou menos preservaria os ângulos retos na geometria.

permite features in-place modification 1.

Ver também

Simplificar, *Simplify coverage*, *Densificar por contagem.*, *Densificar por intervalo*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Entre com uma linha ou camada poligonal vetorizada.
Iterações	`ITERATIONS`	[numeric: integer] Padrão: 1	Aumentar o número de iterações resultará em geometrias mais suaves (e mais vértices).
Offset	`OFFSET`	[numeric: double] Padrão: 0.25	Valores crescentes moverão as linhas suavizadas/ fronteiras para mais longe das linhas de entrada/ fronteiras.

continua na próxima página

Tabela 24.736 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Ângulo máximo do nó para suavizar	`MAX_ANGLE`	[numeric: double] Padrão: 180.0	Cada nó abaixo deste valor será suavizado
Suavizado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output (smoothed) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Suavizado	`OUTPUT`	[o mesmo que entrada]	Camada vetorial de saída (suavizada).

Código Python

Algorithm ID: `native:smoothgeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Ajustar as geometrias para a camada.

Ajuste as geometrias para uma camada ao invés para as geometrias de uma outra camada, ou para as geometrias dentro da mesma camada.

A correspondência é feita baseada na distância de tolerância, e os vértices serão inseridos ou removidos conforme forem requeridos para fazer as geometrias corresponderem com as geometrias de referência.

Allows features in-place modification 1 de ponto, linha e características poligonais.

Ver também

Aderir pontos na grade

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Camada de referência	`REFERENCE_LAYER`	[vector: geometry]	Camada vetorial para encaixar.
Tolerância	`TOLERANCE`	[numeric: double] Padrão: 10.0	Controle o quão próximos os vértices de entrada precisam estar das geometrias da camada de referência antes de serem ajustados.

continua na próxima página

Tabela 24.738 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Comportamento	BEHAVIOR	[enumeração] Padrão: 0	O encaixe pode ser feito em um nó ou segmento existente (seu ponto mais próximo ao vértice a ser movido). Opções de encaixe disponíveis: • 0 — Prefira alinhar nós, inserir vértices extras quando requerido. Prefira encaixar nós, mesmo quando o segmento estiver mais perto que um nó. Nós novos serão inseridos para fazer as geometrias seguirem cada uma quando dentro de uma tolerância aceitável • 1 — prefira pontos próximos, insira vértices extras quando requerido. Encaixe para pontos mais próximos, independentemente de ser um nó ou segmento. Novos nós serão inseridos para fazer geometrias se seguirem exatamente quando dentro de uma tolerância permitida. • 2 — Prefira nós de alinhamento, não insira vértices novos. Prefira ajustar aos nós, mesmo quando o segmento estiver mais perto do nó. Nós novos não serão inseridos. • 3 — Prefira pontos próximos, não insira novos vértices. Encaixe para o ponto mais próximo, independentemente de ser um nó ou segmento. Nenhum nó novo será inserido. • 4 — Mova apenas pontos finais, prefira nós de alinhamento. Somente pontos de linha de encaixe de começo/fim (características chave também serão encaixadas, características poligonais não serão modificadas) prefira encaixar aos nós. • 5 — Mova apenas os pontos finais, prefira o ponto mais próximo Apenas encaixe os pontos inicial/final das linhas (as características do ponto também serão ajustadas, as características do polígono não serão modificadas), encaixe no ponto mais próximo. • 6 — Ajustar pontos finais apenas aos pontos finais. Only snap the start/end points of lines to other start/end points of lines • 7 — Encaixe aos nós ancoras (somente camada única)

continua na próxima página

Tabela 24.738 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Snapped geometry	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output (snapped) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Snapped geometry	`OUTPUT`	[o mesmo que entrada]	Output (snapped) vector layer

Código Python

Algorithm ID: `native:snapgeometries`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Aderir pontos na grade

Modifica as coordenadas das geometrias em uma camada vetorial, para que todos os pontos ou vértices sejam encaixados no ponto mais próximo de uma grade.

If the snapped geometry cannot be calculated (or is totally collapsed) the feature's geometry will be cleared.

O encaixe pode ser executado nos eixos X, Y, Z ou M. Um espaçamento de grade de 0 para qualquer eixo desabilitará o encaixe para esse eixo.

Allows features in-place modification 1 de ponto, linha e características poligonais.

Aviso

Ajustar à grade pode gerar uma geometria inválida em alguns casos de canto.

Ver também

Ajustar as geometrias para a camada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Espaçamento da grade em X	`HSPACING`	[numeric: double] Padrão: 1,0	Espaçamento da grade no eixo X
Espaçamento da grade em Y	`VSPACING`	[numeric: double] Padrão: 1,0	Espaçamento da grade no eixo Y
Espaçamento da grade em Z	`ZSPACING`	[numeric: double] Padrão: 0.0	Espaçamento da grade no eixo Z
Espaçamento da grade em M	`MSPACING`	[numeric: double] Padrão: 0.0	Espaçamento da grade no eixo M
Snapped	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output (snapped) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Snapped	`OUTPUT`	[o mesmo que entrada]	Output (snapped) vector layer

Código Python

Algorithm ID: `native:snappointstogrid`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Dividir linhas por comprimento máximo.

Pega uma camada de linha (ou curva) e divide cada feição em várias partes, onde cada parte tem um comprimento máximo especificado. Os valores Z e M no início e no final das novas sub-cadeias de linha são interpolados linearmente a partir dos valores existentes.

Permite features in-place modification 1 da linha de características.

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	A entrada da camada de linha vetorial.
Comprimento máximo da linha	`LENGTH`	[numeric: double] Padrão: 10.0	O comprimento máximo de uma linha na saída.
Dividir	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Dividir	`OUTPUT`	[vetor: linha]	The new line vector layer. The length of the feature geometries is less than or equal to the length specified in the LENGTH parameter. An additional field containing the `order` will be added.

Código Python

Algorithm ID: `native:splitlinesbylength`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Subdividir

Subdivide a geometria. A geometria retornada será uma coleção contendo partes subdivididas da geometria original, onde nenhuma parte possui mais do que o número máximo especificado de nós.

Isso é útil para dividir uma geometria complexa em partes menos complexas, mais fáceis de indexar espacialmente e mais rápidas para realizar operações espaciais. As geometrias curvas serão segmentadas antes da subdivisão.

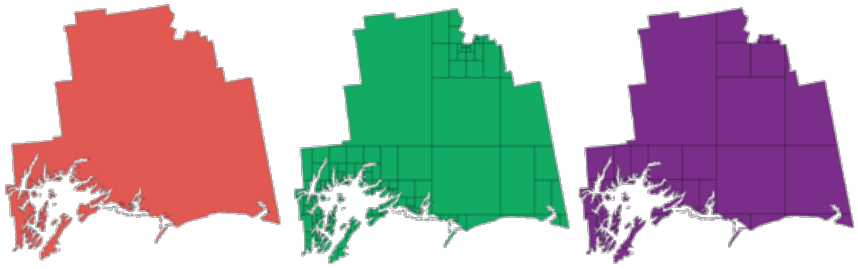


Figura24.142: À esquerda da camada de entrada, o valor máximo médio dos nós é 100 e o valor máximo direito é 200

Allows features in-place modification 1 de ponto, linha e características poligonais.

Nota

A subdivisão de uma geometria pode gerar partes de geometria que podem não ser válidas e podem conter autointerseções.

Ver também

Explodir linhas, *Linha de substring.*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	A camada vetorial de entrada
Maximum nodes in parts	`MAX_NODES`	[numeric: integer] Padrão: 256	Número máximo de vértices que cada nova peça de geometria é permitida ter. Menos *subpartes* para valores mais altos.

continua na próxima página

Tabela 24.744 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Subdividido	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output (subdivided) vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Subdividido	`OUTPUT`	[o mesmo que entrada]	Camada de vetor de saída

Código Python

Algorithm ID: `native:subdivide`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Trocar as coordenadas X e Y

Altera os valores das coordenadas X e Y nas geometrias de entrada.

Ele pode ser usado para reparar geometrias que acidentalmente tiveram seus valores de latitude e longitude invertidos.

Allows features in-place modification 1 de ponto, linha e características poligonais.

Ver também

Transladar, *Rotacionar*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	A camada vetorial de entrada
Trocado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Trocado	`OUTPUT`	[o mesmo que entrada]	Camada vetorial de saída (trocada).

Código Python

Algorithm ID: `native:swapxy`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Buffers cônicos.

Cria um buffer cônico ao longo de geometrias de linha, usando um diâmetro de buffer inicial e final especificado.

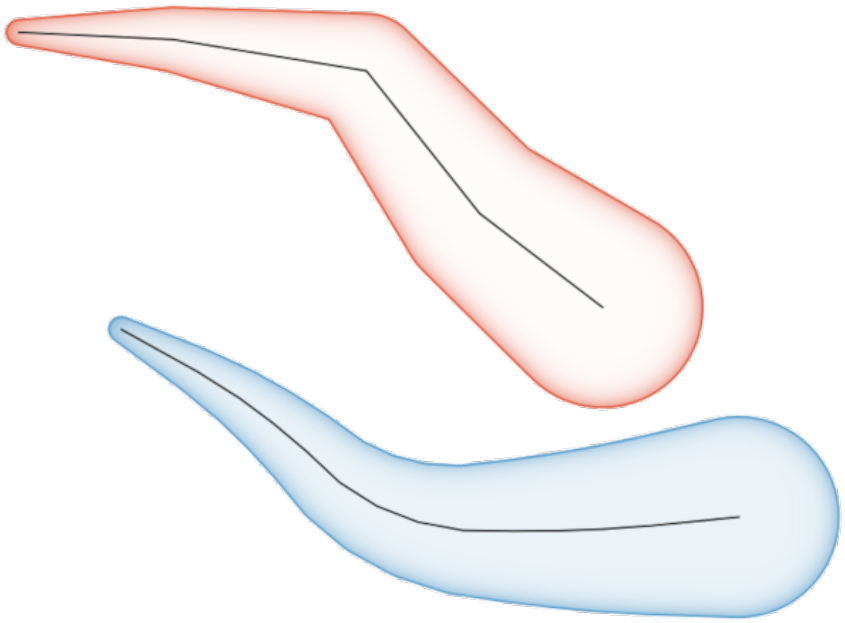


Figura24.143: Tapered buffer example

Ver também

Buffer de largura variável (por valor M), *Buffer*, *Crie buffers de cunha (wedge buffer)*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Largura inicial	`START_WIDTH`	[numeric: double] Padrão: 0.0	Representa o raio do buffer aplicado no ponto inicial da feição de linha
Largura final	`END_WIDTH`	[numeric: double] Padrão: 0.0	Representa o raio do buffer aplicado no ponto final do recurso de linha.
Segmentos	`SEGMENTS`	[numeric: integer] Padrão: 16	Controla o número de segmentos usadas para aproximar um quarte de círculo quando criando desvios redondos.

continua na próxima página

Tabela 24.748 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Buffered	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output (buffer) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Buffered	`OUTPUT`	[vetor: polígono]	Saída (tampão) camada poligonal.

Código Python

Algorithm ID: `native:taperedbuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Tessellate (mosaico)

Forma uma camada de geometria poligonal, dividindo as geometrias em componentes triangulares.

A camada de saída consiste em geometrias de multipolígonos para cada feição de entrada, com cada multipolígono consistindo em vários polígonos de componentes de triângulo.

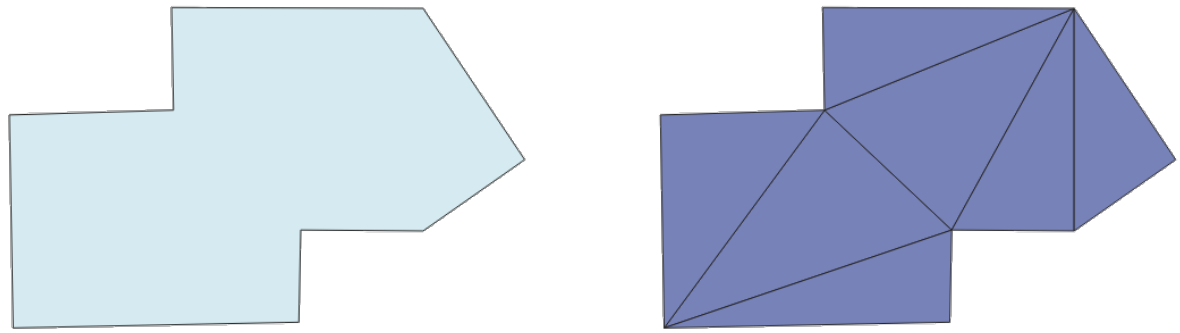


Figura24.144: Polígono em mosaico (direita)

Allows features in-place modification 1 das características dos polígonos.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: polígono]	insira a camada vetorial de polígonal.
Tesselated	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Tesselated	`OUTPUT`	[vetor: polígono]	Camada de saída multipoligonalZ

Código Python

Algorithm ID: `3d:tessellate`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Transecto

Creates transects on vertices for (multi)linestring.

A transect is a line oriented from an angle (by default perpendicular) to the input polylines (at vertices).

O(s) campo(s) da(s) feição(ões) são retornados na transecção com estes novos campos:

- TR_FID: ID do recurso original
- TR_ID: ID da transecção. Cada transecção tem um ID único.
- TR_SEGMENT: ID do segmento da linestring.
- TR_ANGLE: Ângulo em graus a partir da linha original no vértice
- TR_LENGTH: Comprimento total da transecção retornada
- TR_ORIENT: Lado da transecção (apenas à esquerda ou à direita da linha, ou ambos os lados)

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

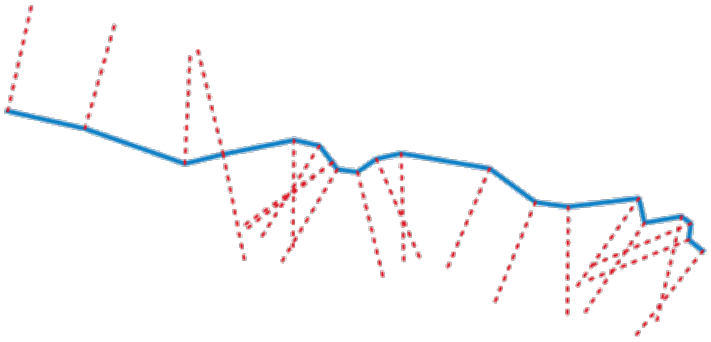


Figura24.145: Linhas vermelhas tracejadas representam a transecção da camada de linha de entrada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Comprimento do transecto	`LENGTH`	[numeric: double] Padrão: 5.0	Comprimento em unidade de mapa da transecção.
Ângulo em graus da linha original nos vértices	`ANGLE`	[numeric: double] Padrão: 90.0	Alterar o ângulo do transecto
Side to create the transect	`SIDE`	[enumeração]	Escolha o lado do transecto. As opções disponíveis são: • 0 — Esquerda • 1 – Direita • 2 — Ambos
Transect	`OUTPUT`	[vetor: linha] Padrão: `[Criar camada temporária]`	Specify the output line layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Transect	`OUTPUT`	[vetor: linha]	Saída da camada de linha

Código Python

Algorithm ID: `native:transect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Transladar

Move as geometrias dentro de uma camada, compensando com um deslocamento X e Y predefinido

Os valores Z e M presentes na geometria também podem ser traduzidos.

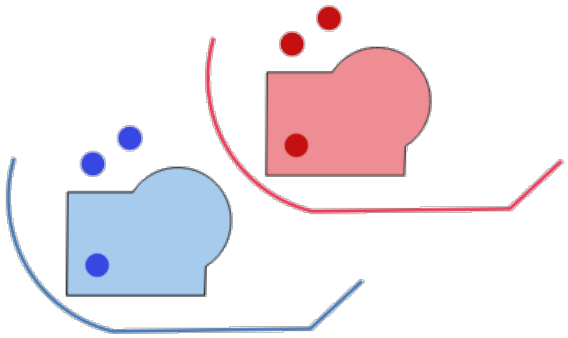


Figura24.146: Features in red represent the translated geometry of the input (blue) layer

Allows features in-place modification 1 de ponto, linha e características poligonais.

Ver também

Array of translated features, *Linhas de desvio.*, *Rotacionar*, *Trocar as coordenadas X e Y*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Offset distance (x-axis)	`DELTA_X`	[numeric: double] Padrão: 0.0	Delocamento a ser aplicado ao eixo X
Offset distance (y-axis)	`DELTA_Y`	[numeric: double] Padrão: 0.0	Delocamento a ser aplicado ao eixo Y
Offset distance (z-axis)	`DELTA_Z`	[numeric: double] Padrão: 0.0	Displacement to apply on the Z axis
Offset distance (m values)	`DELTA_M`	[numeric: double] Padrão: 0.0	Displacement to apply on the M axis
Transladado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Transladado	`OUTPUT`	[o mesmo que entrada]	Camada de vetor de saída

Código Python

Algorithm ID: `native:translategeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Buffer de largura variável (por valor M)

Creates variable width buffers along lines, using the M value of the line geometries as the diameter of the buffer at each vertex.

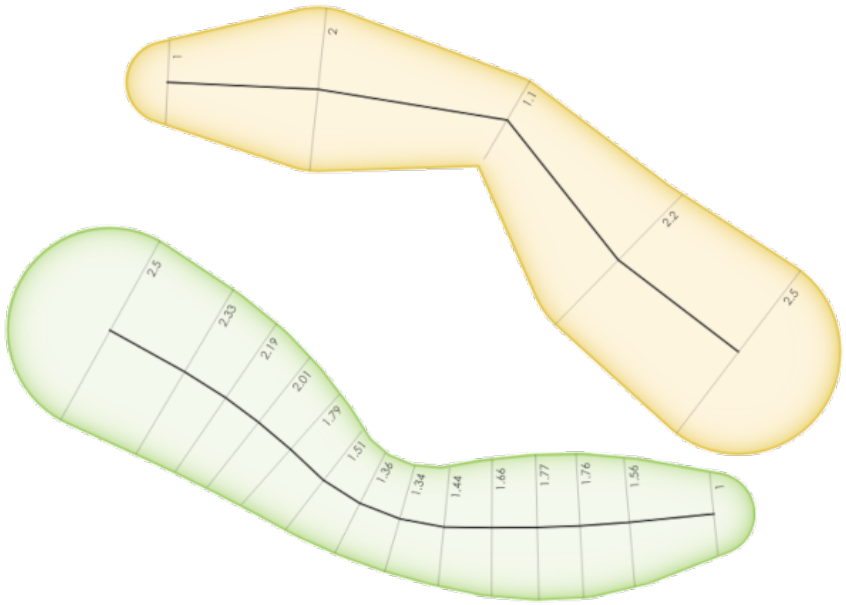


Figura24.147: Exemplo de buffer variável

Ver também

Buffers cônicos., *Buffer*, *Definir valor M*, *Buffer de distância variável*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Insira a linha de camada vetorial.
Segmentos	`SEGMENTS`	[numeric: integer] Padrão: 16	Número de segmentos de buffer por quarto de círculo. Pode ser um valor único (mesmo valor para todos os recursos) ou pode ser obtido dos dados dos recursos (o valor pode depender dos atributos do recurso).
Buffered	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output (buffer) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Buffered	`OUTPUT`	[vetor: polígono]	Variable buffer polygon layer

Código Python

Algorithm ID: `native:bufferbym`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Polígonos de Voronoi

cria uma camada de pontos e gera uma camada de polígonos contendo os polígonos de Voronoi (também conhecidos como polígonos de Thiessen) correspondentes a esses pontos de entrada.

Qualquer local dentro de um polígono de Voronoi está mais próximo do ponto associado do que de qualquer outro ponto.

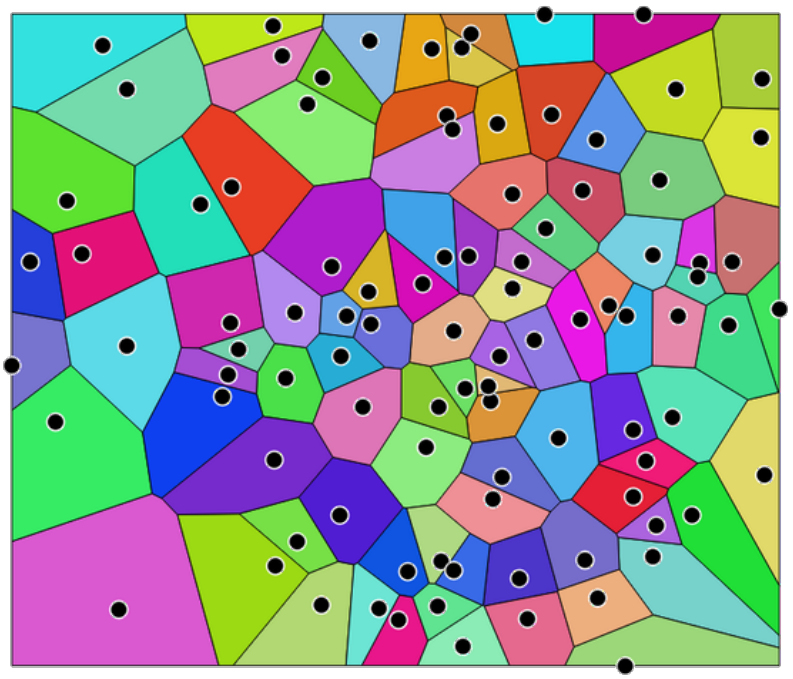


Figura24.148: Polígonos de Voronoi

Default menu: *Vector* ► *Geometry Tools*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Região de buffer (% de extensão)	`BUFFER`	[numeric: double] Padrão: 0.0	The extent of the output layer will be this much bigger than the extent of the input layer
Tolerância Opcional	`TOLERANCE`	[numeric: double] Padrão: 0.0	Specifies an optional snapping tolerance which can be used to improve the robustness of the voronoi.
Copy attributes from input features	`COPY_ATTRIBUTE`	[boleano] Padrão: Verdadeiro	Specifies whether fields storing involved point features ID should be added to the output. If False, an `id` field is created to identify the polygons.
Polígonos de Voronoi	`OUTPUT`	[vetor: polígono] Padrão: `[Criar camada temporária]`	Specify the output layer (with the Voronoi polygons). *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Polígonos de Voronoi	`OUTPUT`	[vetor: polígono]	Voronoi polygons of the input point vector layer

Código Python

Algorithm ID: `native:voronoipolygons`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.27 Vector overlay

Clip

Clips a vector layer using the features of an additional polygon layer.

Only the parts of the features in the input layer that fall within the polygons of the overlay layer will be added to the resulting layer.

Aviso

Apenas modificação de geometria

Esta operação modifica apenas a geometria das feições. Os valores de atributo das feições **não são modificados**, embora propriedades como área ou comprimento das feições sejam modificadas pela operação de sobreposição. Se tais propriedades forem armazenadas como atributos, esses atributos terão que ser atualizados manualmente.

This algorithm uses spatial indexes on the providers, prepared geometries and apply a clipping operation if the geometry isn't wholly contained by the mask geometry.

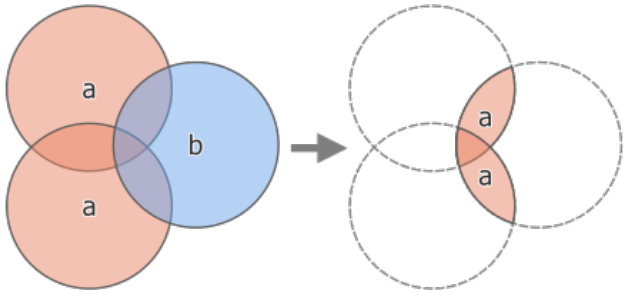


Figura24.149: Clipping operation between a two-feature input layer 'a' and a single feature overlay layer 'b' (left) - resulting in a new layer with the modified 'a' features (right)

Allows features in-place modification 1 de ponto, linha e características poligonais.

Default menu: *Vector* ► *Geoprocessing Tools*

Ver também

Interseção, *Diferença*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Layer containing the features to be clipped
Overlay layer	`OVERLAY`	[vetor: polígono]	Layer containing the clipping features

continua na próxima página

Tabela 24.760 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Clipped	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the layer to contain the features from the input layer that are inside the overlay (clipping) layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Clipped	`OUTPUT`	[o mesmo que entrada]	Layer containing features from the input layer split by the overlay layer.

Código Python

Algorithm ID: `native:clip`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Diferença

Extracts features from the input layer that don't fall within the boundaries of the overlay layer.

Input layer features that partially overlap the overlay layer feature(s) are split along the boundary of those feature(s) and only the portions outside the overlay layer features are retained.

Aviso

Apenas modificação de geometria

Esta operação modifica apenas a geometria das feições. Os valores de atributo das feições **não são modificados**, embora propriedades como área ou comprimento das feições sejam modificadas pela operação de sobreposição. Se tais propriedades forem armazenadas como atributos, esses atributos terão que ser atualizados manualmente.

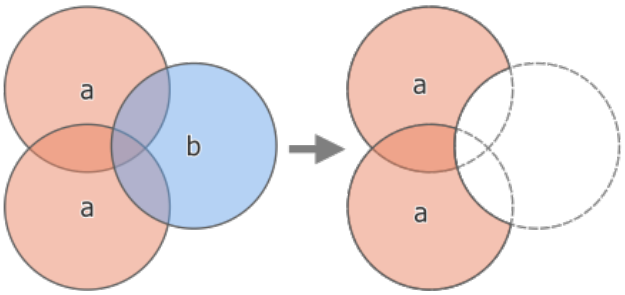


Figura24.150: Difference operation between a two-features input layer 'a' and a single feature overlay layer 'b' (left) - resulting in a new layer with the modified 'a' features (right)

Allows features in-place modification 1 de ponto, linha e características poligonais.

Default menu: *Vector ► Geoprocessing Tools*

Ver também

Difference (multiple), *Diferença simétrica*, *Clip*

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Layer to extract (parts of) features from.
Overlay layer	`OVERLAY`	[vector: geometry]	Layer containing the geometries that will be subtracted from the input layer geometries. It is expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer geometries.
Diferença	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the layer to contain the (parts of) features from the input layer that are not inside the overlay layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Grid size Opcional	`GRID_SIZE`	[numeric: double] Padrão: Não definido	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Saídas

Rótulo	Nome	Tipo	Descrição
Diferença	`OUTPUT`	[o mesmo que entrada]	Camada contendo (partes das) feições da camada de entrada que não se sobrepõe à camada de sobreposição.

Código Python

Algorithm ID: `native:difference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Difference (multiple)

Extracts features from the input layer that fall completely outside or only partially overlap the features from any of the overlay layer(s).

For each overlay layer the difference is calculated between the result of all previous difference operations and this overlay layer. Input layer features that partially overlap feature(s) in the overlay layer are split along those features' boundary and only the portions outside the overlay layer features are retained.

Aviso

Apenas modificação de geometria

Esta operação modifica apenas a geometria das feições. Os valores de atributo das feições **não são modificados**, embora propriedades como área ou comprimento das feições sejam modificadas pela operação de sobreposição. Se tais propriedades forem armazenadas como atributos, esses atributos terão que ser atualizados manualmente.

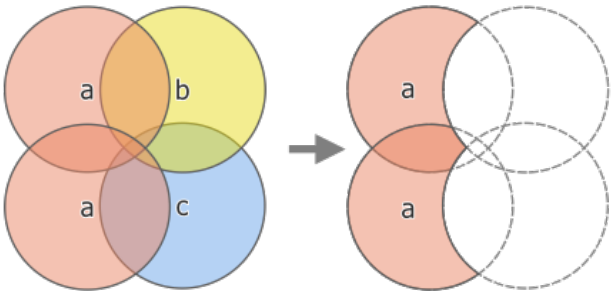


Figura24.151: Difference operation between a two-feature input layer 'a' and single feature overlay layers 'b' and 'c' (left) - resulting in a new layer with the modified 'a' features (right)

Ver também

Diferença, *Diferença simétrica*, *Clip*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Layer to extract (parts of) features from.
Overlay layers	`OVERLAYS`	[vector: geometry] [list]	List of layers containing the geometries that will be subtracted from the input layer geometries. They are expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer geometries.
Diferença	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the layer to contain the (parts of) features from the input layer that do not overlap features of the overlay layers. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Diferença	`OUTPUT`	[o mesmo que entrada]	Layer containing (parts of) features from the input layer not overlapping features from the overlay layers.

Código Python

Algorithm ID: `native:multidifference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extract/clip by extent

Creates a new vector layer that only contains features which fall within a specified extent. Any features which intersect the extent will be included by default.

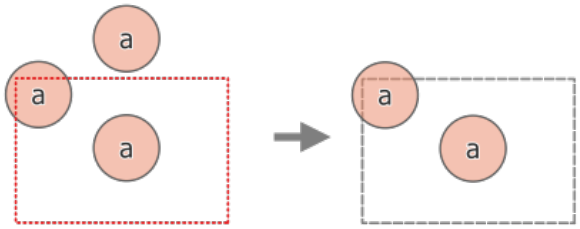


Figura24.152: Operação de extração entre uma camada de entrada de três características “a” e uma extensão tracejada (esquerda) - características resultantes com extensão tracejada para referência (direita)

Optionally, feature geometries can also be clipped to the extent.

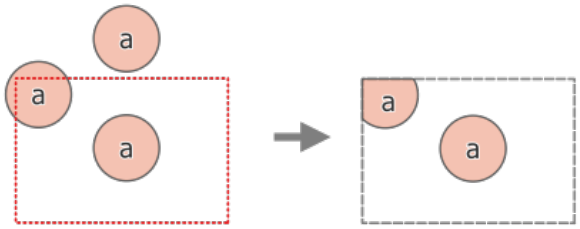


Figura24.153: Operação de extração entre uma camada de entrada de três características “a” e uma extensão tracejada (esquerda) - características resultantes com extensão tracejada para referência (direita)

Ver também

Clip

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Layer to extract (parts of) features from.
Extensão (xmin, xmax, ymin, ymax)	`EXTENT`	[extensão]	Extent for clipping. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Inserir as coordenadas como `xmin, xmax, ymin, ymax`
Clip features to extent	`CLIP`	[boleano] Padrão: Falso	If checked, the geometries will be clipped to the extent chosen instead of taking the whole geometry as output. Moreover, output geometries will be automatically converted to multi geometries to ensure uniform output types.
Extraído	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the layer to contain the features from the input layer that are inside the clip extent. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Extraído	`OUTPUT`	[o mesmo que entrada]	Layer containing the clipped features.

Código Python

Algorithm ID: `native:extractbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Interseção

Extrai as partes das feições da camada de entrada que se sobrepõem aàs feições na camada de sobreposição.

As feições na camada de interseção são atribuídas aos atributos das feições sobrepostas das camadas de entrada e de sobreposição.

Aviso

Apenas modificação de geometria

Esta operação modifica apenas a geometria das feições. Os valores de atributo das feições **não são modificados**, embora propriedades como área ou comprimento das feições sejam modificadas pela operação de sobreposição. Se tais propriedades forem armazenadas como atributos, esses atributos terão que ser atualizados manualmente.

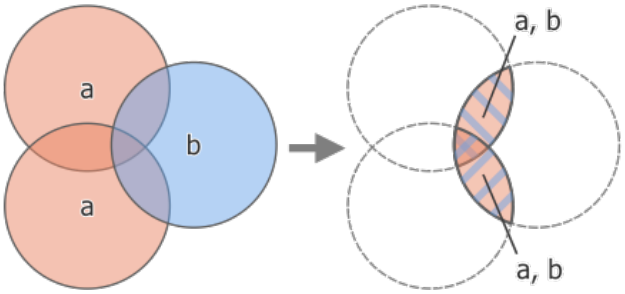


Figura24.154: Intersection operation between a two-feature input layer 'a' and a single feature overlay layer 'b' (left) - overlapping areas become a new two-feature layer with both layers' attributes (right)

Default menu: *Vector ► Geoprocessing Tools*

Aviso

This algorithm drops existing primary keys or FID values and regenerates them in output layers.

Ver também

Intersection (multiple), *Clip*, *Diferença*

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Layer to extract (parts of) features from.
Overlay layer	`OVERLAY`	[vector: geometry]	Layer containing the features to check for overlap. Its features' geometry is expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer's.
Input fields to keep (leave empty to keep all fields) Opcional	`INPUT_FIELDS`	[campo da tabela: qualquer][list] Padrão: Não definido	Field(s) of the input layer to keep in the output. If no fields are chosen all fields are taken.
Overlay fields to keep (leave empty to keep all fields) Opcional	`OVERLAY_FIELDS`	[campo da tabela: qualquer][list] Padrão: Não definido	Field(s) of the overlay layer to keep in the output. If no fields are chosen all fields are taken. Duplicate field names will be appended a count suffix to avoid collision.
Intersecção	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the layer to contain (the parts of) the features from the input layer that overlap one or more features from the overlay layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Overlay fields prefix Opcional	`OVERLAY_FIELDS_PRE`	[string]	Add a prefix to identify fields of the overlay layer. Duplicate field names will be appended a count suffix to avoid collision.
Grid size Opcional	`GRID_SIZE`	[numeric: double] Padrão: Não definido	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Saídas

Rótulo	Nome	Tipo	Descrição
Intersecção	`OUTPUT`	[o mesmo que entrada]	Layer containing (parts of) features from the input layer that overlap the overlay layer.

Código Python

Algorithm ID: `native:intersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Intersection (multiple)

Extracts the overlapping portions of features in the input and all overlay layers.

Features in the output layer are assigned the attributes of the overlapping features from both the input and overlay layers.

Aviso

Apenas modificação de geometria

Esta operação modifica apenas a geometria das feições. Os valores de atributo das feições **não são modificados**, embora propriedades como área ou comprimento das feições sejam modificadas pela operação de sobreposição. Se tais propriedades forem armazenadas como atributos, esses atributos terão que ser atualizados manualmente.

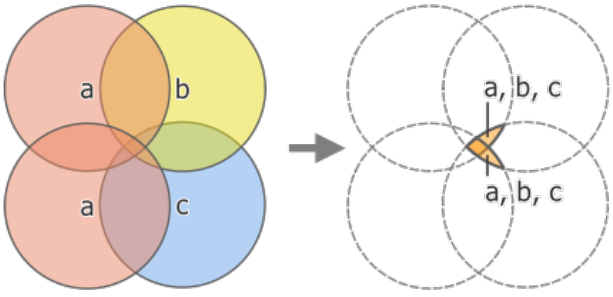


Figura24.155: Intersection operation between a two-feature input layer 'a' and single feature overlay layers 'b' and 'c' (left) - overlapping areas become a new two-feature layer with all layers' attributes (right)

Aviso

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Ver também

Interseção, *Clip*, *Diferença*

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Layer to extract (parts of) features from.
Overlay layers	`OVERLAYS`	[vector: geometry] [list]	Layers containing the features to check for overlap. The features' geometry is expected to have at least as many dimensions (point: 0D, line: 1D, polygon: 2D, volume: 3D) as the input layer's.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Overlay fields prefix Opcional	`OVERLAY_FIELDS_PRE`	[string]	Add a prefix to identify fields of the overlay layers. Duplicate field names will be appended a count suffix to avoid collision.

Saídas

Rótulo	Nome	Tipo	Descrição
Intersecção	`OUTPUT`	[o mesmo que entrada]	Layer containing (parts of) features from the input layer that overlap all the overlay layers.

Código Python

Algorithm ID: `native:multiintersection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Interseções de linha

Cria feições de ponto onde as linhas das duas camadas se cruzam.

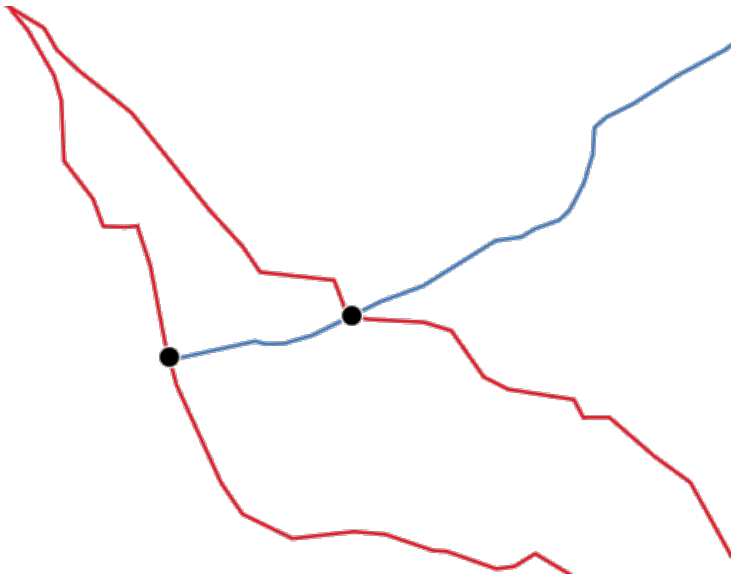


Figura24.156: Pontos de interseção

Default menu: *Vector* ► *Analysis Tools*

> **Aviso**
>
> This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	Input line layer.
camada de interseção*	`INTERSECT`	[vetor: linha]	Layer to use to find line intersections.
Input fields to keep (leave empty to keep all fields) Opcional	`INPUT_FIELDS`	[campo da tabela: qualquer][list] Padrão: Não definido	Field(s) of the input layer to keep in the output. If no fields are chosen all fields are taken.
Intersect fields to keep (leave empty to keep all fields) Opcional	`INTERSECT_FIELDS`	[campo da tabela: qualquer][list] Padrão: Não definido	Field(s) of the intersect layer to keep in the output. If no fields are chosen all fields are taken. Duplicate field names will be appended a count suffix to avoid collision.

continua na próxima página

Tabela 24.775 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Intersecção	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the layer to contain the intersection points of the lines from the input and overlay layers. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Intersect fields prefix Opcional	`INTERSECT_FIELDS_PR`	[string]	Add a prefix to identify fields of the intersect layer.

Saídas

Rótulo	Nome	Tipo	Descrição
Interseções	`OUTPUT`	[vetor: ponto]	Point vector layer of the lines intersections, with both layers' attributes.

Código Python

Algorithm ID: `native:lineintersections`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Split with lines

Splits the lines or polygons in one layer using the lines or polygon rings in another layer to define the breaking points. Intersection between geometries in both layers are considered as split points.

Output will contain multi geometries for split features.

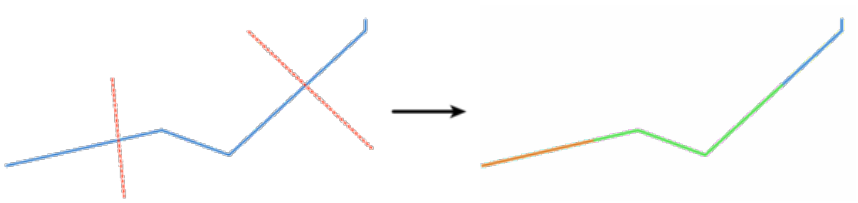


Figura24.157: Linhas divididas

permite features in-place modification 1.

Aviso

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha, polígono]	Camada que contém as linhas ou polígonos a serem divididos.
Split layer	`LINES`	[vetor: linha, polígono]	Layer whose lines or rings are used to define the breaking points.
Dividir	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the layer to contain the splitted (in case they are intersected by a line in the split layer) line/polygon features from the input layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Rótulo	Nome	Tipo	Descrição
Dividir	`OUTPUT`	[o mesmo que entrada]	Output vector layer with split lines or polygons from input layer.

Código Python

Algorithm ID: `native:splitwithlines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Diferença simétrica

Creates a layer containing features from both the input and overlay layers but with the overlapping areas between the two layers removed.

The attribute table of the symmetrical difference layer contains attributes and fields from both the input and overlay layers.

Aviso

Apenas modificação de geometria

Esta operação modifica apenas a geometria das feições. Os valores de atributo das feições **não são modificados**, embora propriedades como área ou comprimento das feições sejam modificadas pela operação de sobreposição. Se tais propriedades forem armazenadas como atributos, esses atributos terão que ser atualizados manualmente.

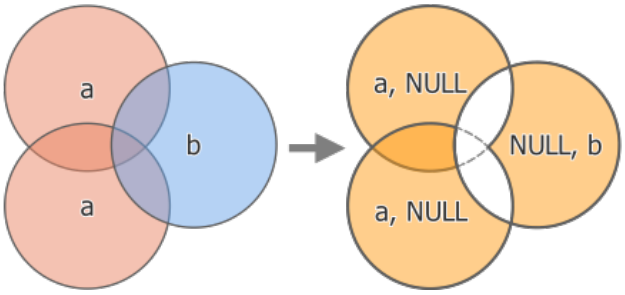


Figura24.158: Symmetrical difference operation between a two-features input layer 'a' and a single feature overlay layer 'b' (left) - resulting three-feature layer with both layers' attributes (right)

Default menu: *Vector ► Geoprocessing Tools*

Aviso

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Ver também

Diferença, *Clip*, *Interseção*

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Primeira camada para extrair (partes de) feições de.
Overlay layer	`OVERLAY`	[vector: geometry]	Second layer to extract (parts of) features from. Ideally the geometry type should be the same as input layer.
Diferença simétrica	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the layer to contain (the parts of) the features from the input and overlay layers that do not overlap features from the other layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Overlay fields prefix Opcional	`OVERLAY_FIELDS_PRE`	[string]	Add a prefix to identify fields of the overlay layer. Duplicate field names will be appended a count suffix to avoid collision.
Grid size Opcional	`GRID_SIZE`	[numeric: double] Padrão: Não definido	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Saídas

Rótulo	Nome	Tipo	Descrição
Diferença simétrica	`OUTPUT`	[o mesmo que entrada]	Layer containing (parts of) features from each layer not overlapping the other layer, with both layers' attributes.

Código Python

Algorithm ID: `native:symmetricaldifference`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

União

Checks overlaps between features within the input layer and creates separate features for overlapping and non-overlapping parts. The area of overlap will create as many identical overlapping features as there are features that participate in that overlap.

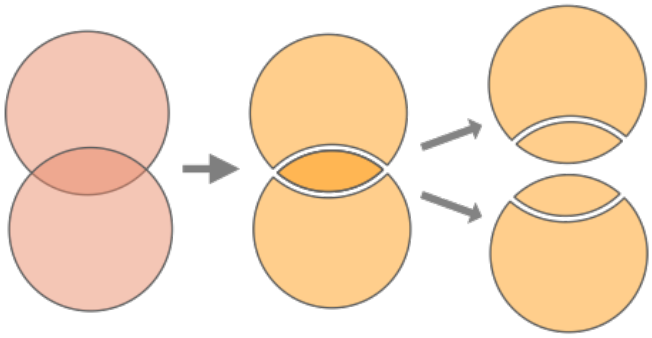


Figura24.159: Union operation with a single input layer with two overlapping features (left) - resulting in four features (middle) - features moved for clarity (right)

An overlay layer can also be used, in which case features from each layer are split at their overlap with features from the other one, creating a layer containing all the portions from both input and overlay layers. Features on the same layer will not split each other. The attribute table of the union layer is filled with attribute values from the respective original layer for non-overlapping features, and attribute values from both layers for overlapping features.

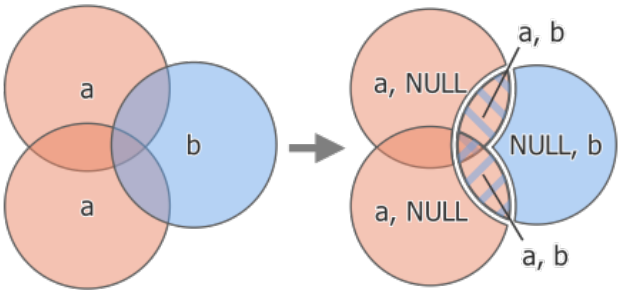


Figura24.160: Union operation between a two-feature input layer 'a' and a single feature overlay layer 'b' (left) - resulting five-feature layer with attributes from both layers (right)

Nota

With an overlay layer, features on the same layer will not split each other. If you want to split overlaps on the same layer as well as other layers, first run the algorithm with multiple layers then run the algorithm again with

only the previous output.

Default menu: *Vector ► Geoprocessing Tools*

Aviso

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Ver também

Union (multiple), *Clip*, *Diferença*, *Interseção*

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Input vector layer to split at any intersections.
Overlay layer Opcional	`OVERLAY`	[vector: geometry]	Layer that will be combined to the first one. Ideally the geometry type should be the same as input layer.
União	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the layer to contain the (split and duplicated) features from the input layer and the overlay layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Overlay fields prefix Opcional	`OVERLAY_FIELDS_PRE`	[string]	Add a prefix to identify fields of the overlay layer. Duplicate field names will be appended a count suffix to avoid collision.
Grid size Opcional	`GRID_SIZE`	[numeric: double] Padrão: Não definido	If provided, the input geometries are snapped to a grid of the given size, and the result vertices are computed on that same grid.

Saídas

Rótulo	Nome	Tipo	Descrição
União	`OUTPUT`	[o mesmo que entrada]	Layer containing all the overlapping and non-overlapping parts from the processed layer(s).

Código Python

Algorithm ID: `native:union`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Union (multiple)

Checks overlaps between features within the input layer and creates separate features for overlapping and non-overlapping parts. The area of overlap will create as many identical overlapping features as there are features that participate in that overlap.

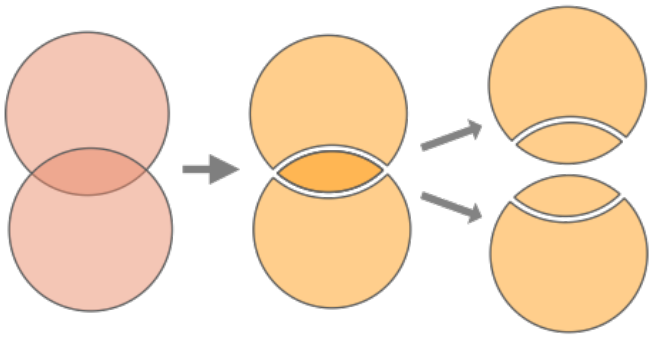


Figura24.161: Union operation with a single input layer with two overlapping features (left) - resulting in four features (middle) - features moved for clarity (right)

Multiple overlay layers can also be used, in which case features from each layer are split at their overlap with features from all other layers, creating a layer containing all the portions from both input and overlay layers. Features on the same layer will not split each other. The attribute table of the Union layer is filled with attribute values from the respective original layer for non-overlapping features, and attribute values from overlay layers for overlapping features.

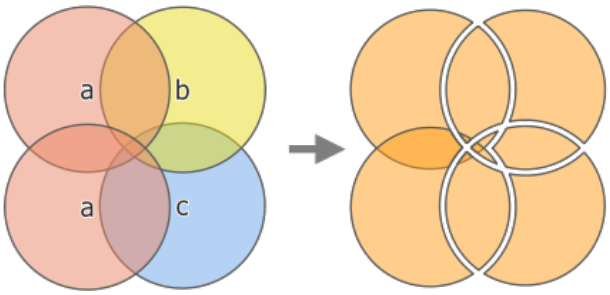


Figura24.162: Union operation between a two-feature input layer 'a' and single feature overlay layers 'b' and 'c' (left) - resulting eleven-feature layer with attributes from all layers (right)

Nota

With an overlay layer, features on the same layer will not split each other. If you want to split overlaps on the same layer as well as other layers, first run the algorithm with multiple layers then run the algorithm again with only the previous output.

Aviso

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Ver também

União, *Clip*, *Diferença*, *Interseção*

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Input vector layer to split at any intersections.
Overlay layers Opcional	`OVERLAYS`	[vector: geometry] [list]	Layers that will be combined to the first one. Ideally the geometry type should be the same as input layer.
União	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the layer to contain the (split and duplicated) features from the input layer and the overlay layers. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Overlay fields prefix Opcional	`OVERLAY_FIELDS_PRE`	[string]	Add a prefix to identify fields of the overlay layers. Duplicate field names will be appended a count suffix to avoid collision.

Saídas

Rótulo	Nome	Tipo	Descrição
União	`OUTPUT`	[o mesmo que entrada]	Layer containing all the overlapping and non-overlapping parts from the processed layer(s), with all layers' attributes.

Código Python

Algorithm ID: `native:multiunion`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.28 Seleção vetorial

Extrair por atributo

Cria duas camadas vetoriais a partir de uma camada de entrada: uma conterá apenas feições correspondentes, enquanto a segunda conterá todas as feições não correspondentes.

Os critérios para adicionar feições à camada resultante são baseados nos valores de um atributo da camada de entrada.

Ver também

Selecionar por atributo

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada para extrair feições.
Atributo de seleção	`FIELD`	[campo da tabela: qualquer]	Campo de filtragem da camada
Operador	`OPERATOR`	[enumeração] Padrão: 0	Muitos operadores diferentes estão disponíveis: • 0 — = • 1 — ≠ • 2 — >> • 3 — >= • 4 — < • 5 — <= • 6 — começa com • 7 — contém • 8 — é nulo • 9 — não é nulo • 10 — não contém
Valor Opcional	`VALUE`	[string]	Valor a ser avaliado
Extraído (atributo)	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar Camada Temporária]`	Specify the output vector layer for matching features. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.
Extraído (não correspondente)	`FAIL_OUTPUT`	[o mesmo que entrada] Default: `[Pular saída]`	Specify the output vector layer for non-matching features. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Extraído (atributo)	`OUTPUT`	[o mesmo que entrada]	Camada vetorial com feições correspondentes da camada de entrada
Extraído (não correspondente)	`FAIL_OUTPUT`	[o mesmo que entrada]	Camada vetorial com feições não correspondentes da camada de entrada

Código Python

Algorithm ID: `native:extractbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair por expressão

Cria duas camadas vetoriais a partir de uma camada de entrada: uma conterá apenas feições correspondentes, enquanto a segunda conterá todas as feições não correspondentes.

The criteria for adding features to the resulting layer is based on a QGIS expression. For more information about expressions see the *Expressões*.

Ver também

Selecionar por expressão

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
Expressão	`EXPRESSION`	[expressão]	Expressão para filtrar a camada vetorial
Feições correspondentes	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar Camada Temporária]`	Specify the output vector layer for matching features. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

continua na próxima página

Tabela 24.791 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Non-matching	`FAIL_OUTPUT`	[o mesmo que entrada] Default: `[Pular saída]`	Specify the output vector layer for non-matching features. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Feições correspondentes	`OUTPUT`	[o mesmo que entrada]	Camada vetorial com feições correspondentes da camada de entrada
Non-matching	`FAIL_OUTPUT`	[o mesmo que entrada]	Camada vetorial com feições não correspondentes da camada de entrada

Código Python

Algorithm ID: `native:extractbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair por localização

Cria uma nova camada vetorial que contém apenas feições correspondentes de uma camada de entrada.

The criteria for adding features to the resulting layer is based on the spatial relationship between each feature and the features in an additional layer.

Ver também

Selecionar pela localização, *Extratoir à distância*

Explorando relações espaciais

Predicados geométricos são funções booleanas usadas para determinar a relação espacial que um recurso tem com outro, comparando se e como suas geometrias compartilham uma parte do espaço.

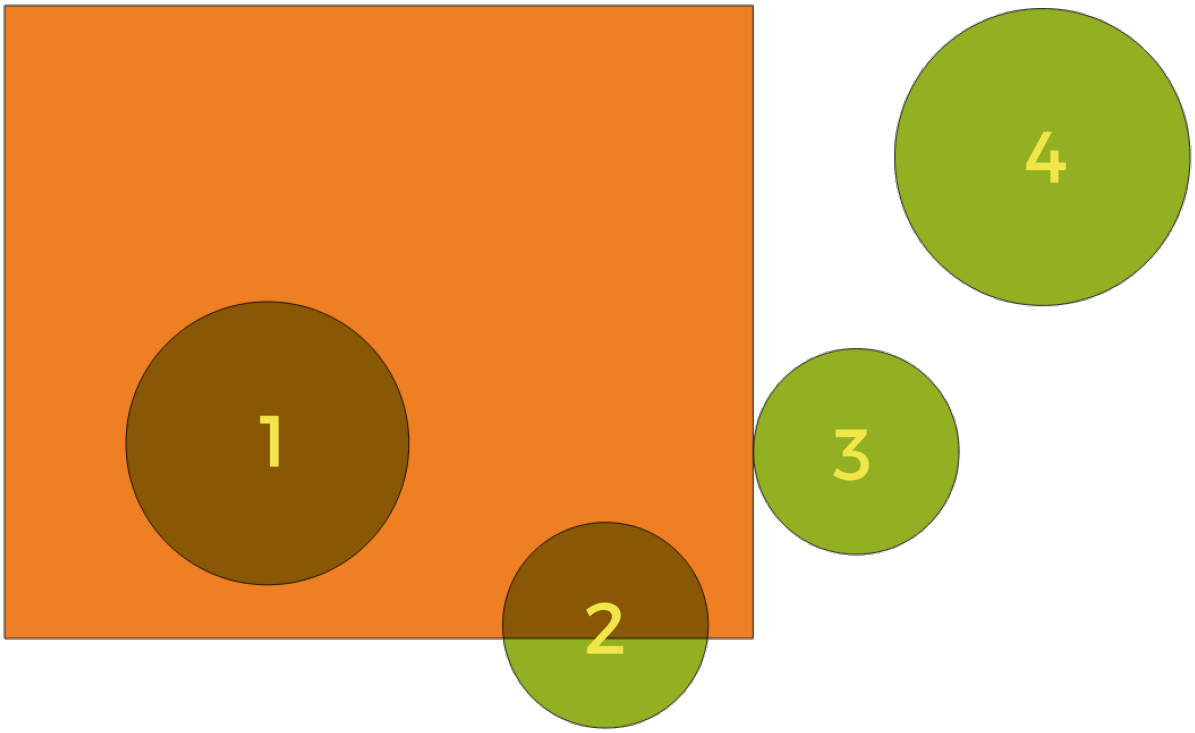


Figura24.163: Procurando relações espaciais entre camadas

Usando a figura acima, estamos procurando os círculos verdes comparando-os espacialmente com o retângulo laranja. Os predicados geométricos disponíveis são:

Intersecção
: Testa se uma geometria intercepta outra. Retorna 1 (verdadeiro) se as geometrias se cruzarem espacialmente (compartilhar qualquer parte do espaço - sobreposição ou toque) e 0 se não houver. Na imagem acima, isso retornará os círculos 1, 2 e 3.

Contém
: Retorna 1 (verdadeiro) se e somente se nenhum ponto de b estiver no exterior de a, e pelo menos um ponto do interior de b estiver no interior de a. Na figura, nenhum círculo é retornado, mas o retângulo seria se você o procurasse ao contrário, porque contém o círculo 1 completamente. Este é o oposto de *estão dentro*.

Disjoint
: Retorna 1 (verdadeiro) se as geometrias não compartilham nenhuma parte do espaço (sem sobreposição, sem toque). Apenas o círculo 4 é retornado.

Igual
: Retorna 1 (verdadeiro) se e somente se as geometrias forem exatamente iguais. Nenhum círculo será retornado.

Toca
: Testa se uma geometria toca outra. Retorna 1 (verdadeiro) se as geometrias tiverem pelo menos um ponto em comum, mas seus interiores não se cruzarem. Apenas o círculo 3 é retornado.

Sobrepor
: Testa se uma geometria se sobrepõe a outra. Retorna 1 (verdadeiro) se as geometrias compartilham espaço, são da mesma dimensão, mas não estão completamente contidas umas nas outras. Apenas o círculo 2 é retornado.

Estão dentro

Testa se uma geometria está dentro de outra. Retorna 1 (verdadeiro) se a geometria a estiver completamente dentro da geometria b. Apenas o círculo 1 é retornado.

Cruza
Retorna 1 (verdadeiro) se as geometrias fornecidas tiverem alguns, mas não todos, pontos internos em comum e o cruzamento real for de uma dimensão menor que a geometria fornecida mais alta. Por exemplo, uma linha que cruza um polígono cruzará como uma linha (verdadeiro). Duas linhas que se cruzam se cruzarão como um ponto (verdadeiro). Dois polígonos se cruzam como um polígono (falso). Na imagem, nenhum círculo será devolvido.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Extrair feições de	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Where the features (geometric predicate)	`PREDICATE`	[enumeração] [lista] Padrão: [0]	Type of spatial relation the input feature should have with an intersect feature so that it could be selected. One or more of: • 0 — Interseção • 1 — contém • 2 — disjoint • 3 — igual • 4 — toca • 5 — sobrepor • 6 — estão dentro • 7 — cruza Se mais de uma condição for escolhida, pelo menos uma delas (operação OU) tem que ser atendida para que uma característica seja extraída.
Por comparação com as feições de	`INTERSECT`	[vector: geometry]	Camada vetorial de interseção
Extraído (local)	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the features that have the chosen spatial relationship(s) with one or more features in the comparison layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Extraído (local)	`OUTPUT`	[o mesmo que entrada]	Vector layer with features from the input layer that have the chosen spatial relationship(s) with one or more features in the comparison layer.

Código Python

Algorithm ID: `native:extractbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extratoir à distância

Creates a new vector layer that only contains matching features from an input layer. Features are copied wherever they are within the specified maximum distance from the features in an additional reference layer.

Ver também

Selecionar dentro da distância, *Extrair por localização*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Extrair feições de	`INPUT`	[vector: geometry]	Input vector layer to copy features from
Por comparação com as feições de	`REFERENCE`	[vector: geometry]	Camada vetorial cuja proximidade de feições é usada
Onde as feições estão dentro	`DISTANCE`	[numeric: double] Padrão: 100.0	The maximum distance around reference features to select input features within
Modificar a seleção atual	`METHOD`	[enumeração] Padrão: 0	Como a seleção do algoritmo deve ser gerenciada. Um de: • 0 — criando nova seleção • 1 — adicionando à seleção atual • 2 — selecionando dentro da seleção atual • 3 — removendo da seleção atual
Extraído (local)	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the features that are within the set distance from reference features. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Extraído (local)	`OUTPUT`	[o mesmo que entrada]	Camada vetorial com feições da camada de entrada que correspondem à condição de distância das feições de referência

Código Python

Algorithm ID: `native:extractwithindistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Filtrar por tipo de geometria

Filtra recursos por seu tipo de geometria. Os recursos de entrada serão direcionados para diferentes saídas com base no fato de terem uma geometria de ponto, linha ou polígono.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada para avaliar

Saídas

Etiqueta	Nome	Tipo	Descrição
Feições de ponto Opcional	`POINTS`	[vetor: ponto]	Camada com pontos
Feições de linha Opcional	`LINES`	[vetor: linha]	Camada com linhas
Feições de polígono Opcional	`POLYGONS`	[vetor: polígono]	Camada com polígonos
Feições sem geometria Opcional	`NO_GEOMETRY`	[vector: table]	Camada vetorial sem geometria

Código Python

ID do algoritmo: `native:filterbygeometry`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extração aleatória

Pega uma camada vetorial e gera uma nova que contém apenas um subconjunto das feições na camada de entrada.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset.

Aviso

This algorithm drops existing primary keys or FID values and regenerate them in output layers.

Ver também

Seleção aleatória

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de origem para selecionar as feições de
Método	`METHOD`	[enumeração] Padrão: 0	Métodos de seleção aleatória. Um de: • 0 — Número de feições selecionadas • 1 — Porcentagem de feições selecionadas
Número/porcentage de feições selecionadas	`NUMBER`	[numeric: integer] Padrão: 10	Número ou porcentagem de feições a serem selecionadas
Extraído (aleatório)	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the randomly selected features. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Extraído (aleatório)	`OUTPUT`	[o mesmo que entrada]	Camada vetorial contendo feições selecionadas aleatoriamente da camada de entrada

Código Python

Algorithm ID: `native:randomextract`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extração aleatória dentro de subconjuntos

Pega uma camada vetorial e gera uma nova que contém apenas um subconjunto das feições na camada de entrada.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset. The percentage/count value is not applied to the whole layer, but instead to each category. Categories are defined according to a given attribute.

Ver também

Seleção aleatória dentro de subconjuntos

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial para selecionar as feições de
Campo ID	`FIELD`	[campo da tabela: qualquer]	Categoria da camada vetorial de origem para selecionar as feições de
Método	`METHOD`	[enumeração] Padrão: 0	Método de seleção aleatória. Um de: • 0 — Número de feições selecionadas • 1 — Porcentagem de feições selecionadas
Número/porcentage de feições selecionadas	`NUMBER`	[numeric: integer] Padrão: 10	Número ou porcentagem de feições a serem selecionadas

continua na próxima página

Tabela 24.801 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Extraído (estratificado aleatório)	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer for the randomly selected features. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Extraído (estratificado aleatório)	`OUTPUT`	[o mesmo que entrada]	Camada vetorial contendo feições selecionadas aleatoriamente da camada de entrada

Código Python

Algorithm ID: `qgis:randomextractwithinsubsets`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Seleção aleatória

Pega uma camada vetorial e seleciona um subconjunto de suas feições. Nenhuma nova camada é gerada por este algoritmo.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset.

Menu padrão: *Vetor ► Ferramentas de pesquisa*

Ver também

Extração aleatória

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial para a seleção
Método	`METHOD`	[enumeração] Padrão: 0	Método de seleção aleatória. Um de: • 0 — Número de feições selecionadas • 1 — Porcentagem de feições selecionadas
Número/porcentage de feições selecionadas	`NUMBER`	[numeric: integer] Padrão: 10	Número ou porcentagem de feições a serem selecionadas

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[o mesmo que entrada]	A camada de entrada com feições selecionadas

Código Python

Algorithm ID: `qgis:randomselection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Seleção aleatória dentro de subconjuntos

Pega uma camada vetorial e seleciona um subconjunto de suas feições. Nenhuma nova camada é gerada por este algoritmo.

The subset is defined randomly, based on feature IDs, using a percentage or count value to define the total number of features in the subset.

O valor de porcentagem/contagem não é aplicado a toda a camada, mas a cada categoria.

As categorias são definidas de acordo com um determinado atributo, que também é especificado como parâmetro de entrada para o algoritmo.

Nenhuma nova saída é criada.

Menu padrão: *Vetor ► Ferramentas de pesquisa*

Ver também

Extração aleatória dentro de subconjuntos

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial para selecionar feições
Campo ID	`FIELD`	[campo da tabela: qualquer]	Categoria da camada de entrada para selecionar as feições de
Método	`METHOD`	[enumeração] Padrão: 0	Método de seleção aleatória. Um de: • 0 — Número de feições selecionadas • 1 — Porcentagem de feições selecionadas
Número/porcentage de feições selecionadas	`NUMBER`	[numeric: integer] Padrão: 10	Número ou porcentagem de feições a serem selecionadas

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[o mesmo que entrada]	A camada de entrada com feições selecionadas

Código Python

Algorithm ID: `qgis:randomselectionwithinsubsets`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Selecionar por atributo

Cria uma seleção em uma camada vetorial.

O critério para selecionar feições é baseado nos valores de um atributo da camada de entrada.

Ver também

Extrair por atributo

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial para selecionar feições
Atributo de seleção	`FIELD`	[campo da tabela: qualquer]	Campo de filtragem da camada
Operador	`OPERATOR`	[enumeração] Padrão: 0	Muitos operadores diferentes estão disponíveis: • 0 — = • 1 — ≠ • 2 — >> • 3 — >= • 4 — < • 5 — <= • 6 — começa com • 7 — contém • 8 — é nulo • 9 — não é nulo • 10 — não contém
Valor Opcional	`VALUE`	[string]	Valor a ser avaliado
Modificar a seleção atual	`METHOD`	[enumeração] Padrão: 0	Como a seleção do algoritmo deve ser gerenciada. Um de: • 0 — criando nova seleção • 1 — adicionando à seleção atual • 2 — removendo da seleção atual • 3 — selecionando dentro da seleção atual

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[o mesmo que entrada]	A camada de entrada com feições selecionadas

Código Python

Algorithm ID: `qgis:selectbyattribute`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Selecionar por expressão

Cria uma seleção em uma camada vetorial.

The criteria for selecting features is based on a QGIS expression. For more information about expressions see the *Expressões*.

Ver também

Extrair por expressão

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
Expressão	`EXPRESSION`	[expressão]	Expressão para filtrar a camada de entrada
Modificar a seleção atual	`METHOD`	[enumeração] Padrão: 0	Como a seleção do algoritmo deve ser gerenciada. Um de: • 0 — criando nova seleção • 1 — adicionando à seleção atual • 2 — removendo da seleção atual • 3 — selecionando dentro da seleção atual

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[o mesmo que entrada]	A camada de entrada com feições selecionadas

Código Python

Algorithm ID: `qgis:selectbyexpression`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Selecionar pela localização

Cria uma seleção em uma camada vetorial.

Os critérios para selecionar feições são baseados na relação espacial entre cada feição e as feições em uma camada adicional.

Menu padrão: *Vetor ► Ferramentas de pesquisa*

Ver também

Extrair por localização, *Selecionar dentro da distância*

Explorando relações espaciais

Predicados geométricos são funções booleanas usadas para determinar a relação espacial que um recurso tem com outro, comparando se e como suas geometrias compartilham uma parte do espaço.

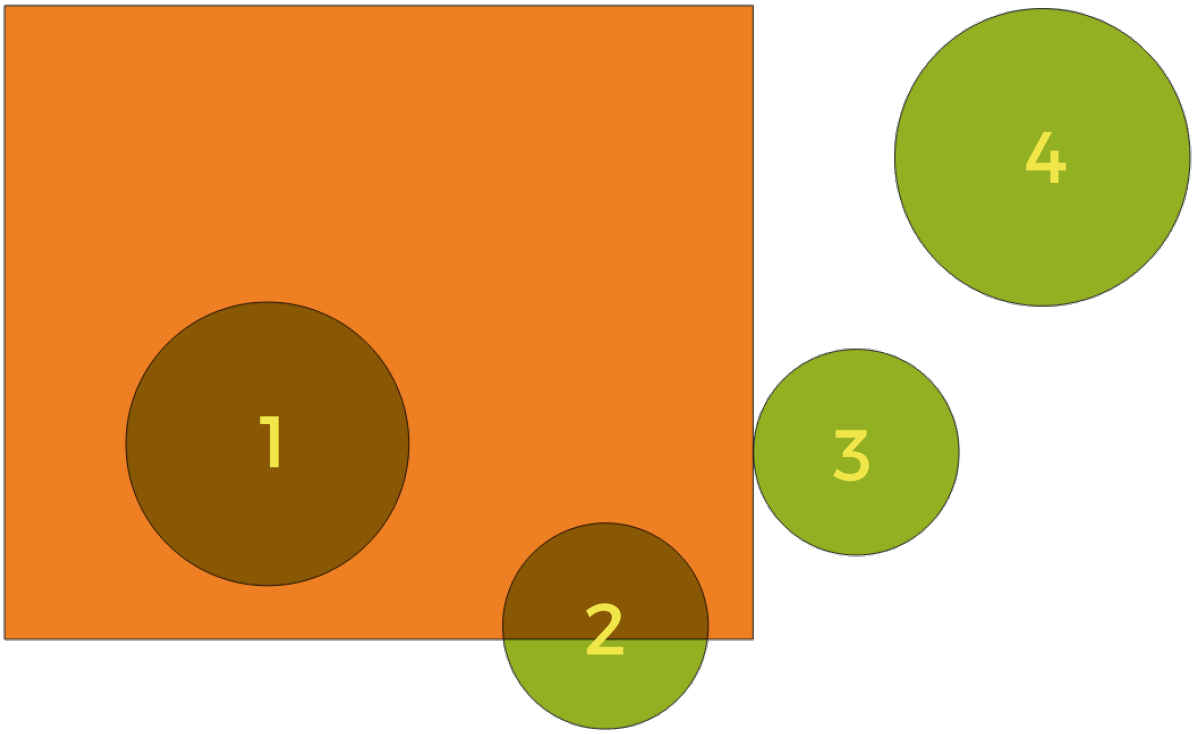


Figura24.164: Procurando relações espaciais entre camadas

Usando a figura acima, estamos procurando os círculos verdes comparando-os espacialmente com o retângulo laranja. Os predicados geométricos disponíveis são:

Intersecção
: Testa se uma geometria intercepta outra. Retorna 1 (verdadeiro) se as geometrias se cruzarem espacialmente (compartilhar qualquer parte do espaço - sobreposição ou toque) e 0 se não houver. Na imagem acima, isso retornará os círculos 1, 2 e 3.

Contém
: Retorna 1 (verdadeiro) se e somente se nenhum ponto de b estiver no exterior de a, e pelo menos um ponto do interior de b estiver no interior de a. Na figura, nenhum círculo é retornado, mas o retângulo seria se você o procurasse ao contrário, porque contém o círculo 1 completamente. Este é o oposto de *estão dentro*.

Disjoint
Retorna 1 (verdadeiro) se as geometrias não compartilham nenhuma parte do espaço (sem sobreposição, sem toque). Apenas o círculo 4 é retornado.

Igual
Retorna 1 (verdadeiro) se e somente se as geometrias forem exatamente iguais. Nenhum círculo será retornado.

Toca
Testa se uma geometria toca outra. Retorna 1 (verdadeiro) se as geometrias tiverem pelo menos um ponto em comum, mas seus interiores não se cruzarem. Apenas o círculo 3 é retornado.

Sobrepor
Testa se uma geometria se sobrepõe a outra. Retorna 1 (verdadeiro) se as geometrias compartilham espaço, são da mesma dimensão, mas não estão completamente contidas umas nas outras. Apenas o círculo 2 é retornado.

Estão dentro
Testa se uma geometria está dentro de outra. Retorna 1 (verdadeiro) se a geometria a estiver completamente dentro da geometria b. Apenas o círculo 1 é retornado.

Cruza
Retorna 1 (verdadeiro) se as geometrias fornecidas tiverem alguns, mas não todos, pontos internos em comum e o cruzamento real for de uma dimensão menor que a geometria fornecida mais alta. Por exemplo, uma linha que cruza um polígono cruzará como uma linha (verdadeiro). Duas linhas que se cruzam se cruzarão como um ponto (verdadeiro). Dois polígonos se cruzam como um polígono (falso). Na imagem, nenhum círculo será devolvido.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Selecionar feições de	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Where the features (geometric predicate)	`PREDICATE`	[enumeração] [lista] Padrão: [0]	Type of spatial relation the input feature should have with an intersect feature so that it could be selected. One or more of: • 0 — Interseção • 1 — contém • 2 — disjoint • 3 — igual • 4 — toca • 5 — sobrepor • 6 — estão dentro • 7 — cruza Se mais de uma condição for escolhida, pelo menos uma delas (operação OU) tem que ser atendida para que uma característica seja extraída.
Por comparação com as feições de	`INTERSECT`	[vector: geometry]	Camada vetorial de interseção
Modificar a seleção atual	`METHOD`	[enumeração] Padrão: 0	Como a seleção do algoritmo deve ser gerenciada. Um de: • 0 — criando nova seleção • 1 — adicionando à seleção atual • 2 — selecionando dentro da seleção atual • 3 — removendo da seleção atual

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[o mesmo que entrada]	A camada de entrada com feições selecionadas

Código Python

Algorithm ID: `native:selectbylocation`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Selecionar dentro da distância

cria uma seleção em uma camada vetorial. As feições são selecionadas onde quer que estejam dentro da distância máxima especificada das feições em uma camada de referência adicional.

Ver também

Extratoir à distância, *Selecionar pela localização*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Selecionar feições de	`INPUT`	[vector: geometry]	Camada vetorial de entrada para selecionar feições de
Por comparação com as feições de	`REFERENCE`	[vector: geometry]	Camada vetorial cuja proximidade de feições é usada
Onde as feições estão dentro	`DISTANCE`	[numeric: double] Padrão: 100.0	A distância máxima em torno das feições de referência para selecionar as feições de entrada
Modificar a seleção atual	`METHOD`	[enumeração] Padrão: 0	Como a seleção do algoritmo deve ser gerenciada. Um de: • 0 — criando nova seleção • 1 — adicionando à seleção atual • 2 — selecionando dentro da seleção atual • 3 — removendo da seleção atual

Saídas

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[o mesmo que entrada]	A camada de entrada com feições selecionadas

Código Python

Algorithm ID: `native:selectwithindistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.29 Tabela de vetores

Adicionar campo de autoincremento

Adiciona um novo campo inteiro a uma camada vetorial, com um valor sequencial para cada feição.

Este campo pode ser usado como um ID exclusivo para as feições da camada. O novo atributo não é adicionado à camada de entrada, mas uma nova camada é gerada.

O valor inicial para a série incremental pode ser especificado. Opcionalmente, a série incremental pode ser baseada em campos de agrupamento e uma ordem de classificação para feições também pode ser especificada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada vetorial de entrada.
Campo nome	`FIELD_NAME`	[string] Padrão: 'AUTO'	Nome do campo com valores autoincrementais
Valores iniciais em Opcional	`START`	[numeric: integer] Padrão: 0	Escolha o número inicial da contagem incremental
Modulus value Opcional	`MODULUS`	[numeric: integer] Padrão: 0	Specifying an optional modulus value will restart the count to START whenever the field value reaches the modulus value. `0` means no restart.
Agrupar valores por Opcional	`GROUP_FIELDS`	[campo da tabela: qualquer][list]	Select grouping field(s): instead of a single count run for the whole layer, a separate count is processed for each value returned by the combination of these fields.
Expressão ordenada Opcional	`SORT_EXPRESSIO`	[expressão]	Usar uma expressão para classificar as feições na camada globalmente ou, se definido, com base nos campos do grupo.

continua na próxima página

Tabela 24.815 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Ordernar ascendente	`SORT_ASCENDING`	[boleano] Padrão: Verdadeiro	Quando uma `expressão ordenada` é definida, use esta opção para controlar a ordem na qual os valores são atribuídos às feições.
Ordenar nulos primeiro	`SORT_NULLS_FIR`	[boleano] Padrão: Falso	Quando uma ``expressão ordenada`` é definida, use esta opção para definir se os valores *Nulo* são contados primeiro ou por último.
Incrementado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer with the auto increment field. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Incrementado	`OUTPUT`	[o mesmo que entrada]	Vector layer with auto incremental field

Código Python

Algorithm ID: `native:addautoincrementalfield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Adicionar um campo à tabela de atributos

Adiciona um novo campo a uma camada vetorial.

O nome e as características do atributo são definidos como parâmetros.

O novo atributo não é adicionado à camada de entrada, mas uma nova camada é gerada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada de entrada
Campo nome	`FIELD_NAME`	[string]	Nome do novo campo
Tipo de campo	`FIELD_TYPE`	[enumeração] Padrão: 0	Tipo do novo campo. Você pode escolher entre: • 0 — Integer (32 bit) • 1 — Decimal (double) • 2 — Text (string) • 3 — Boolean • 4 — Date • 5 — Time • 6 — Date & Time • 7 — Binary Object (BLOB) • 8 — String List • 9 — Integer List • 10 — Decimal (double) List
Comprimento do campo	`FIELD_LENGTH`	[numeric: integer] Padrão: 10	Comprimento do campo
Precisão do campo	`FIELD_PRECISIO`	[numeric: integer] Padrão: 0	Precisão do campo. Usual com campo tipo flutuante.
Field alias Opcional	`FIELD_ALIAS`	[string]	Set a name to use as alias for the field. Not supported by all format types.
Field comment Opcional	`FIELD_COMMENT`	[string]	Store a comment describing the field. Not supported by all format types.
Adicionado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Adicionado	`OUTPUT`	[o mesmo que entrada]	Camada vetorial com novo campo adicionado

Código Python

Algorithm ID: `native:addfieldtoattributestable`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Adicionar índice do campo valores únicos

Pega uma camada vetorial e um atributo e adiciona um novo campo numérico.

Values in this field correspond to values in the specified attribute, so features with the same value for the attribute will have the same value in the new numeric field.

This creates a numeric equivalent of the specified attribute, which defines the same classes.

O novo atributo não é adicionado à camada de entrada, mas uma nova camada é gerada.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada de entrada.
Campo de classe	`FIELD`	[campo da tabela: qualquer]	Feições que têm o mesmo valor para este campo obterão o mesmo índice.
Nome do campo de saída	`FIELD_NAME`	[string] Default: 'NUM_FIELD'	Nome do novo campo que contém os índices.
Layer with index field	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Vector layer with the numeric field containing indexes. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

continua na próxima página

Tabela 24.819 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Class summary	`SUMMARY_OUTPUT`	[vector: table] Default: `[Pular saída]`	Specify the table to contain the summary of the class field mapped to the corresponding unique value. *One of*: • Ignorar Saída • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Layer with index field	`OUTPUT`	[o mesmo que entrada]	Camada vetorial com o campo numérico contendo índices.
Class summary	`SUMMARY_OUTPUT`	[vector: table]	Table with summary of the class field mapped to the corresponding unique value.

Código Python

Algorithm ID: `native:adduniquevalueindexfield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Adicionar campos X/Y à camada

Adds X and Y (or latitude/longitude) fields to a point layer. The X/Y fields can be calculated in a different CRS to the layer (e.g. creating latitude/longitude fields for a layer in a projected CRS).

☑ permite features in-place modification 1 de pontos da característica.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: ponto]	A camada de entrada.
Sistema de coordenadas	`CRS`	[src] Padrão: "EPSG:4326"	Sistema de referência de coordenadas a ser usado para os campos x e y gerados.
Prefixo do campo Opcional	`PREFIX`	[string]	Prefixo a ser adicionado aos novos nomes de campo para evitar conflitos de nomes com campos na camada de entrada.
Campos adicionados	`OUTPUT`	[vetor: ponto] Padrão: `[Criar camada temporária]`	Specify the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Campos adicionados	`OUTPUT`	[vetor: ponto]	A camada de saída - idêntica à camada de entrada, mas com dois novos campos duplos, `x` e `y`.

Código Python

Algorithm ID: `native:addxyfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Calculadora de campo avançada Python

Adiciona um novo atributo a uma camada vetorial, com valores resultantes da aplicação de uma expressão para cada feição.

A expressão é definida como uma função Python.

Aviso

This algorithm is a potential security risk if executed with unchecked inputs, and may result in system damage or data leaks.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
Nome do campo de resultado	`FIELD_NAME`	[string] Padrão: 'Novo-Campo'	Nome do novo campo
Tipo de campo	`FIELD_TYPE`	[enumeração] Padrão: 0	Tipo do novo campo. Um de: • 0 — Integer (32 bit) • 1 — Decimal (double) • 2 — Text (string) • 3 — Boolean • 4 — Date • 5 — Time • 6 — Date & Time • 7 — Binary Object (BLOB) • 8 — String List • 9 — Integer List • 10 — Decimal (double) List
Comprimento do campo	`FIELD_LENGTH`	[numeric: integer] Padrão: 10	Comprimento do campo
Precisão do campo	`FIELD_PRECISIO`	[numeric: integer] Padrão: 3	Precisão do campo. Usual com campo tipo flutuante.
Expressão global Opcional	`GLOBAL`	[string]	The code in the global expression section will be executed only once before the calculator starts iterating through all the features of the input layer. Therefore, this is the correct place to import necessary modules or to calculate variables that will be used in subsequent calculations.

Fórmula	`FORMULA`	[string]	A fórmula Python para avaliar. Exemplo: Para calcular a área de uma camada de polígono de entrada, você pode adicionar:
			`value = $geom.area()`
			To access an existing field use its name enclosed in the angle brackets. For example, to uppercase values of the "address" field:
			`value = <address>.upper()`

continua na próxima página

Tabela 24.823 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Calculado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Especificar a camada vetorial com o novo campo calculado. Um de: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Calculado	`OUTPUT`	[o mesmo que entrada]	Camada vetorial com o novo campo calculado

Código Python

Algorithm ID: `qgis:advancedpythonfieldcalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Descartar campo(s)

Pega uma camada vetorial e gera uma nova com as mesmas feições, mas sem as colunas selecionadas.

Ver também

Campos retidos

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Insira a camada vetorial para soltar o(s) campo(s) de
Fields to drop	`COLUMN`	[campo da tabela: qualquer][list]	The field(s) to drop

continua na próxima página

Tabela 24.825 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Campos remanescentes	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specify the output vector layer with the remaining fields. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Campos remanescentes	`OUTPUT`	[o mesmo que entrada]	Camada vetorial com os campos remanescentes

Código Python

Algorithm ID: `native:deletecolumn`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Explode HStore Field

Creates a copy of the input layer and adds a new field for every unique key in the HStore field.

The expected field list is an optional comma separated list. If this list is specified, only these fields are added and the HStore field is updated. By default, all unique keys are added.

The PostgreSQL HStore is a simple key-value store used in PostgreSQL and GDAL (when reading an OSM file with the `other_tags` field.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
HStore field	`FIELD`	[campo da tabela: qualquer]	The field(s) to drop

continua na próxima página

Tabela 24.827 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Lista esperada de campos separados por vírgula Opcional	`EXPECTED_FIELDS`	[string] Padrão: ''	Comma-separated list of fields to extract. The HStore field will be updated by removing these keys.
Explodido	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Especifique a camada vetorial de saída. Uma de: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Explodido	`OUTPUT`	[o mesmo que entrada]	Camada de vetor de saída

Código Python

Algorithm ID: `native:explodehstorefield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair campo binário

Extracts contents from a binary field, saving them to individual files. Filenames can be generated using values taken from an attribute in the source table or based on a more complex expression.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada de vetor de entrada contendo os dados binários
Campo binário	`FIELD`	[campo da tabela: qualquer]	Campo contendo os dados binários
Nome do arquivo	`FILENAME`	[expressão]	Field or expression-based text to name each output file

continua na próxima página

Tabela 24.829 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Pasta de destino	`FOLDER`	[pasta] Padrão: `[Salvar em pasta temporária]`	Folder in which to store the output files. *One of*: • Salvar como Diretório Temporário • Salvar no Diretório

Saídas

Etiqueta	Nome	Tipo	Descrição
Pasta	`FOLDER`	[pasta]	A pasta que contém os arquivos de saída.

Código Python

Algorithm ID: `native:extractbinary`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Calculadora de campo

Opens the field calculator (see *Expressões*). You can use all the supported expressions and functions.

Uma nova camada é criada com o resultado da expressão.

The field calculator is very useful when used in *The model designer*.

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada a ser calculada em
Nome do campo de saída	`FIELD_NAME`	[string]	O nome do campo para os resultados

continua na próxima página

Tabela 24.831 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de campo de saída	`FIELD_TYPE`	[enumeração] Padrão: 0	O tipo do campo. Um de: • 0 — Decimal (double) • 1 — Integer (32 bit) • 2 — Text (string) • 3 — Data • 4 — Time • 5 — Date & Time • 6 — Boolean • 7 — Binary Object (BLOB) • 8 — String List • 9 — Integer List • 10 — Decimal (double) List
Largura do campo de saída	`FIELD_LENGTH`	[numeric: integer] Padrão: 0	O comprimento do campo de resultado (mínimo 0)
Precisão do campo	`FIELD_PRECISIO`	[numeric: integer] Padrão: 0	A precisão do campo de resultado (mínimo 0, máximo 15)
Criar novo campo	`NEW_FIELD`	[boleano] Padrão: Verdadeiro	O campo de resultado deve ser um novo campo
Fórmula	`FORMULA`	[expressão]	A fórmula a ser usada para calcular o resultado
Calculado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Especificação da camada de saída. • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Calculado	`OUTPUT`	[o mesmo que entrada]	Camada de saída com os valores de campo calculados

Código Python

Algorithm ID: `native:fieldcalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Editar campos

Permite editar a estrutura da tabela de atributos de uma camada vetorial.

Os campos podem ser modificados em seu tipo e nome, utilizando um mapeamento de campos.

A camada original não é modificada. Uma nova camada é gerada, que contém uma tabela de atributos modificados, de acordo com o mapeamento de campos fornecidos.

Nota

When using a template layer with *constraints* on fields, the information is displayed in the widget with a coloured background and tooltip. Treat this information as a hint during configuration. No constraints will be added on an output layer nor will they be checked or enforced by the algorithm.

O algoritmo de campos de refatoração permite:

- Alterar nomes e tipos de campo
- Adicionar e remover campos
- Reordenar campos
- Calcular novos campos baseado nas expressões
- Carregar lista de campos de outra camada

Nota

This algorithm uses ellipsoid based measurements and respects the current *ellipsoid settings*.

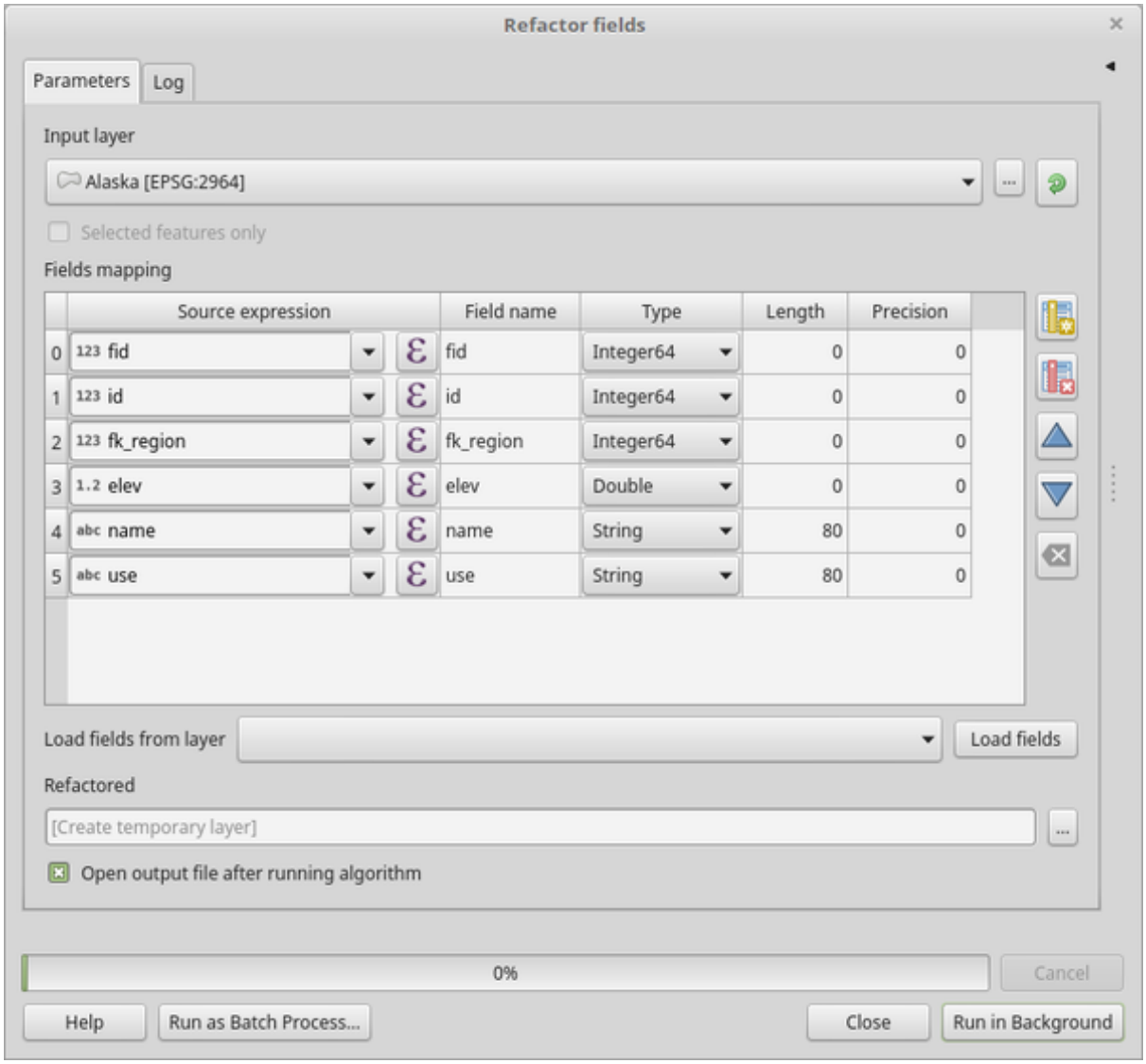


Figura24.165: Refactor fields dialog

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada para modificar

continua na próxima página

Tabela 24.833 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Fields mapping	`FIELDS_MAPPING`	[lista]	Lista de campos de saída com suas definições. A tabela incorporada lista todos os campos da camada de origem e permite editá-los: • Clique em **|novoAtributo|** para criar um novo campo. • Clique em **|excluirAtributo|** para remover um campo. • Use and to change the selected field order. • Click to reset to the default view. • Click to invert the selection in the fields list. Para cada um dos campos que deseja reutilizar, você precisa preencher as seguintes opções: ***Source expression*** **(`expression`) [expression]** Campo ou expressão da camada de entrada ***Name*** **(`name`) [string]** Nome do campo na camada de saída. Por padrão, o nome do campo de entrada é mantido. ***Tipo*** **(`tipo``) [enumeração]** Data type of the output field. Available types may not be compatible with the output layer provider. One of: **Atenção** For certain field types, e.g. lists, an extra `sub_type` parameter helps refine the specific type of the data. It is automatically set in the GUI but may be needed if you're running the algorithm in Python or from the command line. • 1 — Booleano • 2 — Integer (32bit) • 4 — Integer (64bit) • 6 — Decimal (double) • 9 — Integer list • 9 — Integer (64bit) list • 9 — Decimal (double) list • 10 — Text (string) • 11 — String list • 12 — Binary Object (BLOB) • 14 — Data • 15 — Time • 16 — Date & Time ***Sub-type*** **(`sub_type`) [enumeration]** For certain field types, e.g. lists, this parameter helps refine the specific `type` of the data. It is automatically set in the GUI but may be needed if you're running the algorithm in Python or from the command line. *One of*:

Tabela 24.833 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Refatorado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specification of the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Refatorado	`OUTPUT`	[o mesmo que entrada]	Camada de saída com campos refatorados

Código Python

Algorithm ID: `native:refactorfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Renomear campo

Renomeia um campo existente de uma camada vetorial.

A camada original não é modificada. Uma nova camada é gerada onde a tabela de atributos contém o campo renomeado.

Ver também

Editar campos

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada vetorial de entrada
Campo para renomear	`FIELD`	[campo da tabela: qualquer]	O campo a ser alterado
Novo nome de campo	`NEW_NAME`	[string]	O novo nome do campo
Renomeado	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specification of the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Renomeado	`OUTPUT`	[o mesmo que entrada]	Camada de saída com o campo renomeado

Código Python

Algorithm ID: `native:renametablefield`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Campos retidos

Pega uma camada vetorial e gera uma nova que retém apenas os campos selecionados. Todos os outros campos serão descartados.

Ver também

Descartar campo(s)

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada vetorial de entrada
Campos a reter	`FIELDS`	[campo da tabela: qualquer][list]	Lista de campos para manter na camada
Campos retidos	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar camada temporária]`	Specification of the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Campos retidos	`OUTPUT`	[o mesmo que entrada]	Camada de saída com os campos retidos

Código Python

Algorithm ID: `native:retainfields`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Texto para flutuante

Modifies the type of a given attribute in a vector layer, converting a text attribute containing numeric strings into a numeric attribute (e.g. '1' to `1.0`).

The algorithm creates a new vector layer so the source one is not modified.

If the conversion is not possible the selected column will have `NULL` values.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	A camada vetorial de entrada.
Text attribute to convert to float	`FIELD`	[tablefield: string]	The string field for the input layer that is to be converted to a float field.
Float from text	`OUTPUT`	[o mesmo que entrada] Padrão: `[Criar Camada Temporária]`	Specify the output layer. *One of*: • Criar Camada Temporária (`TEMPORARY_OUTPUT`) • Salvar em Arquivo… • Salvar para Geopackage… • Salvar na Tabela de Banco de Dados… • Anexar à Camada… A codificação do arquivo também pode ser alterada aqui.

Saídas

Etiqueta	Nome	Tipo	Descrição
Float from text	`OUTPUT`	[o mesmo que entrada]	Output vector layer with the string field converted into a float field

Código Python

Algorithm ID: `qgis:texttofloat`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.1.30 Vector Tiles

Download vector tiles

Downloads vector tiles of an input vector tile layer and saves them in a local vector tile file.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector tiles]	A vector tile layer to extract some tiles from
Extensão	`EXTENT`	[extensão]	Specify the spatial extent of the downloaded area. It will internally be extended to a multiple of the tile size. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Maximum zoom level to download	`MAX_ZOOM`	[numeric: integer] Padrão: 10	Defines how far to zoom in and fetch data from the tiles
Tile limit	`TILE_LIMIT`	[numeric: integer] Padrão: 100	Maximum number of tiles to download, considering the zoom levels and the extent.
Saída	`OUTPUT`	[vector tiles] Padrão: [Salvar em arquivo temporário]	Specification of the output vector tile file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Saída	`OUTPUT`	[vector tiles]	A local vector tile file storing the downloaded tiles.

Código Python

Algorithm ID: `native:downloadvectortiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Write vector tiles (MBTiles)

Exports one or more vector layers to vector tiles, a data format optimized for fast map rendering and small data size.

MBTiles is a specification for storing tiled map data in SQLite databases for immediate usage and for transfer. MBTiles files are known as tilesets.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[vetor: qualquer] [lista]	Uma lista de camadas a serem combinadas para gerar os blocos vetoriais
Nível mínimo de zoom	`MIN_ZOOM`	[numeric: integer] Padrão: 0	The lowest zoom level for which the tileset provides data. Set between 0 and 24.
Nível máximo de zoom	`MAX_ZOOM`	[numeric: integer] Padrão: 3	The highest zoom level for which the tileset provides data. Set between 0 and 24.
Extensão Opcional	`EXTENT`	[extensão] Padrão: Não definido	The maximum extent of the rendered map area. Bounds must define an area covered by all zoom levels.
Metadado: Nome Opcional	`META_NAME`	[string]	Name of the tileset
Metadados: Descrição Opcional	`META_DESCRIPTI`	[string]	A description of the tileset's contents
Metadados: Atribuição Opcional	`META_ATTRIBUTI`	[string]	Uma string de atribuição, que explica as fontes de dados e/ou estilo do mapa.
Metadados: Versão Opcional	`META_VERSION`	[string]	The version of the tileset. This refers to a revision of the tileset itself, not of the MBTiles specification.
Metadados: Tipo Opcional	`META_TYPE`	[string]	Type of tileset. Possible values are `overlay` or `baselayer`.
Metadados: Centro Opcional	`META_CENTER`	[string]	The center (string of comma-separated numbers: the longitude, latitude, and zoom level) of the default view of the map. Example: `-122.1906,37.7599,11`
Destination MBTiles	`OUTPUT`	[vector tiles] Padrão: [Salvar em arquivo temporário]	Specification of the output MBTiles file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Destination MBTiles	`OUTPUT`	[arquivo]	Output vector tiles `.mbtiles` file.

Código Python

Algorithm ID: `native:writevectortiles_mbtiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Write vector tiles (XYZ)

Exports one or more vector layers to vector tiles, a data format optimized for fast map rendering and small data size.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Modelo de arquivo	`XYZ_TEMPLATE`	[string] Padrão: '{z}/{x}/{y}.pbf'	Template to generate the vector tiles url
Camadas de entrada*	`INPUT`	[vetor: qualquer] [lista]	Uma lista de camadas a serem combinadas para gerar os blocos vetoriais
Nível mínimo de zoom	`MIN_ZOOM`	[numeric: integer] Padrão: 0	The lowest zoom level for which the tileset provides data. Set between 0 and 24.
Nível máximo de zoom	`MAX_ZOOM`	[numeric: integer] Padrão: 3	The highest zoom level for which the tileset provides data. Set between 0 and 24.
Extensão Opcional	`EXTENT`	[extensão] Padrão: Não definido	The maximum extent of the rendered map area. Bounds must define an area covered by all zoom levels.
Diretório de saída	`OUTPUT_DIRECTORY`	[pasta] Padrão: [Salvar em pasta temporária]	Specification of the output vector tiles folder. *One of*: • Salvar como Diretório Temporário • Salvar no Diretório

Saídas

Etiqueta	Nome	Tipo	Descrição
Diretório de saída	`OUTPUT_DIRECTORY`	[pasta]	A folder containing different subsets of the vector tiles files (`.pbf`) stored in subfolders corresponding to the zoom levels.

Código Python

Algorithm ID: `native:writevectortiles_xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.2 Provedor de algoritmo geoespacial GDAL

GDAL (Geospatial Data Abstraction Library) is a translator library for raster and vector geospatial data formats. Algorithms in the Processing Framework are derived from the GDAL raster programs and GDAL vector programs.

24.2.1 Análise de dados Raster

Aspecto

Gera um mapa de aspecto de qualquer elevação raster suportada pelo GDAL. Aspecto é a direção da bússola que um declive enfrenta. Os píxels terão um valor de 0-360° medidos em graus a partir do norte indicando o azimute. No hemisfério norte, o lado norte dos declives é frequentemente sombreado (pequeno azimute de 0-90°), enquanto o lado sul recebe mais radiação solar (azimute maior de 180-270°).

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster* ► *Analysis*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	O número da banda a ser usada como elevação
Retorna ângulo trigonométrico em vez de azimute	`TRIG_ANGLE`	[boleano] Padrão: Falso	A ativação do ângulo trigonométrico resulta em diferentes categorias: 0° (Leste), 90° (Norte), 180° (Oeste), 270° (Sul).
Return 0 for flat instead of -9999	`ZERO_FLAT`	[boleano] Padrão: Falso	Ativando esta opção será inserido um valor 0 para o valor -9999 em áreas planas.
Computar arestas	`COMPUTE_EDGES`	[boleano] Padrão: Falso	Gera arestas a partir do raster de elevação
Use Zevenbergen&Thorne formula instead of the Horn's one	`ZEVENBERGEN`	[boleano] Padrão: Falso	Activates Zevenbergen&Thorne formula for smooth landscapes

continua na próxima página

Tabela 24.847 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Aspecto	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Aspecto	`OUTPUT`	[raster]	Raster de saída com valores de ângulo em graus

Código Python

Algorithm ID: `gdal:aspect`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Cor do relevo

Gera um mapa de relevo colorido a partir de qualquer elevação raster suportada pelo GDAL. Os relevos coloridos podem ser usados particularmente para retratar elevações. A saída do algoritmo produz uma imagem raster de 4 bandas com valores calculados a partir da elevação e um arquivo de configuração de cor baseado em texto. Por padrão, as cores entre os valores de elevação fornecidos são combinadas suavemente e o resultado é uma boa imagem raster com elevação colorida.

This algorithm is derived from the GDAL DEM utility.

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	O número da banda a ser usada como elevação
Computar arestas	`COMPUTE_EDGES`	[boleano] Padrão: Falso	Gera arestas a partir do raster de elevação
Color configuration file	`COLOR_TABLE`	[arquivo]	Um arquivo de configuração de cores baseado em texto
Modo de correspondência	`MATCH_MODE`	[enumeração] Padrão: 2	Um de: • 0 — Use uma correspondência de cores rigorosa • 1 — Usar quádruplos RGBA mais próximos • 2 — Usar suavemente cores misturadas
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL
Cor do relevo	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
Cor do relevo	`OUTPUT`	[raster]	Um raster de saída de 4 bandas

Código Python

Algorithm ID: `gdal:colorrelief`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Fill NoData

Fill raster regions with NoData values by interpolation from edges. The values for the NoData regions are calculated by the surrounding pixel values using inverse distance weighting. After the interpolation a smoothing of the results takes place. Input can be any GDAL-supported raster layer. This algorithm is generally suitable for interpolating missing regions of fairly continuously varying rasters (such as elevation models for instance). It is also suitable for filling small holes and cracks in more irregularly varying images (like airphotos). It is generally not so great for interpolating a raster from sparse point data.

This algorithm is derived from the GDAL fillnodata utility.

Default menu: *Raster* ► *Analysis*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	The band to operate on. NoData values must be represented by the value 0.
Máxima distância (em pixels) para pesquisar valores para interpolar	`DISTANCE`	[numeric: integer] Padrão: 10	O número de pixels a pesquisar em todas as direções para encontrar valores para interpolar a partir
Número de iterações de suavização a serem executadas após a interpolação	`ITERATIONS`	[numeric: integer] Padrão: 0	O número de filtros 3x3 passa a ser executado (0 ou mais) para suavizar os resultados da interpolação.
Validity mask	`MASK_LAYER`	[raster]	Uma camada raster que define as áreas a serem preenchidas.

continua na próxima página

Tabela 24.852 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Preenchido	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Preenchido	`OUTPUT`	[raster]	Raster de saída

Código Python

Algorithm ID: `gdal:fillnodata`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Grid (Data metrics)

Computes some data metrics using the specified window and output grid geometry.

This algorithm is derived from the GDAL grid utility.

Default menu: *Raster* ► *Analysis*

Ver também

GDAL grid tutorial

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de pontos	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Métrica de dados a ser usada	`METRIC`	[enumeração] Padrão: 0	Um de: • 0 — Minimum, minimum value found in grid node search ellipse • 1 — Maximum, maximum value found in grid node search ellipse • 2 — Range, a difference between the minimum and maximum values found in grid node search ellipse • 3 — Contagem, um número de pontos de dados encontrados na elipse de busca da grade de nós • 4 — Average distance, an average distance between the grid node (center of the search ellipse) and all of the data points found in grid node search ellipse • 5 — Average distance between points, an average distance between the data points found in grid node search ellipse. The distance between each pair of points within ellipse is calculated and average of all distances is set as a grid node value
O primeiro raio da elipse de pesquisa	`RADIUS_1`	[numeric: double] Padrão: 0.0	O primeiro raio (eixo X se o ângulo de rotação for 0) da elipse de pesquisa
O segundo raio da elipse de pesquisa	`RADIUS_2`	[numeric: double] Padrão: 0.0	O segundo raio (eixo Y se o ângulo de rotação for 0) da elipse de pesquisa
Ângulo de rotação da elipse de pesquisa em graus (sentido anti-horário)	`ANGLE`	[numeric: double] Padrão: 0.0	Ângulo de rotação da elipse em graus. A elipse é rotacionada no sentido horário.

continua na próxima página

Tabela 24.855 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Número mínimo de pontos de dados a usar	`MIN_POINTS`	[numeric: double] Padrão: 0.0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[numeric: double] Padrão: 0.0	NoData marker to fill empty points
Interpolado (métricas de dados)	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer with interpolated values. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Valor Z do campo Opcional	`Z_FIELD`	[campo de tabela: numérico]	Campo para a interpolação
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

continua na próxima página

Tabela 24.856 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

Saídas

Etiqueta	Nome	Tipo	Descrição
Interpolado (métricas de dados)	`OUTPUT`	[raster]	Raster de saída com valores interpolados

Código Python

Algorithm ID: `gdal:griddatametrics`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Grid (IDW with nearest neighbor searching)

Computes the Inverse Distance to a Power gridding combined to the nearest neighbor method. Ideal when a maximum number of data points to use is required.

This algorithm is derived from the GDAL grid utility.

Ver também

GDAL grid tutorial

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de pontos	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Weighting power	`POWER`	[numeric: double] Padrão: 2.0	Weighting power
Suavização	`SMOOTHING`	[numeric: double] Padrão: 0.0	Parâmetro de suavização
O raio do círculo de busca	`RADIUS`	[numeric: double] Padrão: 1,0	O raio do círculo de busca
Número máximo de pontos de dados a usar	`MAX_POINTS`	[numeric: integer] Padrão: 12	Não pesquise por mais pontos do que este número.
Número mínimo de pontos de dados a usar	`MIN_POINTS`	[numeric: integer] Padrão: 0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[numeric: double] Padrão: 0.0	NoData marker to fill empty points
Interpolated (IDW with NN search)	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer with interpolated values. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Valor Z do campo Opcional	`Z_FIELD`	[campo de tabela: numérico]	Campo para a interpolação
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

continua na próxima página

Tabela 24.859 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

Saídas

Etiqueta	Nome	Tipo	Descrição
Interpolated (IDW with NN search)	`OUTPUT`	[raster]	Raster de saída com valores interpolados

Código Python

Algorithm ID: `gdal:gridinversedistancenearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Grid (Inverse distance to a power)

The Inverse Distance to a Power gridding method is a weighted average interpolator.

You should supply the input arrays with the scattered data values including coordinates of every data point and output grid geometry. The function will compute interpolated value for the given position in output grid.

This algorithm is derived from the GDAL grid utility.

Default menu: *Raster* ► *Analysis*

Ver também

GDAL grid tutorial

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de pontos	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Weighting power	`POWER`	[numeric: double] Padrão: 2.0	Weighting power
Suavização	`SMOOTHING`	[numeric: double] Padrão: 0.0	Parâmetro de suavização
O primeiro raio da elipse de pesquisa	`RADIUS_1`	[numeric: double] Padrão: 0.0	O primeiro raio (eixo X se o ângulo de rotação for 0) da elipse de pesquisa
O segundo raio da elipse de pesquisa	`RADIUS_2`	[numeric: double] Padrão: 0.0	O segundo raio (eixo Y se o ângulo de rotação for 0) da elipse de pesquisa
Ângulo de rotação da elipse de pesquisa em graus (sentido anti-horário)	`ANGLE`	[numeric: double] Padrão: 0.0	Ângulo de rotação da elipse em graus. A elipse é rotacionada no sentido horário.
Número máximo de pontos de dados a usar	`MAX_POINTS`	[numeric: integer] Padrão: 0	Não pesquise por mais pontos do que este número.
Número mínimo de pontos de dados a usar	`MIN_POINTS`	[numeric: integer] Padrão: 0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[numeric: double] Padrão: 0.0	NoData marker to fill empty points
Interpolado (IDW)	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer with interpolated values. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Valor Z do campo Opcional	`Z_FIELD`	[campo de tabela: numérico]	Campo para a interpolação
Opções de criação adicionais Opcional	`CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

Saídas

Etiqueta	Nome	Tipo	Descrição
Interpolado (IDW)	`OUTPUT`	[raster]	Raster de saída com valores interpolados

Código Python

Algorithm ID: `gdal:gridinversedistance`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Grade (Linear)

The Linear method perform linear interpolation by computing a Delaunay triangulation of the point cloud, finding in which triangle of the triangulation the point is, and by doing linear interpolation from its barycentric coordinates within the triangle. If the point is not in any triangle, depending on the radius, the algorithm will use the value of the nearest point or the NoData value.

This algorithm is derived from the GDAL grid utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de pontos	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
Distância de busca	`RADIUS`	[numeric: double] Padrão: -1,0	In case the point to be interpolated does not fit into a triangle of the Delaunay triangulation, use that maximum distance to search a nearest neighbour, or use NoData otherwise. If set to `-1`, the search distance is infinite. If set to `0`, NoData value will be used.
NoData	`NODATA`	[numeric: double] Padrão: 0.0	NoData marker to fill empty points
Interpolado (Linear)	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer with interpolated values. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

continua na próxima página

Tabela 24.865 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Valor Z do campo Opcional	`Z_FIELD`	[campo de tabela: numérico]	Campo para a interpolação
Opções de criação adicionais Opcional	`CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

Saídas

Etiqueta	Nome	Tipo	Descrição
Interpolado (Linear)	`OUTPUT`	[raster]	Raster de saída com valores interpolados

Código Python

Algorithm ID: `gdal:gridlinear`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Grid (média móvel)

The Moving Average is a simple data averaging algorithm. It uses a moving window of elliptic form to search values and averages all data points within the window. Search ellipse can be rotated by specified angle, the center of ellipse located at the grid node. Also the minimum number of data points to average can be set, if there are not enough points in window, the grid node considered empty and will be filled with specified NoData value.

This algorithm is derived from the GDAL grid utility.

Default menu: *Raster* ► *Analysis*

Ver também

GDAL grid tutorial

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de pontos	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
O primeiro raio da elipse de pesquisa	`RADIUS_1`	[numeric: double] Padrão: 0.0	O primeiro raio (eixo X se o ângulo de rotação for 0) da elipse de pesquisa
O segundo raio da elipse de pesquisa	`RADIUS_2`	[numeric: double] Padrão: 0.0	O segundo raio (eixo Y se o ângulo de rotação for 0) da elipse de pesquisa
Ângulo de rotação da elipse de pesquisa em graus (sentido anti-horário)	`ANGLE`	[numeric: double] Padrão: 0.0	Ângulo de rotação da elipse em graus. A elipse é rotacionada no sentido horário.

continua na próxima página

Tabela 24.867 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Número mínimo de pontos de dados a usar	`MIN_POINTS`	[numeric: integer] Padrão: 0	Minimum number of data points to average. If less amount of points found the grid node considered empty and will be filled with NoData marker.
NoData	`NODATA`	[numeric: double] Padrão: 0.0	NoData marker to fill empty points
Interpolado (média móvel)	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Valor Z do campo Opcional	`Z_FIELD`	[campo de tabela: numérico]	Campo para a interpolação
Opções de criação adicionais Opcional	`CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

continua na próxima página

Tabela 24.868 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

Saídas

Etiqueta	Nome	Tipo	Descrição
Interpolado (média móvel)	`OUTPUT`	[raster]	Raster de saída com valores interpolados

Código Python

Algorithm ID: `gdal:gridaverage`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Grade (vizinho mais próximo)

The Nearest Neighbor method doesn't perform any interpolation or smoothing, it just takes the value of nearest point found in grid node search ellipse and returns it as a result. If there are no points found, the specified NoData value will be returned.

This algorithm is derived from the GDAL grid utility.

Default menu: *Raster* ► *Analysis*

Ver também

GDAL grid tutorial

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de pontos	`INPUT`	[vetor: ponto]	Insira uma camada de ponto vetorial.
O primeiro raio da elipse de pesquisa	`RADIUS_1`	[numeric: double] Padrão: 0.0	O primeiro raio (eixo X se o ângulo de rotação for 0) da elipse de pesquisa
O segundo raio da elipse de pesquisa	`RADIUS_2`	[numeric: double] Padrão: 0.0	O segundo raio (eixo Y se o ângulo de rotação for 0) da elipse de pesquisa
Ângulo de rotação da elipse de pesquisa em graus (sentido anti-horário)	`ANGLE`	[numeric: double] Padrão: 0.0	Ângulo de rotação da elipse em graus. A elipse é rotacionada no sentido horário.
NoData	`NODATA`	[numeric: double] Padrão: 0.0	NoData marker to fill empty points
Interpolado (Vizinho mais próximo)	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer with interpolated values. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Valor Z do campo Opcional	`Z_FIELD`	[campo de tabela: numérico]	Campo para a interpolação
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

continua na próxima página

Tabela 24.871 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

Saídas

Etiqueta	Nome	Tipo	Descrição
Interpolado (Vizinho mais próximo)	`OUTPUT`	[raster]	Raster de saída com valores interpolados

Código Python

Algorithm ID: `gdal:gridnearestneighbor`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Sombreamento

Outputs a raster with a nice shaded relief effect. It's very useful for visualizing the terrain. You can optionally specify the azimuth and altitude of the light source, a vertical exaggeration factor and a scaling factor to account for differences between vertical and horizontal units.

This algorithm is derived from the GDAL DEM utility .

Default menu: *Raster* ► *Analysis*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	Banda contendo as informações de elevação
Fator Z (exagero vertical)	`Z_FACTOR`	[numeric: double] Padrão: 1,0	The factor exaggerates the height of the output elevation raster
Scale (ratio of vertical units to horizontal)	`SCALE`	[numeric: double] Padrão: 1,0	A proporção de unidades verticais para unidades horizontais
Azimute da luz	`AZIMUTH`	[numeric: double] Padrão: 315,0	Defines the azimuth of the light shining on the elevation raster in degrees. If it comes from the top of the raster the value is 0, if it comes from the east it is 90 a.s.o.
Altitude da luz	`ALTITUDE`	[numeric: double] Padrão: 45.0	Defines the altitude of the light, in degrees. 90 if the light comes from above the elevation raster, 0 if it is raking light.
Computar arestas	`COMPUTE_EDGES`	[boleano] Padrão: Falso	Gera arestas a partir do raster de elevação
Use Zevenbergen&Thorne formula (instead of the Horn's one)	`ZEVENBERGEN`	[boleano] Padrão: Falso	Activates Zevenbergen&Thorne formula for smooth landscapes
Sombreamento combinado	`COMBINED`	[boleano] Padrão: Falso	
Sombreamento multidirecional	`MULTIDIRECTIONAL`	[boleano] Padrão: Falso	
Sombreamento	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer with interpolated values. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Sombreamento	`OUTPUT`	[raster]	Raster de saída com valores interpolados

Código Python

ID do algoritmo: `gdal:sombra`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Quase preto

Converte bordas quase pretas/brancas em preto.

Esse algoritmo digitalizará uma imagem e tentará definir todos os pixels que são quase ou exatamente pretos, brancos ou uma ou mais cores personalizadas ao redor do colar para preto ou branco. Isso geralmente é usado para "consertar" fotos de ar compactadas com perdas para que os pixels coloridos possam ser tratados como transparentes ao mosaico.

This algorithm is derived from the GDAL nearblack utility.

Default menu: *Raster ► Analysis*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Quão longe do preto (branco)	`NEAR`	[numeric: integer] Padrão: 15	Selecione o quão longe de preto, branco ou cores personalizadas os valores de pixel podem ser e ainda considerados próximos a preto, branco ou cores personalizadas.
Pesquise por pixels quase brancos em vez de quase pretos	`WHITE`	[boleano] Padrão: Falso	Pesquise por pixels quase brancos (255) em vez de pixels quase pretos
Quase preto	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ‘’	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Quase preto	`OUTPUT`	[raster]	Raster de saída

Código Python

ID do algoritmo: `gdal:quasepreto`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Proximidade (distância raster)

Gera um mapa de proximidade raster indicando a distância do centro de cada pixel até o centro do pixel mais próximo identificado como um pixel alvo. Os pixels de destino são aqueles no raster de origem para os quais o valor do pixel do raster está no conjunto de valores de pixel de destino.

This algorithm is derived from the GDAL proximity utility.

Default menu: *Raster* ► *Analysis*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	Banda contendo as informações de elevação
List of target pixels Opcional	`VALUES`	[string] Padrão: ‘’	Comma-separated list of pixel values in the source image to consider as target pixels. If not specified, all non-zero pixels will be considered target pixels.
Unidades de distância	`UNITS`	[enumeração] Padrão: 1	Indique se as distâncias geradas devem ser em pixels ou coordenadas georreferenciadas. Um de: • 0 — Coordenadas georreferenciadas • 1 — Coordenadas do pixel
The maximum distance to generate Opcional	`MAX_DISTANCE`	[numeric: double] Padrão: 0.0	The maximum distance to be generated. The NoData value will be used for pixels beyond this distance. Distance is interpreted according to the value of the *Distance units* parameter.
Value to apply to pixels within the maximum distance Opcional	`REPLACE`	[numeric: double] Padrão: 0.0	Specify a value to apply to all pixels that are within the maximum distance from target pixels (including the target pixels) instead of a distance value.

continua na próxima página

Tabela 24.879 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
NoData value to use for the destination proximity raster Opcional	`NODATA`	[numeric: double] Padrão: 0.0	Specify the NoData value to use for the pixels beyond the maximum distance. If a NoData value is not provided, the output band will be queried for its NoData value. If the output band does not have a NoData value, then the value `65535` will be used.
Mapa de proximidade	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

continua na próxima página

Tabela 24.880 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

Saídas

Etiqueta	Nome	Tipo	Descrição
Mapa de proximidade	`OUTPUT`	[raster]	Raster de saída

Código Python

Algorithm ID: `gdal:proximity`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Rugosidade

Outputs a single-band raster with values computed from the elevation. Roughness is the degree of irregularity of the surface. It's calculated by the largest inter-cell difference of a central pixel and its surrounding cell. The determination of the roughness plays a role in the analysis of terrain elevation data, it's useful for calculations of the river morphology, in climatology and physical geography in general.

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster* ► *Analysis*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	O número da banda a ser usada como elevação
Computar arestas	`COMPUTE_EDGES`	[boleano] Padrão: Falso	Gera arestas a partir do raster de elevação
Rugosidade	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Rugosidade	`OUTPUT`	[raster]	Single-band output roughness raster. The value -9999 is used as NoData value.

Código Python

Algorithm ID: `gdal:roughness`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Sieve

Removes raster polygons smaller than a provided threshold size (in pixels) and replaces them with the pixel value of the largest neighbour polygon. It is useful if you have a large amount of small areas on your raster map.

This algorithm is derived from the GDAL sieve utility.

Default menu: *Raster* ► *Analysis*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Threshold.	`THRESHOLD`	[numeric: integer] Padrão: 10	Apenas polígonos matriciais menores que este tamanho serão removidos
Utilizar 8-conexões	`EIGHT_CONNECTE`	[boleano] Padrão: Falso	If False, consider as neighbours only the (4) pixels that connect the edges horizontally or vertically. If True, every pixel that touches an edge or a corner is considered as neighbour.
Do not use the default validity mask for the input band	`NO_MASK`	[boleano] Padrão: Falso	
Validity mask Opcional	`MASK_LAYER`	[raster]	Validity mask to use instead of the default
Sieved	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Parâmetros adicionais da linha de comando Opcional	EXTRA	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Sieved	OUTPUT	[raster]	Camada raster de saída

Código Python

Algorithm ID: `gdal:sieve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Inclinação

Generates a slope map from any GDAL-supported elevation raster. Slope is the angle of inclination to the horizontal. You have the option of specifying the type of slope value you want: degrees or percent slope.

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster* ► *Analysis*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	INPUT	[raster]	Camada raster de elevação de entrada
Número da banda	BAND	[banda raster] Padrão: 1	Banda contendo as informações de elevação
Ratio of vertical units to horizontal	SCALE	[numeric: double] Padrão: 1,0	A proporção de unidades verticais para unidades horizontais
Express slope as percent (instead of degrees)	AS_PERCENT	[boleano] Padrão: Falso	Expressar declive como porcentagem em vez de graus

continua na próxima página

Tabela 24.888 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Computar arestas	`COMPUTE_EDGES`	[boleano] Padrão: Falso	Gera arestas a partir do raster de elevação
Use Zevenbergen&Thorne formula (instead of the Horn's one)	`ZEVENBERGEN`	[boleano] Padrão: Falso	Activates Zevenbergen&Thorne formula for smooth landscapes
Declive	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Declive	`OUTPUT`	[raster]	Raster de saída

Código Python

Algorithm ID: `gdal:slope`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Terrain Ruggedness Index (TRI)

Outputs a single-band raster with values computed from the elevation. TRI stands for Terrain Ruggedness Index, which is defined as the mean difference between a central pixel and its surrounding cells.

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster* ► *Analysis*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	O número da banda a ser usada como elevação
Computar arestas	`COMPUTE_EDGES`	[boleano] Padrão: Falso	Gera arestas a partir do raster de elevação
Terrain Ruggedness Index	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ”	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

Saídas

Etiqueta	Nome	Tipo	Descrição
Terrain Ruggedness Index	`OUTPUT`	[raster]	Output ruggedness raster. The value -9999 is used as NoData value.

Código Python

Algorithm ID: `gdal:triterrainruggednessindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Topographic Position Index (TPI)

Outputs a single-band raster with values computed from the elevation. TPI stands for Topographic Position Index, which is defined as the difference between a central pixel and the mean of its surrounding cells.

This algorithm is derived from the GDAL DEM utility.

Default menu: *Raster* ► *Analysis*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	O número da banda a ser usada para valores de elevação
Computar arestas	`COMPUTE_EDGES`	[boleano] Padrão: Falso	Gera arestas a partir do raster de elevação
Terrain Ruggedness Index	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ”	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

Saídas

Etiqueta	Nome	Tipo	Descrição
Terrain Ruggedness Index	`OUTPUT`	[raster]	Matriz de saída.

Código Python

Algorithm ID: `gdal:tpitopographicpositionindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.2.2 Conversão de raster

gdal2xyz

Converte dados raster em formato de arquivo ASCII XYZ

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster para converter
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada de entrada	Se o raster for multibandas, escolha a banda que você quer converter
Input pixel value to treat as NoData Opcional	`NODATA_INPUT`	[numeric: integer] Padrão: Não definido	Input pixel value to treat as "NoData" **Atenção** Using this parameter requires QGIS installed with GDAL >= 3.7 (see *Help* ► *About* menu).
Destination NoData Opcional	`NODATA_OUTPUT`	[numeric: integer] Padrão: Não definido	Assign specified "NoData" value to output. **Atenção** Using this parameter requires QGIS installed with GDAL >= 3.7 (see *Help* ► *About* menu).

continua na próxima página

Tabela 24.897 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Do not output NoData values	`SKIP_NODATA`	[boleano] Padrão: Falso	Do not output "NoData" values. **Atenção** Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* ► *About* menu).
Valores separados por vírgula	`CSV`	[boleano] Padrão: Falso	Define se o arquivo de saída deve ser do tipo valores separados por vírgula (csv).
arquivo ASCII XYZ	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
arquivo ASCII XYZ	`INPUT`	[tabela]	Arquivo de tabela que contém os valores exportados da banda raster

Código Python

ID do algoritmo: `gdal:gdal2xyz`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

PCT para RGB

Converte uma imagem paletada de 8 bits para uma RGB de 24 bits. Converte uma banda pseudo-colorida do arquivo de entrada para um arquivo RGB do formato desejado.

This algorithm is derived from the GDAL pct2rgb utility.

Menu padrão: *Raster* ► *Conversão*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Imagem raster de 8 bits de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada de entrada	Se o raster for multibandas, escolha a banda que você quer converter
Gerar um arquivo RGBA	`RGBA`	[boleano] Padrão: Falso	Define se o arquivo de saída deve ser do tipo RGBA.
PCT para RGB	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Specification of the output file. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
PCT para RGB	`OUTPUT`	[raster]	Imagem raster RGB de 24 bits

Código Python

ID do algoritmo: `gdal:pcttorgb`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Poligonizar (raster para vetor)

Cria polígonos vetoriais para todas as regiões conectadas de pixels no raster que compartilha um valor de pixel comum. Cada polígono é criado com um atributo que indica o valor de pixel desse polígono.

This algorithm is derived from the GDAL polygonize utility.

Menu padrão: *Raster ► Conversão*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: A primeira faixa da camada de entrada	Se o raster for multibandas, escolha a banda que você quer usar
Nome do campo a ser criado	`FIELD`	[string] Padrão: 'DN'	Especifique o nome do campo para os atributos das regiões conectadas.
Utilizar 8-conexões	`EIGHT_CONNECTE`	[boleano] Padrão: Falso	Se não estiver definido, as células raster devem ter uma fronteira comum para serem consideradas conectadas (*4-conectadas*). Se definidas, as células raster que se tocam também são consideradas conectadas (*8-conectadas*).
Vetorizado	`OUTPUT`	[vetor: polígono] Padrão: `[Salvar em arquivo temporário]`	Specification of the output (polygon) vector layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Vetorizado	`OUTPUT`	[vetor: polígono]	Camada de vetor de saída

Código Python

Algoritmo ID: `gdal:polygonize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Reorganizar bandas

Cria um novo raster usando banda (s) selecionada de uma determinada camada raster. O algoritmo também torna possível reordenar as bandas para o raster recém-criado.

This algorithm is derived from the GDAL translate utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Selecionar banda(s)	`BANDS`	[banda raster] [lista] Padrão: Não definido	Lista ordenada das faixas a serem usadas para criar o novo raster
Convertido	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).

continua na próxima página

Tabela 24.905 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 0	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Usar o tipo de dados da camada de entrada • 1 — Byte (Eight bit unsigned integer (quint8)) • 2 — Int16 (Sixteen bit signed integer (qint16)) • 3 — UInt16 (Sixteen bit unsigned integer (quint16)) • 4 — UInt32 (Thirty two bit unsigned integer (quint32)) • 5 — Int32 (Thirty two bit signed integer (qint32)) • 6 — Float32 (Thirty two bit floating point (float)) • 7 — Float64 (Sixty four bit floating point (double)) • 8 — CInt16 (Complex Int16) • 9 — CInt32 (Complex Int32) • 10 — CFloat32 (Complex Float32) • 11 — CFloat64 (Complex Float64) • 12 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

Saídas

Etiqueta	Nome	Tipo	Descrição
Convertido	`OUTPUT`	[raster]	Camada raster de saída com bandas rearranjadas.

Código Python

Algoritmo ID: `gdal:rearrange_bands`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

RGB para PCT

Converte uma imagem RGB de 24 bits em uma paleta de 8 bits. Computa uma tabela pseudo-colorida ideal para uma dada imagem RGB usando um algoritmo de corte mediano em um histograma RGB abaixo exemplificado. Depois, converte a imagem em uma imagem pseudo-colorida usando a tabela de cores. Esta conversão utiliza o dithering de Floyd-Steinberg (difusão de erros) para maximizar a qualidade visual da imagem de saída.

Se você quiser classificar um mapa raster e quiser reduzir o número de classes que pode ser útil para downsample sua imagem com este algoritmo antes.

This algorithm is derived from the GDAL rgb2pct utility.

Menu padrão: *Raster ► Conversão*

Parâmetros

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster (RGB) de entrada
Número de cores	`NCOLORS`	[numeric: integer] Padrão: 2	O número de cores que a imagem resultante conterá. Um valor de 2-256 é possível.
RGB para PCT	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Saídas

Etiqueta	Nome	Tipo	Descrição
RGB para PCT	`OUTPUT`	[raster]	Camada raster de saída

Código Python

ID do algoritmo: `gdal:rgbtopct`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Traduzir (converter formato)

Converte dados raster entre diferentes formatos.

This algorithm is derived from the GDAL translate utility.

Menu padrão: *Raster ► Conversão*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Sobreescrever a projeção do arquivo de saída Opcional	`TARGET_CRS`	[src]	Especificar uma projeção para o arquivo de saída
Assign a specified NoData value to output bands Opcional	`NODATA`	[numeric: double] Padrão: Não definido	Defines the value to use for NoData in the output raster
Copiar todos os subconjuntos deste arquivo para arquivos de saída individuais	`COPY_SUBDATASE`	[boleano] Padrão: Falso	Criar arquivos individuais para subconjuntos de dados
Convertido	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output (translated) raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

continua na próxima página

Tabela 24.910 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 0	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Usar o tipo de dados da camada de entrada • 1 — Byte (Eight bit unsigned integer (quint8)) • 2 — Int16 (Sixteen bit signed integer (qint16)) • 3 — UInt16 (Sixteen bit unsigned integer (quint16)) • 4 — UInt32 (Thirty two bit unsigned integer (quint32)) • 5 — Int32 (Thirty two bit signed integer (qint32)) • 6 — Float32 (Thirty two bit floating point (float)) • 7 — Float64 (Sixty four bit floating point (double)) • 8 — CInt16 (Complex Int16) • 9 — CInt32 (Complex Int32) • 10 — CFloat32 (Complex Float32) • 11 — CFloat64 (Complex Float64) • 12 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

Saídas

Etiqueta	Nome	Tipo	Descrição
Convertido	`OUTPUT`	[raster]	Saída (traduzida) da camada raster.

Código Python

ID do algoritmo: `gdal:traduzir`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.2.3 Extrair raster

Recortar raster extensão

Clips qualquer GDAL suportado raster arquivo para um dado extensão

This algorithm is derived from the GDAL translate utility.

Default menu: *Raster ► Extraction*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	A matriz de saída
Clipping extent	`EXTENT`	[extensão]	Extent that should be used for the output raster. Only pixels within the specified bounding box will be included in the output. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Override the projection for the output file	`OVERCRS`	[boleano] Padrão: Falso	If checked, the output file is assigned the input layer CRS.
Assign a specified NoData value to output bands Opcional	`NODATA`	[numeric: double] Padrão: Não definido	Defines a value that should be inserted for the NoData values in the output raster
Clipped (extent)	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 0	Define o formato do arquivo de raster saída. Opções: • 0 — Usar o tipo de dados da camada de entrada • 1 — Byte (Eight bit unsigned integer (quint8)) • 2 — Int16 (Sixteen bit signed integer (qint16)) • 3 — UInt16 (Sixteen bit unsigned integer (quint16)) • 4 — UInt32 (Thirty two bit unsigned integer (quint32)) • 5 — Int32 (Thirty two bit signed integer (qint32)) • 6 — Float32 (Thirty two bit floating point (float)) • 7 — Float64 (Sixty four bit floating point (double)) • 8 — CInt16 (Complex Int16) • 9 — CInt32 (Complex Int32) • 10 — CFloat32 (Complex Float32) • 11 — CFloat64 (Complex Float64) • 12 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Clipped (extent)	`OUTPUT`	[raster]	Output raster layer clipped by the given extent

Código Python

Algorithm ID: `gdal:cliprasterbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Recortar raster por camada de máscara

Clips qualquer raster com suporte a GDAL por uma camada de máscara de Vetorial.

This algorithm is derived from the GDAL warp utility.

Default menu: *Raster* ► *Extraction*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	A matriz de saída
Camada de máscara	`MASK`	[vetor: polígono]	Vector mask for clipping the raster
SRC de Origem	`SOURCE_CRS`	[src]	Definir a referência de coordenada a ser usada para a matriz de entrada
SRC Alvo	`TARGET_CRS`	[src]	Set the coordinate reference to use for the mask layer

continua na próxima página

Tabela 24.915 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Target extent Opcional	TARGET_EXTENT	[extensão]	Extent of the output file to be created Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Assign a specified NoData value to output bands Opcional	NODATA	[numeric: double] Padrão: Não definido	Defines a value that should be inserted for the NoData values in the output raster
Criar uma banda alfa de saída	ALPHA_BAND	[boleano] Padrão: Falso	Cria uma banda alfa para o resultado, resulta. A banda alfa inclui os valores de transparência dos pixels.
Match the extent of the clipped raster to the extent of the mask layer	CROP_TO_CUTLIN	[boleano] Padrão: Verdadeiro	Aplica a extensão da camada vetorial à saída raster se checado.
Mantenha a resolução do raster de entrada	KEEP_RESOLUTIO	[boleano] Padrão: Falso	A resolução do raster de saída não será alterada
Set output file resolution	SET_RESOLUTION	[boleano] Padrão: Falso	Shall the output resolution (cell size) be specified
Resolução X para as bandas de saída Opcional	X_RESOLUTION	[numeric: double] Padrão: Não definido	The width of the cells in the output raster
Resolução Y para a banda de saída Opcional	Y_RESOLUTION	[numeric: double] Padrão: Não definido	The height of the cells in the output raster
Use multithreaded warping implementation	MULTITHREAD-ING	[boleano] Padrão: Falso	Two threads will be used to process chunks of image and perform input/output operation simultaneously. Note that computation is not multithreaded itself.

continua na próxima página

Tabela 24.915 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Clipped (mask)	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 0	Define o formato do arquivo de raster saída. Opções: • 0 — Usar o tipo de dados da camada de entrada • 1 — Byte (Eight bit unsigned integer (quint8)) • 2 — Int16 (Sixteen bit signed integer (qint16)) • 3 — UInt16 (Sixteen bit unsigned integer (quint16)) • 4 — UInt32 (Thirty two bit unsigned integer (quint32)) • 5 — Int32 (Thirty two bit signed integer (qint32)) • 6 — Float32 (Thirty two bit floating point (float)) • 7 — Float64 (Sixty four bit floating point (double)) • 8 — CInt16 (Complex Int16) • 9 — CInt32 (Complex Int32) • 10 — CFloat32 (Complex Float32) • 11 — CFloat64 (Complex Float64) • 12 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Clipped (mask)	`OUTPUT`	[raster]	Output raster layer clipped by the vector layer

Código Python

Algorithm ID: `gdal:cliprasterbymasklayer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

contorno

Extrai linhas de contorno de qualquer raster de elevação suportada pelo GDAL.

This algorithm is derived from the GDAL contour utility.

Default menu: *Raster ► Extraction*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	Raster band to create the contours from
Intervalo entre curvas de nível	`INTERVAL`	[numeric: double] Padrão: 10.0	Defines the interval between the contour lines in the given units of the elevation raster (minimum value 0)
Nome do atributo (se não definido, nenhum atributo de elevação é anexado) Opcional	`FIELD_NAME`	[string] Padrão: 'ELEV'	Fornece um nome para o atributo no qual colocar a elevação.
Offset from zero relative to which to interpret intervals Opcional	`OFFSET`	[numeric: double] Padrão: 0.0	

continua na próxima página

Tabela 24.918 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Contornos	`OUTPUT`	[vetor: linha] Padrão: `[Salvar em arquivo temporário]`	Specification of the output vector layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Produzir vetor 3D	`CREATE_3D`	[boleano] Padrão: Falso	Força a produção de vetores 3D em vez de 2D. Inclui elevação em todos os vértice.
Tratar todos os valores raster como válidos	`IGNORE_NODATA`	[boleano] Padrão: Falso	Ignores any NoData values in the dataset.
Input pixel value to treat as "No-Data" Opcional	`NODATA`	[numeric: double] Padrão: Não definido	Defines a value that should be inserted for the NoData values in the output raster
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Add extra GDAL command line options. Refer to the corresponding GDAL utility documentation.

Saídas

Etiqueta	Nome	Tipo	Descrição
Contornos	`OUTPUT`	[vetor: linha]	Output vector layer with contour lines

Código Python

Algorithm ID: `gdal:contour`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Contour Polygons

Extracts contour polygons from any GDAL-supported elevation raster.

This algorithm is derived from the GDAL contour utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Raster de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	Raster band to create the contours from
Intervalo entre curvas de nível	`INTERVAL`	[numeric: double] Padrão: 10.0	Defines the interval between the contour lines in the given units of the elevation raster (minimum value 0)
Offset from zero relative to which to interpret intervals Opcional	`OFFSET`	[numeric: double] Padrão: 0.0	
Attribute name for minimum elevation of contour polygon Opcional	`FIELD_NAME_MIN`	[string] Padrão: 'ELEV_MIN'	Provides a name for the attribute in which to put the minimum elevation of contour polygon. If not provided no minimum elevation attribute is attached.
Attribute name for maximum elevation of contour polygon Opcional	`FIELD_NAME_MAX`	[string] Padrão: 'ELEV_MAX'	Provides a name for the attribute in which to put the maximum elevation of contour polygon. If not provided no maximum elevation attribute is attached.
Contornos	`OUTPUT`	[vetor: polígono] Padrão: `[Salvar em arquivo temporário]`	Specification of the output vector layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Produzir vetor 3D	`CREATE_3D`	[boleano] Padrão: Falso	Força a produção de vetores 3D em vez de 2D. Inclui elevação em todos os vértice.
Tratar todos os valores raster como válidos	`IGNORE_NODATA`	[boleano] Padrão: Falso	Ignores any NoData values in the dataset.
Input pixel value to treat as "No-Data" Opcional	`NODATA`	[numeric: double] Padrão: Não definido	Defines a value that should be inserted for the NoData values in the output raster

continua na próxima página

Tabela 24.922 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Parâmetros adicionais da linha de comando Opcional	EXTRA	[string] Padrão: Não definido	Add extra GDAL command line options. Refer to the corresponding GDAL utility documentation.

Saídas

Etiqueta	Nome	Tipo	Descrição
Contornos	OUTPUT	[vetor: polígono]	Output vector layer with contour polygons

Código Python

Algorithm ID: `gdal:contour_polygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.2.4 Miscelânea de Raster

Construir Visão geral (Pirâmides)

Para acelerar o tempo de renderização de visões de camadas as pirâmides podem ser criadas. As visões gerais são cópias de resolução mais baixa dos dados que o QGIS usa dependendo do nível de zoom.

This algorithm is derived from the GDAL addo utility.

Default menu: *Raster ► Miscellaneous*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	INPUT	[raster]	Camada raster de entrada
Remove all existing overviews	CLEAN	[boleano] Padrão: Falso	Removes existing overviews from the raster. By default these are not removed.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Overview levels	`LEVELS`	[string] Padrão: '2 4 8 16'	Define o número de níveis de visão geral calculados pela resolução original da camada de rastreio de entrada. Por padrão, 4 níveis serão levados em consideração.
Método de reamostragem Opcional	`RESAMPLING`	[enumeração] Padrão: 0	Calculates the overviews with a defined resampling method. Possible resampling methods are: • 0 – Vizinho mais próximo (`mais próximo`) • 1 – Média (`média`) • 2 – Gaussiano (`gauss`) • 3 – Cubic (4x4 kernel) (`cubic`) • 4 – Cubic B-Spline (4x4 kernel) (`cubicspline`) • 5 – Lanczos (6x6 kernel) (`lanczos`) • 6 – Average MP (`average_mp`) • 7 – Average in Mag/Phase Space (`average_magphase`) • 8 – Modo (`modo`)
Overviews format Opcional	`FORMAT`	[enumeração] Padrão: 0	The overviews can be stored internally, or externally as GTiff or ERDAS Imagine file. By default the overviews are stored in the output raster. Possible formats methods are: • 0 – Interno (se possível) • 1 – External (GTiff .ovr) • 2 – External (ERDAS Imagine .aux)
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Piramidizado	`OUTPUT`	[raster]	Output raster layer with overviews

Código Python

Algorithm ID: `gdal:overviews`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Construir raster virtual

Cria um VRT (Coleção Virtual) que é um mosaico da lista de rasteres suportados pelo GDAL de entrada. Com um mosaico você pode mesclar vários arquivos raster.

This algorithm is derived from the GDAL buildvrt utility.

Default menu: *Raster* ► *Miscellaneous*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Importar raster suportado GDAL
Resolução	`RESOLUTION`	[enumeração] Padrão: 0	The output resolution of the mosaic. By default the average resolution of the raster files will be chosen. Opções: • 0 — Média (`média`) • 1 — Mais alto (“mais alto”) • 2 — Mais baixo (``mais baixo”)
Place each input file into a separate band	`SEPARATE`	[boleano] Padrão: Falso	Com ‘Verdadeiro’ você pode definir que cada arquivo de raster vai para uma banda separada na faixa da banda VRT.
Permitir diferença de projeção	`PROJ_DIFFERENC`	[boleano] Padrão: Falso	Permite que as bandas de saída tenham diferentes projeções derivadas da projeção das camadas de varredura de entrada.
Virtual	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster layer. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Add alpha mask band to VRT when source raster has none	`ADD_ALPHA`	[boleano] Padrão: Falso	Adicionar canal de máscara alfa para VTR quando não houver nenhum raster
Sobreescrever a projeção para o arquivo de saída Opcional	`ASSIGN_CRS`	[src] Padrão: Não definido	Substituir a projeção do arquivo de saída. Sem reprojeção estar pronta.
Resampling algorithm	`RESAMPLING`	[enumeração] Padrão: 0	The resampling algorithm to use. Options: • 0 — Vizinho mais próximo (`mais próximo`) • 1 — Bilinear (2x2 kernel) (`bilinear`) • 2 — Cubic (4x4 kernel) (`cubic`) • 3 — Cubic B-Spline (4x4 kernel) (`cubicspline`) • 4 — Lanczos (6x6 kernel) (`lanczos`) • 0 — Média (`média``) • 6 — Modo (`modo`)
NoData value(s) for input bands (space separated) Opcional	`SRC_NODATA`	[string] Padrão: Não definido	Space separated NoData value(s) for input band(s)
Parâmetros adicionais da linha de comando	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Virtual	`OUTPUT`	[raster]	Camada raster de saída

Código Python

Algorithm ID: `gdal:buildvirtualraster`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

gdal2tiles

Generates a directory with small tiles and metadata, following the OSGeo Tile Map Service Specification. See also the OpenGIS Web Map Tile Service Implementation Standard. Simple web pages with viewers based on Google Maps, OpenLayers and Leaflet are generated as well. To explore your maps on-line in the web browser, you only need to upload the generated directory onto a web server.

This algorithm also creates the necessary metadata for Google Earth (KML SuperOverlay), in case the supplied map uses `EPSG:4326` projection.

ESRI world files and embedded georeferencing is used during tile generation, but you can publish a picture without proper georeferencing too.

This algorithm is derived from the GDAL gdal2tiles utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	GDAL-supported raster layer.
Tile cutting profile	`PROFILE`	[enumeração] Padrão: 0	Um de: • 0 — Mercator (`mercator`) • 1 — Geodésico (`geodésico`) • 2 — Matriz (`matriz`)
Zoom levels to render Opcional	`ZOOM`	[string] Padrão: ‘’	
Web viewer to generate	`VIEWER`	[enumeração] Padrão: 0	Um de: • 0 — Todos (`todos`) • 1 — GoogleMaps (`google`) • 2 — OpenLayers (`openlayers`) • 3 — Leaflet (`leaflet`) • 4 — Nenhum (`nenhum`)
Título do mapa Opcional	`TITLE`	[string] Padrão: ‘’	
Copyright of the map	`COPYRIGHT`	[string] Padrão: ‘’	
Diretório de saída	`OUTPUT`	[pasta] Padrão: `[Salvar em pasta temporária]`	Specify the output folder for the tiles. *One of*: • Salvar como Diretório Temporário • Salvar no Diretório

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Método de reamostragem	`RESAMPLING`	[enumeração] Padrão: 0	The resampling algorithm to use. Options: • 0 — Média (`média`) • 1 — Nearest Neighbour (`near`) • 2 — Bilinear (2x2 kernel) (`bilinear`) • 3 — Cubic (4x4 kernel) (`cubic`) • 4 — Cubic B-Spline (4x4 kernel) (`cubicspline`) • 5 — Lanczos (6x6 kernel) (`lanczos`) • 6 — Antialias (`antialias`)
The spatial reference system used for the source input data Opcional	`SOURCE_CRS`	[src] Padrão: Não definido	
Transparency value to assign to the input data Opcional	`NODATA`	[numeric: double] Padrão: 0.0	
URL address where the generated tiles are going to be published Opcional	`URL`	[string] Padrão: ‘’	
Google Maps API key (http://code.google.c Opcional	`GOOGLE_KEY`	[string] Padrão: ‘’	Your Google maps API key.
Bing Maps API key (https://www.bingm Opcional	`BING_KEY`	[string] Padrão: ‘’	Your Bing maps API key.
Gerar apenas arquivos faltantes	`RESUME`	[boleano] Padrão: Falso	
Generate KML for Google Earth	`KML`	[boleano] Padrão: Falso	
Avoid automatic generation of KML files for EPSG:4326	`NO_KML`	[boleano] Padrão: Falso	

Saídas

Etiqueta	Nome	Tipo	Descrição
Diretório de saída	`OUTPUT`	[pasta]	The output folder (for the tiles)

Código Python

Algorithm ID: `gdal:gdal2tiles`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Mesclar

Mescla arquivos raster de maneira simples. Aqui você pode usar uma tabela pseudocor de uma entrada raster e definir o tipo de rasterização de saída. Todas as imagens devem estar no mesmo sistema de coordenadas.

This algorithm is derived from the GDAL merge utility.

Default menu: *Raster ► Miscellaneous*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camadas de entrada*	`INPUT`	[raster] [lista]	Camadas raster de entrada
Grab pseudocolor table from first layer	`PCT`	[boleano] Padrão: Falso	The pseudocolor table from the first layer will be used for the coloring
Place each input file into a separate band	`SEPARATE`	[boleano] Padrão: Falso	Colocar cada arquivo de entrada em uma banda separada

continua na próxima página

Tabela 24.933 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o formato do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Mesclado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster layer. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Input pixel value to treat as "No-Data" Opcional	`NODATA_INPUT`	[numeric: double] Padrão: Não definido	Ignores pixels from files being merged in with this pixel value
Assign specified "NoData" value to output Opcional	`NODATA_OUTPUT`	[numeric: double] Padrão: Não definido	Assigns the specified NoData value to output bands.
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (`|`).

continua na próxima página

Tabela 24.934 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Parâmetros adicionais da linha de comando	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Mesclado	`OUTPUT`	[raster]	Camada raster de saída

Código Python

Algorithm ID: `gdal:merge`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pansharpening

Performs a pan-sharpening operation. It can create a "classic" output dataset (such as GeoTIFF), or a VRT dataset describing the pan-sharpening operation.

See GDAL Pansharpen.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Spectral dataset	`SPECTRAL`	[raster]	Input (spectral) raster layer
Panchromatic dataset	`PANCHROMATIC`	[raster]	Input (panchromatic) raster layer
Saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output (sharpened) raster layer. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Resampling algorithm	`RESAMPLING`	[enumeração] Padrão: 2	The resampling algorithm to use. Options: • 0 — Vizinho mais próximo (`mais próximo`) • 1 — Bilinear (2x2 kernel) (`bilinear`) • 2 — Cubic (4x4 kernel) (`cubic`) • 3 — Cubic B-Spline (4x4 kernel) (`cubicspline`) • 4 — Lanczos (6x6 kernel) (`lanczos`) • 0 — Média (`média``)
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Saída	`OUTPUT`	[raster]	Output (sharpened) raster layer

Código Python

Algorithm ID: `gdal:pansharp`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Calculadora Raster

Command line raster calculator with numpy syntax. Use any basic arithmetic supported by numpy arrays, such as +, -, *, and / along with logical operators, such as >. Note that all input rasters must have the same dimensions, but no projection checking is performed.

See the GDAL Raster Calculator utility docs.

Ver também

Calculadora Raster, *Raster calculator (virtual)*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada A	`INPUT_A`	[raster]	Primeira camada raster de entrada (obrigatório)
Número de banda raster para A	`BAND_A`	[banda raster]	Banda para camada de entrada A (obrigatório)
Camada de entrada B Opcional	`INPUT_B`	[raster] Padrão: Não definido	Segunda camada raster de entrada
Número de banda raster para B Opcional	`BAND_B`	[banda raster]	Banda para camada de entrada B
Camada de entrada C Opcional	`INPUT_C`	[raster] Padrão: Não definido	Terceira camada raster de entrada
Número de banda raster para C Opcional	`BAND_C`	[banda raster]	Banda para a camada de entrada C
Camada de entrada D Opcional	`INPUT_D`	[raster] Padrão: Não definido	Quarta camada raster de entrada
Número de banda raster para D Opcional	`BAND_D`	[banda raster]	Banda para camada de entrada D
Camada de entrada E Opcional	`INPUT_E`	[raster] Padrão: Não definido	Quinta camada raster de entrada
Número de banda raster para E Opcional	`BAND_E`	[banda raster]	Banda para camada de entrada E
Camada de entrada F Opcional	`INPUT_F`	[raster]	Sexta camada raster de entrada
Número de banda raster para F Opcional	`BAND_F`	[banda raster] Padrão: Não definido	Banda para camada de entrada F

continua na próxima página

Tabela 24.939 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Calculation in gdalnumeric syntax using +-/* or any numpy array functions (i.e. logical_and())	`FORMULA`	[string] Padrão: ‘’	A fórmula de cálculo. Exemplos: • `A*(A>0)` — outputs the value of the raster A if the value of A is greater than 0. If not, outputs 0. • `A*(A>0 and A>B)` — outputs the value of A if that value is bigger than 0 and bigger than the value of B. If not, outputs 0. • `A*logical_or(A<=177, A>=185)` — outputs the value of A if A <= 177 or A >= 185. If not, outputs 0. • `sqrt(A*A+B*B)` — Outputs the square root of the sum of the value of A squared and the value of B squared.
Set output NoData value Opcional	`NO_DATA`	[numeric: double] Padrão: Não definido	Value to use for NoData
Handling of extent differences	`EXTENT_OPT`	[enumeração] Padrão: 0	Determines how to handle rasters with different extents. Supported options are: • 0 — Ignore • 1 — Fail • 2 — Union • 3 — Intersect **Atenção** Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* ► *About* menu).
Extensão da produção Opcional	`INPUT`	[extensão]	Custom extent of the output raster. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax` **Atenção** Using this parameter requires QGIS installed with GDAL >= 3.3 (see *Help* ► *About* menu).

continua na próxima página

Tabela 24.939 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tipo de saída raster	`RTYPE`	[enumeração] Padrão: 5	Define o tipo de dados do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Calculado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specify the output (calculated) raster layer. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: ''	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Calculado	`OUTPUT`	[raster]	Saída (calculada) camada rasterizada

Código Python

Algorithm ID: `gdal:rastercalculator`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Informação de raster

Este utilitário cria uma lista com várias informações sobre o conjunto de dados matricial GDAL suportado.

This algorithm is derived from the GDAL info utility.

Default menu: *Raster ► Miscellaneous*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de entrada
Force computation of the actual min/max values for each band	`MIN_MAX`	[boleano] Padrão: Falso	Forces computation of the actual min/max values for each band in the dataset
Read and display image statistics (force computation if necessary)	`STATS`	[boleano] Padrão: Falso	Lê e exibe estatísticas da imagem. Força o cálculo se nenhuma estatística for armazenada em uma imagem.
Suppress GCP info	`NO_GCP`	[boleano] Padrão: Falso	Suprime a impressão da lista de pontos de controle no solo. Pode ser útil para conjuntos de dados com grande quantidade de GCPs, como L1B AVHRR ou HDF4 MODIS, que contêm milhares deles.
Suprimir informação dos metadados	`NO_METADATA`	[boleano] Padrão: Falso	Suprime a impressão de metadados. Alguns conjuntos de dados podem conter muitas cadeias de metadados.

continua na próxima página

Tabela 24.942 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Informação da camada	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the HTML file for output. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Informação da camada	`OUTPUT`	[html]	The HTML file containing information about the input raster layer

Código Python

Algorithm ID: `gdal:gdalinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Retile

Retiles a set of input tiles. All the input tiles must be georeferenced in the same coordinate system and have a matching number of bands. Optionally pyramid levels are generated.

This algorithm is derived from the GDAL Retile utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Input files	`INPUT`	[raster] [lista]	The input raster files
Largura do bloco	`TILE_SIZE_X`	[numeric: integer] Padrão: 256	Largura dos blocos em pixels (mínimo 0)
Altura de bloco	`TILE_SIZE_Y`	[numeric: integer] Padrão: 256	Height of the tiles in pixels (minimum 0)
Overlap in pixels between consecutive tiles	`OVERLAP`	[numeric: integer] Padrão: 0	
Número de níveis de pirâmide para construir	`LEVELS`	[numeric: integer] Padrão: 1	Mínimo: 0
Diretório de saída	`OUTPUT`	[pasta] Padrão: `[Salvar em pasta temporária]`	Specify the output folder for the tiles. *One of*: • Salvar como Diretório Temporário • Salvar no Diretório
CSV file containing the tile(s) georeferencing information	`OUTPUT_CSV`	[arquivo] Default: `[Pular saída]`	Specify the output file for the tiles. *One of*: • Ignorar Saída • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Source coordinate reference system Opcional	`SOURCE_CRS`	[src] Padrão: Não definido	
Método de reamostragem	`RESAMPLING`	[enumeração] Padrão: 0	The resampling algorithm to use. Options: • 0 — Vizinho mais próximo (`mais próximo`) • 1 — Bilinear (2x2 kernel) (`bilinear`) • 2 — Cubic (4x4 kernel) (`cubic`) • 3 — Cubic B-Spline (4x4 kernel) (`cubicspline`) • 4 — Lanczos (6x6 kernel) (`lanczos`)

continua na próxima página

Tabela 24.946 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Delimitador de coluna usado no arquivo CSV Opcional	`DELIMITER`	[string] Padrão: ';'	Delimiter to use in the CSV file containing the tile(s) georeferencing information
Opções de criação adicionais Opcional	`CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: ''	Adicionar opções extras de linha de comando GDAL
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o formato do arquivo raster de saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Construit apenas as pirâmides	`ONLY_PYRAMIDS`	[boleano] Padrão: Falso	
Use separate directory for each tile row	`DIR_FOR_ROW`	[boleano] Padrão: Falso	

Saídas

Etiqueta	Nome	Tipo	Descrição
Diretório de saída	`OUTPUT`	[pasta]	The output folder for the tiles.
CSV file containing the tile(s) georeferencing information	`OUTPUT_CSV`	[arquivo]	O arquivo CSV com informações de georreferenciamento para os mosaícos.

Código Python

Algorithm ID: `gdal:retile`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Índice deslizante

Cria uma camada vetorial com um registro para cada arquivo de varredura de entrada, um atributo contendo o nome do arquivo e uma geometria de polígono definindo a varredura. Essa saída é adequada para uso com o MapServer como um indexador de rasterização.

This algorithm is derived from the GDAL Tile Index utility.

Default menu: *Raster ► Miscellaneous*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Input files	`LAYERS`	[raster] [lista]	Arquivos de rasterização de entrada. Podem ser vários arquivos.
Field name to hold the file path to the indexed rasters	`PATH_FIELD_NAM` Optional	[string] Padrão: 'localização'	Nome do arquivo para manter o caminho do arquivo para os rasteres indexados
Store absolute path to the indexed rasters	`ABSOLUTE_PATH`	[boleano] Padrão: Falso	Defina se o caminho absoluto para os arquivos raster é armazenado no arquivo de índice tile. Por padrão, os nomes dos arquivos raster serão colocados no arquivo exatamente como são especificados no comando.
Skip files with different projection reference	`PROJ_DIFFERENC`	[boleano] Padrão: Falso	Somente arquivos com a mesma projeção dos arquivos já inseridos no índice de blocos serão inseridos. O padrão não verifica a projeção e aceita todas as entradas.

continua na próxima página

Tabela 24.948 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Tile index	`OUTPUT`	[vetor: polígono] Padrão: `[Salvar em arquivo temporário]`	Specify the polygon vector layer to write the index to. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Transformar geometrias para o SRC fornecido Opcional	`TARGET_CRS`	[src]	As geometrias dos arquivos de entrada serão transformadas para o sistema de referência de coordenadas alvo especificado. O padrão cria polígonos retangulares simples no mesmo sistema de referência de coordenadas que os rasters de entrada.
The name of the field to store the SRS of each tile Opcional	`CRS_FIELD_NAME`	[string]	The name of the field to store the SRS of each tile
The format in which the CRS of each tile must be written	`CRS_FORMAT`	[enumeração] Padrão: 0	Formato para o SRC. Um de: • 0 – Automático (`AUTO`) • 1 – Well-known text (`WKT`) • 2 – EPSG (`EPSG`) • 3 – Proj.4 (`PROJ`)

Saídas

Etiqueta	Nome	Tipo	Descrição
Tile index	`OUTPUT`	[vetor: polígono]	The polygon vector layer with the tile index.

Código Python

Algorithm ID: `gdal:tileindex`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Viewshed

Calculates a viewshed raster from an input raster DEM using method defined in Wang2000 for a user defined point.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada raster de elevação de entrada
Número da banda	`BAND`	[banda raster] Padrão: 1	O número da banda a ser usada como elevação
Local do observador	`OBSERVER`	[coordinates]	A localização do observador
Altura do observador	`OBSERVER_HEIGHT`	[numeric: double] Padrão: 1,0	A altitude do observador, nas unidades MDE
Target height	`TARGET_HEIGHT`	[numeric: double] Padrão: 1,0	A altitude do elemento alvo, nas unidades MDE
Distância máxima do observador para calcular a visibilidade	`MAX_DISTANCE`	[numeric: double] Padrão: 100.0	Maximum distance from observer to compute visibility, in the DEM units
Saída	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Output raster layer. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Parâmetros adicionais da linha de comando	`EXTRA`	[string] Padrão: Não definido	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Saída	`OUTPUT`	[raster]	The raster layer displaying the viewshed.

Código Python

Algorithm ID: `gdal:viewshed`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.2.5 Projeções raster

Atribuir projeção

Aplica um sistema de coordenadas a um conjunto de dados raster.

This algorithm is derived from the GDAL edit utility.

Menu padrão: *Raster ► Projeções*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT_LAYER`	[raster]	Camada raster de entrada
SRC desejada	`CRS`	[src]	A projeção (SRC) da camada de saída

Saídas

Rótulo	Nome	Tipo	Descrição
Camada com projeção	`OUTPUT`	[raster]	A camada raster de saída (com as novas informações de projeção)

Código Python

Algorithm ID: `gdal:assignprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Extrair projeção

Extrai a projeção de um arquivo raster e o grava em um arquivo *global* com extensão `.wld`.

This algorithm is derived from the GDAL srsinfo utility.

Menu padrão: *Raster ► Projeções*

Parâmetros

Rótulo	Nome	Tipo	Descrição
Arquivo de entrada	`INPUT_LAYER`	[raster]	Input raster The raster layer has to be file based, as the algorithm uses the path to the raster file as the location of the generated `.wld` file. Using a non-file raster layer will lead to an error.
Criar também arquivo .prj	`PRJ_FILE_CREAT`	[boleano] Padrão: Falso	Se isto for ativado, um arquivo `.prj` contendo as informações de projeção também é criado.

Saídas

Rótulo	Nome	Tipo	Descrição
Arquivo global	`WORLD_FILE`	[arquivo]	Arquivo texto com extensão `.wld` contendo parâmetros de transformação para o arquivo raster.
ESRI Shapefile prj file	`PRJ_FILE`	[arquivo]	Arquivo texto com `.prj` extensão que descreve o SRC. Será `Nenhum` se *Criar também o arquivo .prj* for Falso.

Código Python

Algorithm ID: `gdal:extractprojection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Warp (reproject)

Reproje uma camada rasterizada em outro Sistema de Referência Coordenada (SRC). A resolução do arquivo de saída e o método de reamostragem podem ser escolhidos.

This algorithm is derived from the GDAL warp utility.

Menu padrão: *Raster ► Projeções*

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[raster]	Camada de entrada raster para reprojetar
SRC de Origem Opcional	`SOURCE_CRS`	[src]	Define o SRC da camada raster de entrada
SRC Alvo Opcional	`TARGET_CRS`	[src] Padrão: `EPSG:4326`	O SRC da camada de saída
Método de amostragem a ser utilizado	`RESAMPLING`	[enumeração] Padrão: 0	Método de reamostragem de valor de pixel a ser usado. Opções: • 0 — Nearest Neighbour • 1 — Bilinear (2x2 kernel) • 2 — Cubic (4x4 kernel) • 3 — Cubic B-Spline (4x4 kernel) • 4 — Lanczos (6x6 kernel) • 5 — Média • 6 — Modo • 7 — Máximo • 8 — Mínimo • 9 — Mediana • 10 — First quartile (Q1) • 11 — Third quartile (Q3)
NoData value for output bands Opcional	`NODATA`	[numeric: double] Padrão: Não definido	Sets NoData value for output bands. If not provided, then NoData values will be copied from the source dataset.

continua na próxima página

Tabela 24.958 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Resolução do arquivo de saída em unidades georreferenciadas de destino Opcional	`TARGET_RESOLUTION`	[numeric: double] Padrão: Não definido	Define a resolução do arquivo de saída do resultado da reprojeção
Reprojetado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster layer. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 0	Define o formato do arquivo raster de saída. Opções: • 0 — Usar o tipo de dados da camada de entrada • 1 — Byte (Eight bit unsigned integer (quint8)) • 2 — Int16 (Sixteen bit signed integer (qint16)) • 3 — UInt16 (Sixteen bit unsigned integer (quint16)) • 4 — UInt32 (Thirty two bit unsigned integer (quint32)) • 5 — Int32 (Thirty two bit signed integer (qint32)) • 6 — Float32 (Thirty two bit floating point (float)) • 7 — Float64 (Sixty four bit floating point (double)) • 8 — CInt16 (Complex Int16) • 9 — CInt32 (Complex Int32) • 10 — CFloat32 (Complex Float32) • 11 — CFloat64 (Complex Float64) • 12 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)

continua na próxima página

Tabela 24.959 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Extensões georreferenciadas do arquivo de saída a ser criado Opcional	`TARGET_EXTENT`	[extensão]	Sets the georeferenced extent of the output file to be created (in the *Target CRS* by default. In the *CRS of the target raster extent*, if specified). Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Inserir as coordenadas como `xmin, xmax, ymin, ymax`
CRS of the target raster extent Opcional	`TARGET_EXTENT_CRS`	[src]	Specifies the CRS in which to interpret the coordinates given for the extent of the output file. This must not be confused with the target CRS of the output dataset. It is instead a convenience e.g. when knowing the output coordinates in a geodetic long/lat CRS, but wanting a result in a projected coordinate system.
Use multithreaded warping implementation	`MULTITHREADING`	[boleano] Padrão: Falso	Duas linhas serão usadas para processar pedaços da imagem e realizar operações de entrada/saída simultaneamente. Note que o cálculo em si não é multi-tarefa.
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: Não definido	Adicione opções extras de linha de comando GDAL.

Saídas

Rótulo	Nome	Tipo	Descrição
Reprojetado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Reprojetar camada rasterizada de saída

Código Python

Algorithm ID: `gdal:warpreproject`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.2.6 Conversão vetorial

Converter formato

Converte qualquer camada vetorial compatível com OGR em outro formato compatível com OGR.

This algorithm is derived from the ogr2ogr utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	Camada vetorial de entrada
Convert all layers from dataset	`CONVERT_ALL_LAYER`	[boleano] Padrão: Falso	Converts the whole dataset. Supported output formats for this option are `GPKG` and `GML`.
Convertido	`OUTPUT`	[o mesmo que entrada]	Specification of the output vector layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo… Para "Salvar em arquivo", o formato de saída tem que ser especificado. Todos os formatos de raster GDAL são suportados. Para "Salvar em um arquivo temporário" será utilizado o formato de vetor padrão do QGIS.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Etiqueta	Nome	Tipo	Descrição
Convertido	`OUTPUT`	[o mesmo que entrada]	A camada vetorial de saída.

Código Python

ID do algoritmo: `gdal:converterformato`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Rasterizar (sobrescrever com atributo)

Substitui uma camada raster com valores de uma camada vetorial. Novos valores são atribuídos baseados no valor do atributo da feição vetorial sobreposta.

This algorithm is derived from the GDAL rasterize utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Camada matricial de entrada	`INPUT_RASTER`	[raster]	Camada raster de entrada
Campo a ser utilizado para uma queima em valor Opcional	`FIELD`	[campo de tabela: numérico]	Define o campo de atributo a ser usado para definir os valores de pixels

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Adicionar a queima em valores aos valores raster existentes	`ADD`	[boleano] Padrão: Falso	Se Falso, os pixels são atribuídos ao valor do campo selecionado. Se Verdadeiro, o valor do campo selecionado é adicionado ao valor da camada matricial de entrada.

continua na próxima página

Tabela 24.965 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Parâmetros adicionais da linha de comando Opcional	EXTRA	[string] Padrão: ”	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Rasterizado	OUTPUT	[raster]	A camada raster de entrada sobrescrita

Código Python

Algoritmo ID: `gdal:rasterize_over`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Rasterize (substituir com valor fixo)

Substitui partes de uma camada raster com um valor fixo. Os pixels a serem sobrescritos são escolhidos com base na camada vetorial (sobreposta) fornecida.

This algorithm is derived from the GDAL rasterize utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	INPUT	[vector: geometry]	Camada vetorial de entrada
Camada matricial de entrada	INPUT_RASTER	[raster]	Camada raster de entrada
Um valor fixo para queimar	BURN	[numeric: double] Padrão: 0.0	O valor para queimar

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Adicionar a queima em valores aos valores raster existentes	ADD	[boleano] Padrão: Falso	Se Falso, é atribuído o valor fixo aos pixels. Se Verdadeiro, o valor fixo é adicionado ao valor da camada raster de entrada.
Parâmetros adicionais da linha de comando Opcional	EXTRA	[string] Padrão: ''	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Rasterizado	OUTPUT	[raster]	A camada raster de entrada sobrescrita

Código Python

Algoritmo ID: `gdal:rasterize_over_fixed_value`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Rasterizar (vetor a rasterizar)

Converte geometrias vetoriais (pontos, linhas e polígonos) em uma imagem matricial.

This algorithm is derived from the GDAL rasterize utility.

Menu padrão: *Raster ► Conversão*

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	INPUT	[vector: geometry]	Camada vetorial de entrada
Campo a ser utilizado para uma queima em valor Opcional	FIELD	[campo de tabela: numérico]	Define o campo de atributo do qual os atributos para os pixels devem ser escolhidos

continua na próxima página

Tabela 24.970 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Um valor fixo para queimar Opcional	`BURN`	[numeric: double] Padrão: 0.0	Um valor fixo para gravar em uma banda para todas as feições.
Burn value extracted from the "Z" values of the feature	`USE_Z`	[boleano] Padrão: Falso	Indicates that a burn value should be extracted from the "Z" values of the feature. Works with points and lines (linear interpolation along each segment). For polygons, works properly only if they are flat (same Z value for all vertices)
Unidades de tamanho rasterizado de saída	`UNITS`	[enumeração] Padrão: 0	Unidades a usar ao definir o tamanho/resolução raster de saída. Um de: • 0 — Pixels • 1 — Unidades Georreferenciadas
Largura/Resolução horizontal	`WIDTH`	[numeric: double] Padrão: 0.0	Define a largura (se as unidades de tamanho forem "Pixels") ou a resolução horizontal (se as unidades de tamanho forem "Unidades Georreferenciadas") do raster de saída. Valor mínimo: 0,0.
Altura/resolução vertical	`HEIGHT`	[numeric: double] Padrão: 0.0	Define a altura (se unidades de tamanho for "Pixels") ou resolução vertical (se unidades de tamanho for "Unidades Georreferenciadas") do raster de saída.
Extensão da produção Opcional	`EXTENT`	[extensão]	Extensão da camada raster de saída. Se a extensão não for especificada, será utilizada a extensão mínima que cobre a(s) camada(s) de referência selecionada(s). Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Assign a specified NoData value to output bands Opcional	`NODATA`	[numeric: double] Padrão: 0.0	Assigns a specified NoData value to output bands

continua na próxima página

Tabela 24.970 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Rasterizado	`OUTPUT`	[raster] Padrão: `[Salvar em arquivo temporário]`	Specification of the output raster layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo… Para "Salvar para arquivar", o formato de saída tem que ser especificado. Todos os formatos de raster GDAL são suportados. Para "Salvar em um arquivo temporário" será utilizado o formato raster padrão QGIS.

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: ''	Para adicionar uma ou mais opções de criação que controlam o raster a ser criado (cores, tamanho do bloco, compactação de arquivos …). Por conveniência, você pode contar com perfis predefinidos (veja *seção de opções do driver GDAL*). Batch Process and Model Designer: separate multiple options with a pipe character (|).
Tipo de dados de saída	`DATA_TYPE`	[enumeração] Padrão: 5	Define o formato do arquivo de raster saída. Opções: • 0 — Byte (Eight bit unsigned integer (quint8)) • 1 — Int16 (Sixteen bit signed integer (qint16)) • 2 — UInt16 (Sixteen bit unsigned integer (quint16)) • 3 — UInt32 (Thirty two bit unsigned integer (quint32)) • 4 — Int32 (Thirty two bit signed integer (qint32)) • 5 — Float32 (Thirty two bit floating point (float)) • 6 — Float64 (Sixty four bit floating point (double)) • 7 — CInt16 (Complex Int16) • 8 — CInt32 (Complex Int32) • 9 — CFloat32 (Complex Float32) • 10 — CFloat64 (Complex Float64) • 11 — Int8 (Eight bit signed integer (qint8)) Available options depend on the GDAL version built with QGIS (see *Help* ► *About* menu)
Pré-inicializar a imagem de saída com valor Opcional	`INIT`	[numeric: double]	Pre-initializes the output image bands with this value. Not marked as the NoData value in the output file. The same value is used in all the bands.

continua na próxima página

Tabela 24.971 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Inverter rasterização	`INVERT`	[boleano] Padrão: Falso	Burns the fixed burn value, or the burn value associated with the first feature into all parts of the image not inside the provided polygon.
Parâmetros adicionais da linha de comando Opcional	`EXTRA`	[string] Padrão: ''	Adicionar opções extras de linha de comando GDAL

Saídas

Etiqueta	Nome	Tipo	Descrição
Rasterizado	`OUTPUT`	[raster]	Camada raster de saída

Código Python

Algorithm ID: `gdal:rasterize`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.2.7 Geoprocessamento vetorial

Buffer vectors

Criar buffers em torno das feições de uma camada vetorial.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	A camada vetorial de entrada
Nome da coluna geometria	`GEOMETRY`	[string] Padrão: 'geometria'	O nome da coluna de geometria da camada de entrada a ser usada
Distância do buffer	`DISTANCE`	[numeric: double] Padrão: 10.0	Mínimo: 0,0

continua na próxima página

Tabela 24.973 – continuação da página anterior

Etiqueta	Nome	Tipo	Descrição
Dissolver por atributo Opcional	`FIELD`	[campo da tabela: qualquer] Padrão: Não definido	Campo a ser utilizado para dissolver
Dissolver resultados	`DISSOLVE`	[boleano] Padrão: Falso	If set, the result is dissolved. If no field is set for dissolving, all the buffers are dissolved into one feature.
Produce one feature for each geometry in any kind of geometry collection in the source file	`EXPLODE_COLLECTI`	[boleano] Padrão: Falso	
Buffer	`OUTPUT`	[vetor: polígono] Padrão: `[Salvar em arquivo temporário]`	Specify the output buffer layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Etiqueta	Nome	Tipo	Descrição
Buffer	`OUTPUT`	[vetor: polígono]	The output buffer layer

Código Python

Algorithm ID: `gdal:buffervectors`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Recortar vetor pela extensão

Clips any OGR-supported vector file to a given extent.

This algorithm is derived from the GDAL ogr2ogr utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	A camada vetorial de entrada
Clip extent	`EXTENT`	[extensão]	Defines the bounding box that should be used for the output vector file. It has to be defined in target CRS coordinates. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Insira as coordenadas como `xmin, xmax, ymin, ymax`
Clipped (extent)	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specify the output (clipped) layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Etiqueta	Nome	Tipo	Descrição
Clipped (extent)	`OUTPUT`	[o mesmo que entrada]	The output (clipped) layer. The default format is "ESRI Shapefile".

Código Python

Algorithm ID: `gdal:clipvectorbyextent`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Recortar vetor por camada de máscara

Clips any OGR-supported vector layer by a mask polygon layer.

This algorithm is derived from the GDAL ogr2ogr utility.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	A camada vetorial de entrada
Camada de máscara	`MASK`	[vetor: polígono]	Layer to be used as clipping extent for the input vector layer.
Clipped (mask)	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	The output (masked) layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Etiqueta	Nome	Tipo	Descrição
Clipped (mask)	`OUTPUT`	[o mesmo que entrada]	The output (masked) layer. The default format is "ESRI Shapefile".

Código Python

Algorithm ID: `gdal:clipvectorbypolygon`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Dissolver

Dissolver (combinar) geometrias que tenham o mesmo valor para um determinado atributo/campo. As geometrias de saída são multipartes.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	A camada de entrada para dissolver
Dissolver campo Opcional	`FIELD`	[campo da tabela: qualquer]	O campo da camada de entrada a ser usado para dissolver
Nome da coluna geometria	`GEOMETRY`	[string] Padrão: 'geometria'	O nome da coluna de geometria da camada de entrada a ser usada para dissolução.
Dissolvido	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specify the output layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Produce one feature for each geometry in any kind of geometry collection in the source file	`EXPLODE_COLLECTI`	[boleano] Padrão: Falso	Produzir uma feição para cada geometria em qualquer tipo de coleção de geometria no arquivo de origem
Manter atributos de entrada	`KEEP_ATTRIBUTE`	[boleano] Padrão: Falso	Manter todos os atributos da camada de entrada
Contar feições dissolvidas	`COUNT_FEATURES`	[boleano] Padrão: Falso	Contar as feições dissolvidas e inclua-as na camada de saída.
Calcular área e perímetro de feições dissolvidas	`COMPUTE_AREA`	[boleano] Padrão: Falso	Compute the area and perimeter of dissolved features and include them in the output layer
Calcular mín/máx/soma/méd para atributo	`COMPUTE_STATISTIC`	[boleano] Padrão: Falso	Calcular estatísticas (min, max, soma e média) para o atributo numérico especificado e incluí-los na camada de saída
Atributo numérico para calcular estatísticas Opcional	`STATISTICS_ATTRIBUTE`	[campo de tabela: numérico]	O atributo numérico para calcular estatísticas
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Etiqueta	Nome	Tipo	Descrição
Dissolvido	`OUTPUT`	[o mesmo que entrada]	The output multipart geometry layer (with dissolved geometries)

Código Python

Algorithm ID: `gdal:dissolve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Offset curve

Desloca as linhas por uma distância especificada. Distâncias positivas deslocarão as linhas à esquerda e distâncias negativas deslocarão as linhas à direita.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	A camada de linha de entrada
Nome da coluna geometria	`GEOMETRY`	[string] Padrão: 'geometria'	O nome da coluna de geometria da camada de entrada a ser usada
Distância de deslocamento (lado esquerdo: positivo, lado direito: negativo)	`DISTANCE`	[numeric: double] Padrão: 10.0	
Offset curve	`OUTPUT`	[vetor: linha] Padrão: `[Salvar em arquivo temporário]`	Specify the output line layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Etiqueta	Nome	Tipo	Descrição
Offset curve	`OUTPUT`	[vetor: linha]	The output offset curve layer

Código Python

Algorithm ID: `gdal:offsetcurve`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

One side buffer

Cria um buffer em um lado (direito ou esquerdo) das linhas em uma camada de vetor de linha.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	A camada de linha de entrada
Nome da coluna geometria	`GEOMETRY`	[string] Padrão: 'geometria'	O nome da coluna de geometria da camada de entrada a ser usada
Distância do buffer	`DISTANCE`	[numeric: double] Padrão: 10.0	
Lado do buffer	`BUFFER_SIDE`	[enumeração] Padrão: 0	Um de: • 0 — Direita • 1 — Esquerda
Dissolver por atributo Opcional	`FIELD`	[campo da tabela: qualquer] Padrão: Não definido	Campo a ser utilizado para dissolver
Dissolver todos os resultados	`DISSOLVE`	[boleano] Padrão: Falso	**Se definido, o resultado é dissolvido.** Se nenhum campo for definido para dissolver, todos os buffers serão dissolvidos em uma feição.
Produce one feature for each geometry in any kind of geometry collection in the source file	`EXPLODE_COLLECTI`	[boleano] Padrão: Falso	
One-sided buffer	`OUTPUT`	[vetor: polígono] Padrão: `[Salvar em arquivo temporário]`	Specify the output buffer layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Etiqueta	Nome	Tipo	Descrição
One-sided buffer	`OUTPUT`	[vetor: polígono]	The output buffer layer

Código Python

Algorithm ID: `gdal:onesidebuffer`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Pontos ao longo das linhas

Generates a point on each line of a line vector layer at a distance from start. The distance is provided as a fraction of the line length.

Parâmetros

Parâmetros básicos

Etiqueta	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: linha]	A camada de linha de entrada
Nome da coluna geometria	`GEOMETRY`	[string] Padrão: 'geometria'	O nome da coluna de geometria da camada de entrada a ser usada
Distância do início da linha representada como uma fração do comprimento da linha	`DISTANCE`	[numeric: double] Padrão: 0,5	
Pontos ao longo das linhas	`OUTPUT`	[vetor: ponto] Padrão: `[Salvar em arquivo temporário]`	Specify the output point layer. *One of*: • Salvar como Arquivo Temporário • Salvar em Arquivo…

Parâmetros avançados

Etiqueta	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CRE-ATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Etiqueta	Nome	Tipo	Descrição
Pontos ao longo da linha	`OUTPUT`	[vetor: ponto]	A camada de ponto de saída

Código Python

Algorithm ID: `gdal:pointsalonglines`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

24.2.8 Diversos vetoriais

Construir vetor virtual

Creates a virtual vector layer that contains a set of vector layers.

Este algoritmo é especialmente útil no caso de outro algoritmo precisar de várias camadas, mas aceitar apenas um `vrt` no qual as camadas são especificadas.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Fontes de dados de entrada	`INPUT`	[vetor: qualquer] [lista]	Selecione as camadas vetoriais que você deseja usar para construir o vetor virtual
Create "unioned" VRT	`UNIONED`	[boleano] Padrão: Falso	Verifique se você deseja unir todos os vetores em um único arquivo `vrt`

continua na próxima página

Tabela 24.994 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Vetor virtual	`OUTPUT`	[o mesmo que entrada] Padrão: `[Salvar em arquivo temporário]`	Specify the output layer containing only the duplicates. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo…

Saídas

Rótulo	Nome	Tipo	Descrição
Vetor virtual	`OUTPUT`	[vetor: qualquer]	O vetor virtual de saída feito a partir das fontes escolhidas

Código Python

Algorithm ID: `gdal:buildvirtualvector`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Executar SQL

Executa uma consulta simples ou complexa com sintaxe SQL na camada de origem. O resultado da consulta será adicionado como uma nova camada.

This algorithm is derived from the GDAL ogr2ogr utility.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	OGR-supported input vector layer
Expressão SQL	`SQL`	[string]	Defines the SQL query, for example `SELECT * FROM my_table WHERE name is not null`.
Dialeto SQL	`DIALECT`	[enumeração] Padrão: 0	dialeto SQL a ser usado. Um de: • 0 — Nenhum • 1 — OGR SQL • 2 — SQLite

continua na próxima página

Tabela 24.996 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Resultado SQL	`OUTPUT`	[vetor: qualquer]	Specification of the output layer. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo… For `Save to File`, the output format has to be specified. All GDAL vector formats are supported. For `Save to a Temporary File` the default output vector layer format will be used.

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Rótulo	Nome	Tipo	Descrição
Resultado SQL	`OUTPUT`	[vetor: qualquer]	Camada vetorial criada pela consulta

Código Python

Algorithm ID: `gdal:executesql`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exportação para PostgreSQL (conexões disponíveis)

Imports vector layers inside a PostgreSQL database on the basis of an available connection. The connection has to *be defined properly* beforehand. Be aware that the checkboxes 'Save Username' and 'Save Password' are activated. Then you can use the algorithm.

This algorithm is derived from the GDAL ogr2ogr utility.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Banco de dados (nome da conexão)	`DATABASE`	[string]	O banco de dados PostgreSQL para conexão com
Camada de entrada	`INPUT`	[vetor: qualquer]	OGR-supported vector layer to export to the database
Codificação de forma Opcional	`SHAPE_ENCODING`	[string] Padrão: ‘’	Define a codificação a ser aplicada aos dados
Tipo de geometria de saída	`GTYPE`	[enumeração] Padrão: 0	Define o tipo de geometria de saída. Uma das: • 0 — • 1 — NENHUM • 2 — GEOMETRIA • 3 — PONTO • 4 — FIO DE LINHA • 5 — POLÍGONO • 6 –COLEÇÃO DE GEOMETRIA • 7 — MULTIPONTO • 8 — MULTIPOLÍGONO • 9 — CADEIA DE LINHAS MÚLTIPLAS • 10 — CIRCULARSTRING • 11 — COUMPOUNDCURVE • 12 — CURVEPOLYGON • 13 — MULTICURVE • 14 — MULTISURFACE • 15 — CONVERT_TO_LINEAR • 16 — CONVERT_TO_CURVE
Atribua um SRC de saída Opcional	`A_SRS`	[src] Padrão: Não definido	Define o SRC de saída da tabela de banco de dados
Reprojetar para este SRC na saída Opcional	`T_SRS`	[src] Padrão: Não definido	Reprojeta/transforma para este SRC na saída
Substitui a fonte do SRC Opcional	`S_SRS`	[src] Padrão: Não definido	Substitui o SRC da camada de entrada
Esquema (nome do esquema) Opcional	`SCHEMA`	[string] Padrão: ‘público’	Define o esquema para a tabela do banco de dados
Tabela para exportar (deixe em branco para usar o nome da camada) Opcional	`TABLE`	[string] Padrão: ‘’	Define um nome para a tabela que será importada para o banco de dados. Por padrão, o nome da tabela é o nome do arquivo vetorial de entrada.
Chave Primária (novo campo) Opcional	`PK`	[string] Padrão: ‘id’	Define qual campo de atributo será a chave primária da tabela do banco de dados

continua na próxima página

Tabela 24.999 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Chave primária (campo existente, usado se a opção acima for deixada vazia) Opcional	`PRIMARY_KEY`	[campo da tabela: qualquer] Padrão: Não definido	Define qual campo de atributo na camada exportada será a chave primária da tabela do banco de dados
Nome da coluna geometria Opcional	`GEOCOLUMN`	[string] Padrão: 'geom'	Define em qual campo de atributo do banco de dados estarão as informações de geometria
Dimensões vetoriais Opcional	`DIM`	[enumeração] Padrão: 0 (2D)	Defines the coordinate dimensions of the imported vector data. One of: • 0 — 2 (`XY`) • 1 — 3 (`XYZ`) • 2 — 4 (`XYZM`)
Distância tolerada para simplificação Opcional	`SIMPLIFY`	[string] Padrão: ''	Define uma tolerância de distância para a simplificação das geometrias vetoriais a serem importadas. Por padrão, não há simplificação.
Distância máxima entre 2 nós (densificação) Opcional	`SEGMENTIZE`	[string] Padrão: ''	A distância máxima entre dois nós. Utilizada para criar pontos intermediários. Por padrão, não há densificação.
Selecionar feições por extensão (definido na camada de entrada SRC) Opcional	`SPAT`	[extensão] Padrão: Não definido	Você pode selecionar características de uma determinada extensão que estarão na tabela de saída. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Inserir as coordenadas como `xmin, xmax, ymin, ymax`
Recortar a camada de entrada usando a extensão acima (retângulo)*	`CLIP`	[boleano] Padrão: Falso	A camada de entrada será recortada pela extensão que você definiu anteriormente
Select features using a SQL "WHERE" statement (Ex: column="value") Opcional	`WHERE`	[string] Padrão: ''	Defines with a SQL "WHERE" statement which features should be selected from the input layer

continua na próxima página

Tabela 24.999 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Agrupar N feições por transação (Padrão: 2000) Opcional	GT	[string] Padrão: ‘’	Você pode agrupar as feições de entrada em transações em que N define o tamanho. Por padrão, N limita o tamanho da transação a 20.000 recursos.
Sobrescrever a tabela existente	OVERWRITE	[boleano] Padrão: Verdadeiro	Se houver uma tabela com o mesmo nome no banco de dados, e se esta opção estiver configurada para Verdadeiro tabela será sobrescrita.
Adicionar para a tabela existente	APPEND	[boleano] Padrão: Falso	Se marcado / Verdadeiro, os dados vetoriais serão anexados a uma tabela existente. Novos campos encontrados na camada de entrada são ignorados. Por padrão, uma nova tabela será criada.
Alterar e acrescentar novos campos à tabela existente	ADDFIELDS	[boleano] Padrão: Falso	Se ativado, os dados vetoriais serão anexados a uma tabela existente, não será criada uma nova tabela. Novos campos encontrados na camada de entrada são adicionados à tabela. Por padrão, uma nova tabela será criada.
Do not launder columns/table names	LAUNDER	[boleano] Padrão: Falso	Com esta opção marcada, você pode evitar o comportamento padrão (converter nomes de colunas para minúsculas, remover espaços e outros caracteres inválidos).
Não criar Índice Espacial	INDEX	[boleano] Padrão: Falso	Impede que um índice espacial para a tabela de saída seja criado. Por padrão, um índice espacial é adicionado.
Continuar após uma falha, pulando a feição com falha	SKIPFAILURES	[boleano] Padrão: Falso	Continue after a failure, skipping the failed feature.
Validate geometries based on Simple Features specification	MAKEVALID	[boleano] Padrão: Falso	Applies cleanup operations on geometries to ensure they are valid regarding the rules of the Simple Features specification.
Promover para Multipartes	PROMOTETOMULTI	[boleano] Padrão: Verdadeiro	Converte o tipo de geometria de recursos para várias partes na tabela de saída
Mantenha a largura e precisão dos atributos de entrada	PRECISION	[boleano] Padrão: Verdadeiro	Evita modificar os atributos da coluna para cumprir os dados de entrada
Opções de criação adicionais Opcional	CREATION_OPTIONS (for QGIS <= 3.42, this was OPTIONS)	[string] Padrão: ‘’ (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Este algoritmo não tem saída.

Código Python

Algorithm ID: `gdal:importvectorintopostgisdatabaseavailableconnections`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Exportação para PostgreSQL (nova conexão)

Imports vector layers inside a PostgreSQL database. A new connection to the PostgreSQL database must be created.

This algorithm is derived from the GDAL ogr2ogr utility.

Parâmetros

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vetor: qualquer]	OGR-supported vector layer to export to the database
Codificação de forma Opcional	`SHAPE_ENCODING`	[string] Padrão: ''	Define a codificação a ser aplicada aos dados
Tipo de geometria de saída	`GTYPE`	[enumeração] Padrão: 0	Define o tipo de geometria de saída. Uma das: • 0 — • 1 — NENHUM • 2 — GEOMETRIA • 3 — PONTO • 4 — FIO DE LINHA • 5 — POLÍGONO • 6 –COLEÇÃO DE GEOMETRIA • 7 — MULTIPONTO • 8 — MULTIPOLÍGONO • 9 — CADEIA DE LINHAS MÚLTIPLAS • 10 — CIRCULARSTRING • 11 — COUMPOUNDCURVE • 12 — CURVEPOLYGON • 13 — MULTICURVE • 14 — MULTISURFACE • 15 — CONVERT_TO_LINEAR • 16 — CONVERT_TO_CURVE
Atribua um SRC de saída Opcional	`A_SRS`	[src] Padrão: Não definido	Define o SRC de saída da tabela de banco de dados

continua na próxima página

Tabela 24.1000 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Reprojetar para este SRC na saída Opcional	`T_SRS`	[src] Padrão: Não definido	Reprojeta/transforma para este SRC na saída
Substitui a fonte do SRC Opcional	`S_SRS`	[src] Padrão: Não definido	Substitui o SRC da camada de entrada
Host Opcional	`HOST`	[string] Padrão: 'localhost'	Nome do host do banco de dados
Porta Opcional	`PORT`	[string] Padrão: '5432'	Número da porta que o servidor de banco de dados PostgreSQL ouve em
Nomedeusuário Opcional	`USER`	[string] Padrão: ''	Nome de usuário usado para fazer login no banco de dados
Nome do banco de dados Opcional	`DBNAME`	[string] Padrão: ''	Nome do banco de dados
Senha Opcional	`PASSWORD`	[string] Padrão: ''	Senha usada com o nome de usuário para se conectar ao banco de dados
Esquema (nome do esquema) Opcional	`SCHEMA`	[string] Padrão: 'público'	Define o esquema para a tabela do banco de dados
Nome da tabela, deixe em branco para usar o nome de entrada Opcional	`TABLE`	[string] Padrão: ''	Define um nome para a tabela que será importada para o banco de dados. Por padrão, o nome da tabela é o nome do arquivo vetorial de entrada.
Chave Primária (novo campo) Opcional	`PK`	[string] Padrão: 'id'	Define qual campo de atributo será a chave primária da tabela do banco de dados
Chave primária (campo existente, usado se a opção acima for deixada vazia) Opcional	`PRIMARY_KEY`	[campo da tabela: qualquer] Padrão: Não definido	Define qual campo de atributo na camada exportada será a chave primária da tabela do banco de dados
Nome da coluna geometria Opcional	`GEOCOLUMN`	[string] Padrão: 'geom'	Define em qual campo de atributo armazenar as informações de geometria
Dimensões vetoriais Opcional	`DIM`	[enumeração] Padrão: 0 (2D)	Defines the coordinate dimensions of the imported vector data. One of: • 0 — 2 (`XY`) • 1 — 3 (`XYZ`) • 2 — 4 (`XYZM`)
Distância tolerada para simplificação Opcional	`SIMPLIFY`	[string] Padrão: ''	Define uma tolerância de distância para a simplificação das geometrias vetoriais a serem importadas. Por padrão, não há simplificação.
Distância máxima entre 2 nós (densificação) Opcional	`SEGMENTIZE`	[string] Padrão: ''	A distância máxima entre dois nós. Utilizada para criar pontos intermediários. Por padrão, não há densificação.

continua na próxima página

Tabela 24.1000 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Selecionar feições por extensão (definido na camada de entrada SRC) Opcional	`SPAT`	[extensão] Padrão: Não definido	Você pode selecionar características de uma determinada extensão que estarão na tabela de saída. Os métodos disponíveis são: • Calcular a partir da camada…: utiliza a extensão de uma camada carregada no projeto atual • Calculado a partir do layout do mapa…: usa extensão do tipo *Item do layout do mapa* no projeto ativo. • Calcular a partir do bookmark…: usa a extensão de uma já salvada *bookmark* • Usar extensão de tela de mapa • Desenhe na tela: Clique e arreste um retângulo delimitando a área que deve ser levada em consideração. • Inserir as coordenadas como `xmin, xmax, ymin, ymax`
Recortar a camada de entrada usando a extensão acima (retângulo)*	`CLIP`	[boleano] Padrão: Falso	A camada de entrada será recortada pela extensão que você definiu anteriormente
Campos a serem incluídos (deixe em branco para usar todos os campos) Opcional	`FIELDS`	[cadeia] [lista] Padrão: []	Define campos a serem mantidos do arquivo vetorial importado. Se nenhum for selecionado, todos os campos são importados.
Select features using a SQL "WHERE" statement (Ex: column="value") Opcional	`WHERE`	[string] Padrão: ''	Define com uma declaração SQL "WHERE" quais características devem ser selecionadas para a tabela de saída
Agrupar N feições por transação (Padrão: 2000) Opcional	`GT`	[string] Padrão: ''	Você pode agrupar as feições de entrada em transações em que N define o tamanho. Por padrão, N limita o tamanho da transação a 20.000 recursos.
Sobrescrever a tabela existente	`OVERWRITE`	[boleano] Padrão: Verdadeiro	Se houver uma tabela com o mesmo nome no banco de dados, e se esta opção estiver configurada para Verdadeiro tabela será sobrescrita.
Adicionar para a tabela existente	`APPEND`	[boleano] Padrão: Falso	Se marcado / Verdadeiro, os dados vetoriais serão anexados a uma tabela existente. Novos campos encontrados na camada de entrada são ignorados. Por padrão, uma nova tabela será criada.

continua na próxima página

Tabela 24.1000 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Alterar e acrescentar novos campos à tabela existente	`ADDFIELDS`	[boleano] Padrão: Falso	Se ativado, os dados vetoriais serão anexados a uma tabela existente, não será criada uma nova tabela. Novos campos encontrados na camada de entrada são adicionados à tabela. Por padrão, uma nova tabela será criada.
Do not launder columns/table names	`LAUNDER`	[boleano] Padrão: Falso	Com esta opção marcada, você pode evitar o comportamento padrão (converter nomes de colunas para minúsculas, remover espaços e outros caracteres inválidos).
Não criar Índice Espacial	`INDEX`	[boleano] Padrão: Falso	Impede que um índice espacial para a tabela de saída seja criado. Por padrão, um índice espacial é adicionado.
Continuar após uma falha, pulando a feição com falha	`SKIPFAILURES`	[boleano] Padrão: Falso	Continue after a failure, skipping the failed feature.
Validate geometries based on Simple Features specification	`MAKEVALID`	[boleano] Padrão: Falso	Applies cleanup operations on geometries to ensure they are valid regarding the rules of the Simple Features specification.
Promover para Multipartes	`PROMOTETOMULTI`	[boleano] Padrão: Verdadeiro	Converte o tipo de geometria de recursos para várias partes na tabela de saída
Mantenha a largura e precisão dos atributos de entrada	`PRECISION`	[boleano] Padrão: Verdadeiro	Evita modificar os atributos da coluna para cumprir os dados de entrada
Opções de criação adicionais Opcional	`CREATION_OPTIONS` (for QGIS <= 3.42, this was `OPTIONS`)	[string] Padrão: '' (sem opções adicionais)	Opções adicionais de criação de GDAL.

Saídas

Este algoritmo não tem saída.

Código Python

Algorithm ID: `gdal:importvectorintopostgisdatabasenewconnection`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Informações vetoriais

Creates an information file that lists information about an OGR-supported data source. The output will be shown in a 'Result' window and can be written into a HTML-file. The information includes the geometry type, feature count, the spatial extent, the projection information, the list of attributes and related type, list of relations and field domains, and many more.

This algorithm is derived from the GDAL ogrinfo utility.

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Enable listing of all layers in the dataset	`ALL_LAYERS`	[boleano] Padrão: Falso	If checked, QGIS will output the information of all the layers in the input dataset instead of information of the first layer only. Convenient when a multi layer format (GeoPackage, GML, KML, SpatiaLite, SQLite, …) is used as input.
Somente a saída resumida	`SUMMARY_ONLY`	[boleano] Padrão: Verdadeiro	Suppress listing of individual features and show only summary information like projection, schema, feature count and extents.
Suprimir informação dos metadados	`NO_METADATA`	[boleano] Padrão: Falso	Suppress metadata printing. Some datasets may contain a lot of metadata strings.
Informação da camada	`OUTPUT`	[html] Padrão: `[Salvar em arquivo temporário]`	Specify the output HTML file that includes the file information. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo… Se nenhum arquivo HTML for definido, a saída será gravada em um arquivo temporário

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Additional command line parameters Opcional	`EXTRA`	[string] Padrão: '' (sem opções adicionais)	Adicionar opções extras de linha de comando GDAL

Saídas

Rótulo	Nome	Tipo	Descrição
Informação da camada	`OUTPUT`	[html]	O arquivo HTML de saída que inclui as informações do arquivo.

Código Python

Algorithm ID: `gdal:ogrinfo`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

Vector Information (JSON)

Creates an information file that lists information about an OGR-supported data source. The output will be shown in a 'Result' window and can be written into a `.JSON` file. The information includes the geometry type, feature count, the spatial extent, the projection information, the list of attributes and related type, list of relations and field domains, and many more.

This algorithm is derived from the GDAL ogrinfo utility.

Atenção

Running this algorithm requires QGIS installed with GDAL >= 3.7 (see *Help* ► *About* menu).

Parâmetros

Parâmetros básicos

Rótulo	Nome	Tipo	Descrição
Camada de entrada	`INPUT`	[vector: geometry]	Camada vetorial de entrada
Enable listing of all layers in the dataset	`ALL_LAYERS`	[boleano] Padrão: Falso	If checked, QGIS will output the information of all the layers in the input dataset instead of information of the first layer only. Convenient when a multi layer format (GeoPackage, GML, KML, SpatiaLite, SQLite, …) is used as input.
Enable listing of features	`FEATURES`	[boleano] Padrão: Falso	
Suprimir informação dos metadados	`NO_METADATA`	[boleano] Padrão: Falso	Suppress metadata printing. Some datasets may contain a lot of metadata strings.

continua na próxima página

Tabela 24.1004 – continuação da página anterior

Rótulo	Nome	Tipo	Descrição
Informação da camada	`OUTPUT`	[arquivo] Padrão: `[Salvar em arquivo temporário]`	Specify the output JSON file that includes the file information. *One of*: • Salvar em Arquivo Temporário • Salvar em Arquivo… If no JSON file is defined the output will be written to a temporary file

Parâmetros avançados

Rótulo	Nome	Tipo	Descrição
Additional command line parameters Opcional	`EXTRA`	[string] Padrão: '' (sem opções adicionais)	Adicionar opções extras de linha de comando GDAL

Saídas

Rótulo	Nome	Tipo	Descrição
Informação da camada	`OUTPUT`	[arquivo]	The output *.JSON* file that includes the file information.

Código Python

Algorithm ID: `gdal:ogrinfojson`

```
import processing
processing.run("algorithm_id", {parameter_dictionary})
```

O *id do algoritmo* é exibido quando você passa o mouse sobre o algoritmo na Caixa de ferramentas de processamento. O *dicionário de parâmetros* fornece os nomes e valores dos parâmetros. Veja *Usando os algoritmos do processamento a partir do Terminal Python.* para detalhes sobre como executar algoritmos de processamento a partir do console Python.

CAPÍTULO 25

Complementos

25.1 Complementos QGIS

O QGIS foi projetado com uma arquitetura de plugins. Isso permite que muitos novos recursos e funções sejam facilmente adicionados ao aplicativo. Alguns dos recursos do QGIS são realmente implementados como plugins.

25.1.1 Core and External plugins

Os plug-ins do QGIS são implementados como **Plugins principais** ou **Plugins externos**.

Core Plugins are maintained by the QGIS Development Team and are automatically part of every QGIS distribution. They are written in one of two languages: **C++** or **Python**.

Most of External Plugins are currently written in Python. They are stored either in the 'Official' QGIS Repository at https://plugins.qgis.org/plugins/ or in external repositories and are maintained by the individual authors. Detailed documentation about the usage, minimum QGIS version, home page, authors, and other important information are provided for the plugins in the Official repository. For other external repositories, documentation might be available with the external plugins themselves. External plugins documentation is not included in this manual.

To install or activate a plugin, go to *Plugins* menu and select *Manage and install plugins...*. Installed external python plugins are placed under the `python/plugins` folder of the active *user profile* path.

Paths to Custom C++ plugins libraries can also be added under *Settings* ► *Options* ► *System*.

25.1.2 Diálogo de Complementos

A aba Configurações

At the bottom of the left panel, the *Settings* tab is the main place you can configure which plugins can be displayed in your application. You can use the following options:

- *Check for Updates on Startup*. QGIS will automatically check for plugin updates at startup, at most once every 3 days. Whenever an installed plugin has an update available, QGIS will inform you.
- *Show also Experimental Plugins*. QGIS will show you plugins in early stages of development, which are generally unsuitable for production use. For these plugins, you can install either the stable or the experimental version, and at any moment switch from one to the other.
- *Show also Deprecated Plugins*. These plugins are usually unmaintained because they have replacement functions in QGIS, a lack of maintainers, they rely on functions that are no longer available in QGIS… They are generally unsuitable for production use and appear grayed in the plugins list.

By default, in the *Plugin Repositories* section, QGIS provides you with its official plugin repository with the URL `https://plugins.qgis.org/plugins/plugins.xml?qgis=version` (where `<version>` represents the exact QGIS version you are running). To add external author repositories, click *Add…* and fill in the *Repository Details* form with a name and the URL. The URL can be of `http://` or `file://` protocol type.

The default QGIS repository is an open repository and you don't need any authentication to access it. You can however deploy your own plugin repository and require an authentication (basic authentication, PKI). You can get more information on QGIS authentication support in *Autenticação* chapter.

If you do not want one or more of the added repositories, they can be disabled from the Settings tab via the *Edit…* button, or completely removed with the *Delete* button.

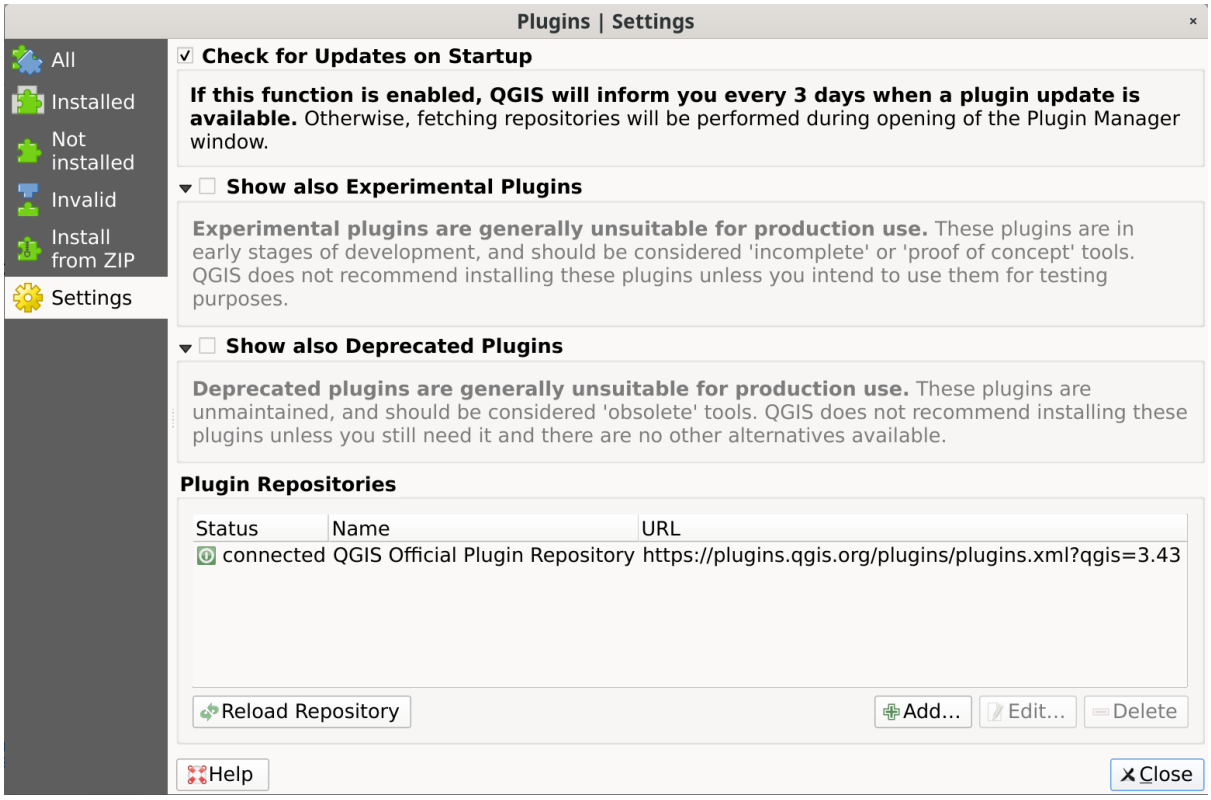


Figura25.1: The *Settings* tab

Browsing the plugins

The tabs

The upper tabs in the *Plugins* dialog provide you with lists of plugins based on their install, creation or update status. Depending on the plugins settings, available tabs can be:

- *All*: shows all the available plugins in the enabled repositories
- *Installed*: shows both the plugins you installed and the core plugins that are installed by default and you can not uninstall
- *Not installed*: shows uninstalled or not yet installed plugins in the enabled repositories
- *New*: shows plugins released since the last *Check for Updates on Startup*
- *Upgradeable*: shows installed plugins that have published a more recent version in the repository
- *Invalid*: shows all installed plugins that are currently broken for any reason (missing dependency, errors while loading, incompatible functions with QGIS version…)

At the top of the tabs, a *Search* function helps you find any plugin using metadata information (author, name, description, tag,…).

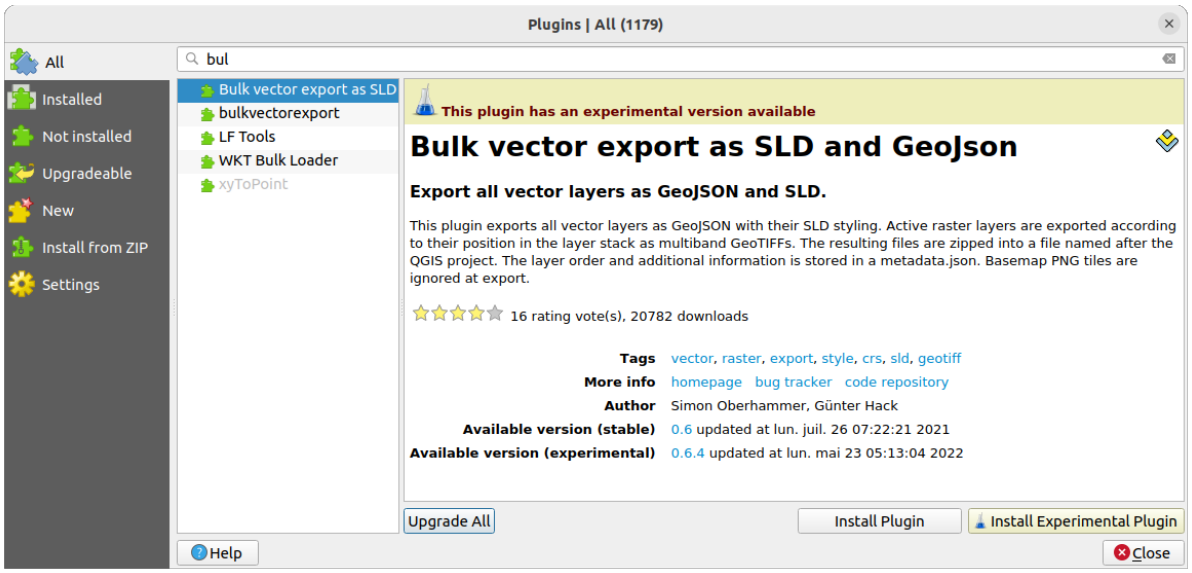


Figura25.2: Searching a plugin from the *All* tab

The Plugins

Select a plugin and you will have some metadata displayed in the right panel:

- information on whether the plugin is experimental or has an experimental version available (if *Show also Experimental Plugins* is checked)
- summary and description
- rating vote(s) (you can vote for your preferred plugin!)
- etiquetas
- alguns links úteis para a página inicial, rastreador e repositório de código

- autor(es)
- version(s) available with link to download page in the repository, or path to local folder for installed plugins

The *Plugin Manager* dialog allows you to interact with the latest version of the plugins. When enabled, the experimental version can be shown only if it is more recent than the latest stable version. Depending on the active tab, whether the selected plugin is installed, you will be given some of the following options:

- *Install*: installs the latest stable version of the selected plugin
- *Install Experimental Plugin*: installs the experimental version of the selected plugin
- *Reinstall Plugin*: installs the same stable version of the plugin e.g. after it has failed to load
- *Reinstall Experimental Plugin*: installs the same experimental version of the plugin e.g. after it has failed to load
- *Upgrade Plugin*: upgrades selected plugin to its latest stable version
- *Upgrade Experimental Plugin*: upgrades selected plugin to its experimental version
- *Upgrade All*: upgrades all installed plugins to their more recent stable or experimental version (depending on whether their previously installed version was stable or experimental).
- *Downgrade Plugin*: moves from the experimental version of the plugin to its previous stable version
- *Downgrade Experimental Plugin*: moves from an experimental version of the plugin to its latest published experimental version. This may occur when playing with a not yet published version.
- *Uninstall Plugin*: removes the installed plugin from the user profile

An installed plugin displays a checkbox on its left. Uncheck it to temporarily deactivate the plugin.

Right-click on a plugin in the list and you will be able to sort the plugins list by various metadata. The new order applies to all the tabs. Sort options are:

- *Sort by Name*
- *Sort by Downloads*
- *Sort by Vote*
- *Sort by Status*
- *Sort by Date Created*
- *Sort by Date Updated*

The Install from ZIP tab

The *Install from ZIP* tab provides a file selector widget to import plugins in a zipped format, e.g. plugins downloaded directly from their repository. Encrypted files are supported.

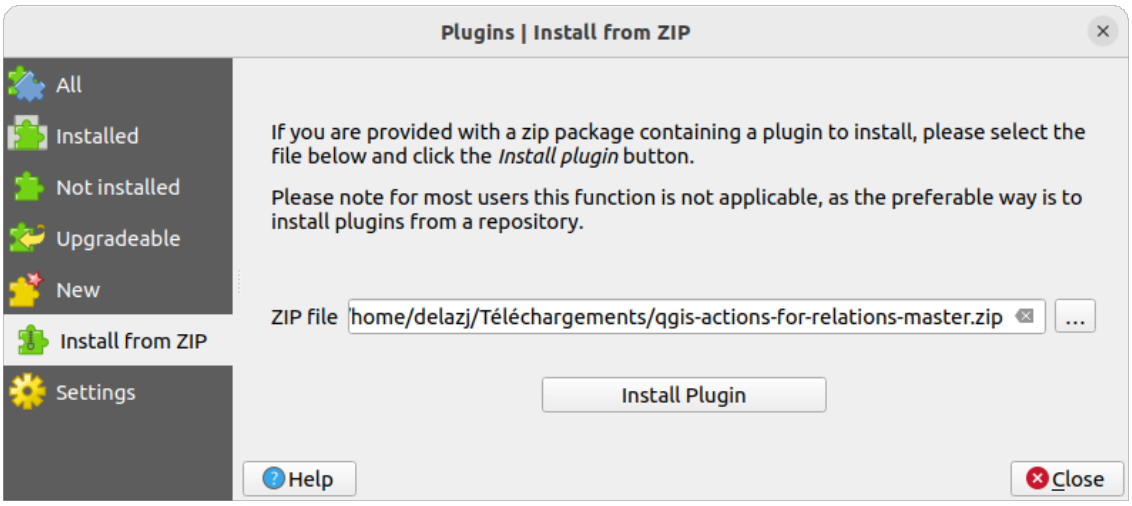


Figura25.3: The *Install from zip* tab

25.2 Usando os plug-ins do núcleo QGIS

25.2.1 Plugin do gerenciador de banco de dados

The DB Manager Plugin is intended to be the main tool to integrate and manage spatial database formats supported by QGIS (PostgreSQL, SpatiaLite, GeoPackage, Oracle Spatial, Virtual layers) in one user interface. The DB Manager Plugin provides several features. You can drag layers from the QGIS Browser into the DB Manager, and it will import your layer into your spatial database. You can drag and drop tables between spatial databases and they will get imported.

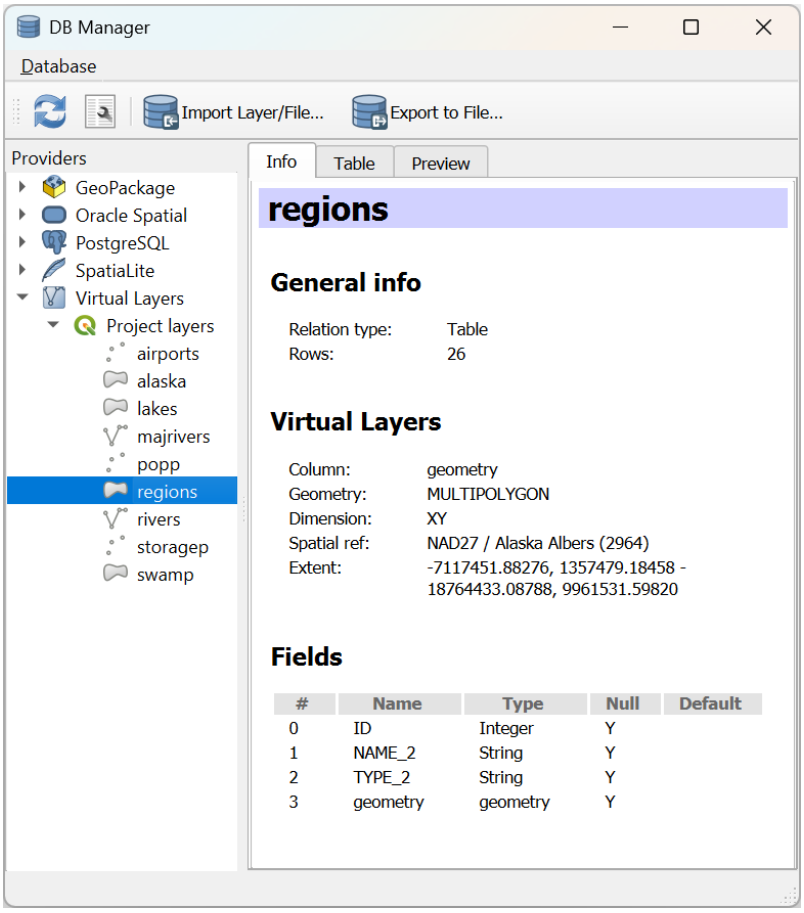


Figura25.4: Caixa de diálogo Gerenciador DB

The *Database* menu allows you to connect to an existing database, to start the SQL window and to exit the DB Manager Plugin. Once you are connected to an existing database, the menus *Schema* (relevant for DBMSs, such as PostgreSQL) and *Table* will appear.

O menu :Seleção de menu:*Esquema* inclui ferramentas para criar e excluir (somente se vazios) esquemas e, se a topologia estiver disponível (por exemplo, com topologia PostGIS), para iniciar um *Visualizador de Topo*.

O menu : seleção de menu:*Tabela* permite que você crie e edite tabelas e exclua tabelas e visualizações. Também é possível esvaziar tabelas e mover tabelas entre esquemas. Você pode *Executar análise de vácuo* para a tabela selecionada. *Vácuo* recupera espaço e o disponibiliza para reutilização, e *analisa* atualiza estatísticas que são usadas para determinar a maneira mais eficiente de executar uma consulta. *Alterar registro…* permite que você adicione suporte ao alterar registro a uma tabela. Finalmente, você pode *importar camada/arquivo…* e *Exportar para arquivo…*.

Nota

Usando o Gerenciador de banco de dados é possível adicionar comentários para tabelas e colunas de um Banco de Dados PostgreSQL.

A janela *Provedores* lista todos os bancos de dados existentes suportados pelo QGIS. Com um clique duplo, você pode se conectar ao banco de dados. Com o botão direito do mouse, você pode renomear e excluir esquemas e tabelas existentes. As tabelas também podem ser adicionadas à tela do QGIS com o menu de contexto.

Se conectado a um banco de dados, a janela **principal** do Gerenciador de banco de dados oferece quatro guias. A aba *Info* fornece informações sobre a tabela e sua geometria, bem como sobre campos, restrições e índices existentes. Ele permite que você crie um índice espacial em uma tabela selecionada. A aba *Tabela* mostra a tabela, e a aba *Visualizar* renderiza as geometrias como visualizadas. Quando você abre uma *Janela SQL*, ela será colocada em uma nova aba.

Trabalhando com a janela SQL

Você pode usar o Gerenciador de banco de dados para executar consultas SQL em seu banco de dados espacial. As consultas podem ser salvas e carregadas, e lá o *Construtor de consultas SQL* o ajudará a formular suas consultas. Você pode até mesmo visualizar a saída espacial marcando *Carregar como nova camada* e especificando *Coluna(s) com valores exclusivos* (IDs), *Coluna de geometria* e *Nome da camada (prefixo)». É possível destacar uma parte do SQL para executar apenas essa parte pressionando :kbd:`Ctrl+R* ou clicando no botão *Executar*.

QGIS also adds support for the `REGEXP` function in some providers. This allows users to use regular expressions in SQL filters or expressions, for example:

```
SELECT * FROM places WHERE name REGEXP '^A';
```

This returns all features where the `name` field starts with the letter A.

After executing your query, you can select specific cells in the result set. Use the `Ctrl+C` shortcut to copy the selected cells to the clipboard. The copied data is available as a formatted table. This allows you to paste the data into other applications, such as spreadsheet where it will show up as a table.

O botão *Histórico de consultas* armazena as últimas 20 consultas de cada banco de dados e provedor.

Clicar duas vezes em uma entrada adicionará o fragmento à janela SQL.

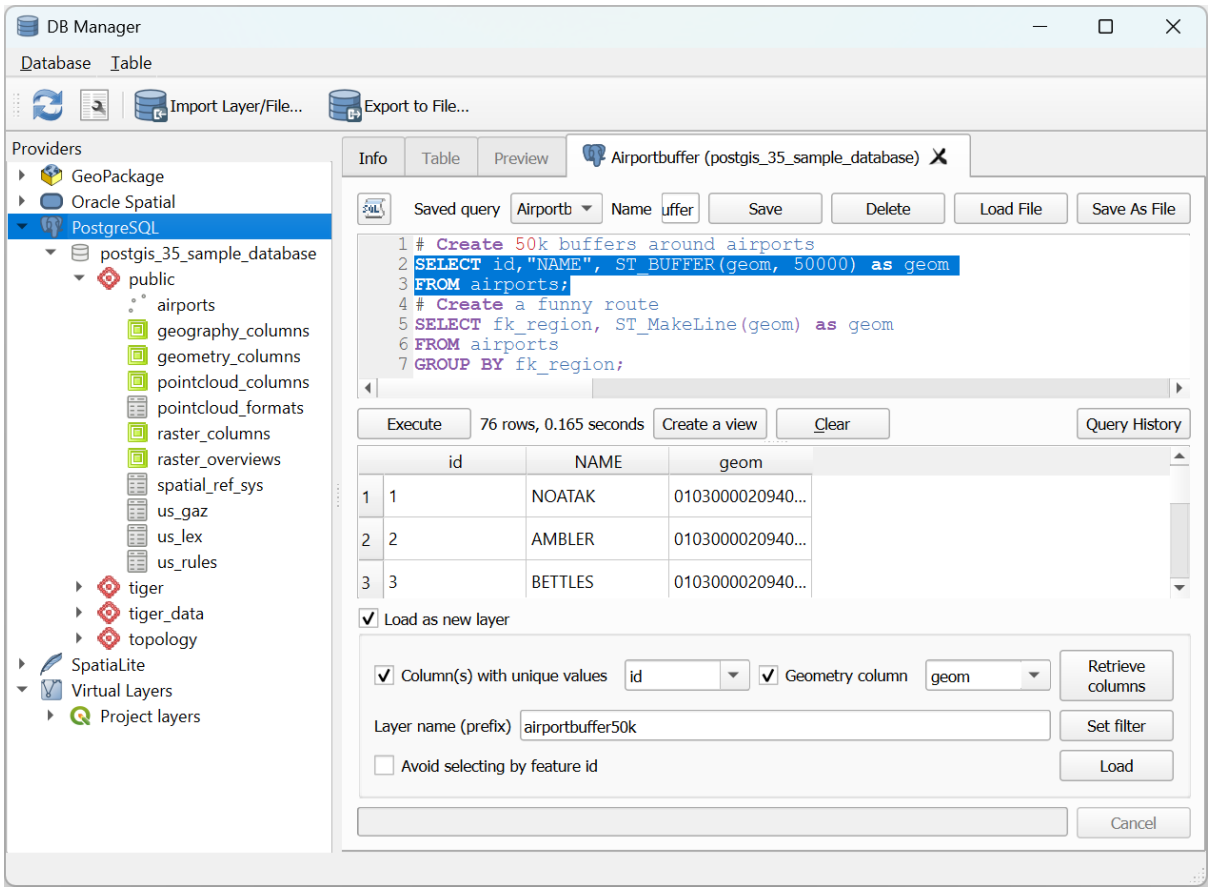


Figura25.5: Executando consultas SQL na janela SQL do Gerenciador de banco de dados

Nota

A Janela SQL também pode ser usada para criar Camadas Virtuais. Nesse caso, em vez de selecionar um banco de dados, selecione **Camadas QGIS** em **Camadas Virtuais** antes de abrir a janela SQL. Veja vetor_camadas_virtuais para instruções sobre a sintaxe SQL a ser usada.

Import Vector Layer

You can import layer or file into your database. Here are the parameters you can set for the import process:

- *Input*: Select the layer or file to import. Using the dropdown menu, select from the list of loaded layers in QGIS or click on the … button to select a file from disk. Check the *Import only selected features* to import only the selected features of the layer.
- *Output table*: Choose the *Schema* and provide a name for the new table.
- *Options*: Here are some options for the import process:
  - *Primary key*: Provide naming for the primary key field. By default, it is named `id`.
  - *Geometry column*: Provide naming for the geometry column. By default, it is named `geom`.
  - *Source SRID*: Define the SRID for the geometry column. By default, it uses the layer's CRS.
  - *Target SRID*: Define the target SRID to reproject the geometries during the import process. By default, it uses the layer's CRS.
  - *Encoding*: Define the encoding of the source data. By default, it uses `Automatic`. It is QGIS's automatic detection mode that attempts to guess the file's character encoding based on available metadata or system locale.
  - *Replace destination table (if exists)*: If a table with the same name already exists in the selected schema, it will be replaced.
  - *Do not promote to multi-part*: Geometries will be imported as single-part geometries.
  - *Convert field names to lower case*: All field names will be converted to lower case.
  - *Create spatial index*: A spatial index will be created on the geometry column after import.
  - *Comment*: Add comments to table. Only available for PostgreSQL databases.

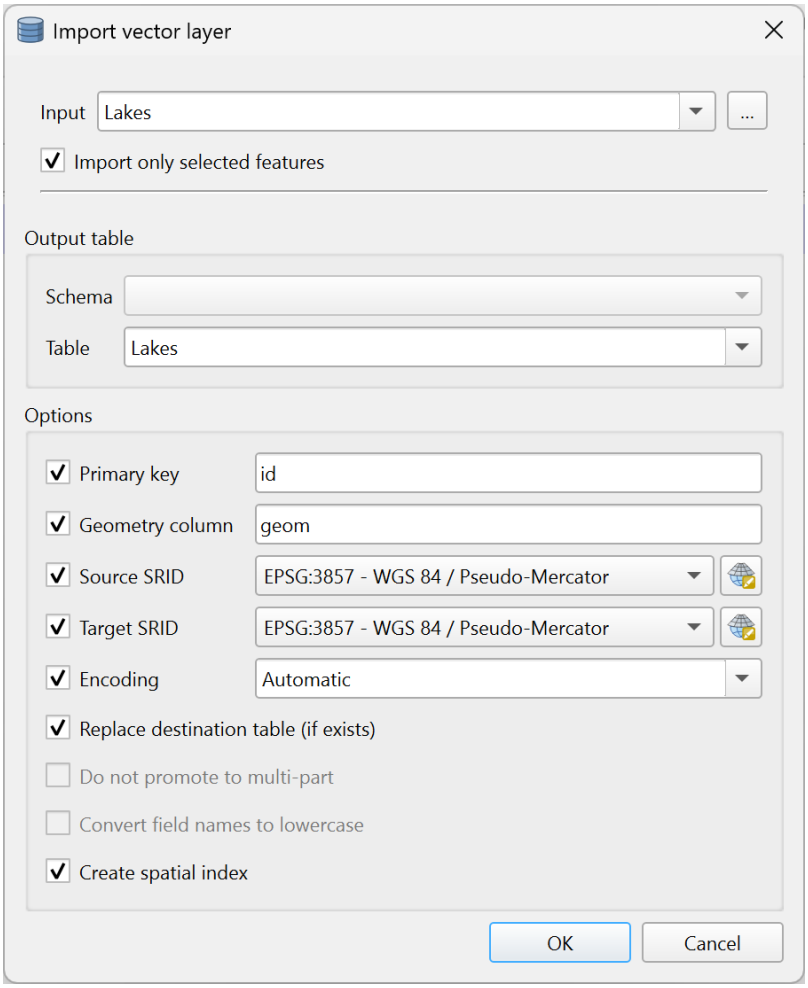


Figura25.6: Importing a vector layer into a spatial database using DB Manager

Export to Vector File

To export a table from your database to a vector file, select the desired *Format* and *Save as* location. Under *Options*, you can set the following parameters:

- *Source SRID*: Define the source SRID of the geometry column. By default, it uses the layer's CRS.
- *Target SRID*: Define the target SRID to reproject the geometries during the export process. By default, it uses the layer's CRS.
- *Encoding*: Define the encoding of the output data.
- *Replace destination table (if exists)*: If a table with the same name already exists at the selected location, it will be replaced.

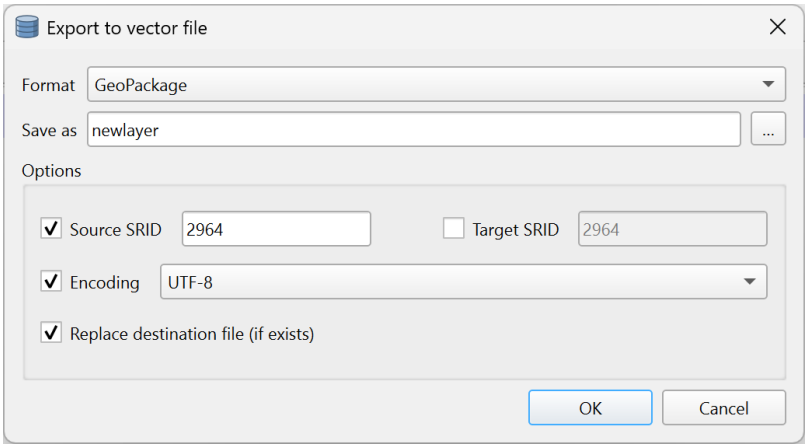


Figura25.7: Exporting a table to a vector file using DB Manager

25.2.2 Plugin do Verificador de Geometria

Verificador de geometria é um poderoso plugin central para verificar e corrigir a validade da geometria de uma camada. Ele está disponível no menu :seleção de menu:*Vetor* (**|verificador de geometria|** :seleção de menu:*Verificar geometrias…*).

Configurando as verificações

A caixa de diálogo *Verificar geometrias* mostra diferentes configurações agrupadas na primeira aba (*Configuração*):

- *Inserir camadas vetoriais*: para selecionar as camadas a serem verificadas. Uma **|caixa de seleção|** A caixa de seleção *Somente feições selecionadas* pode ser usada para restringir a verificação às geometrias das feições selecionadas.
- *Tipos de geometria permitidos* dá a chance de restringir o tipo de geometria da(s) camada(s) de entrada para:
  - **|caixa de seleção|** Apontar
  - **|caixa de seleção|** Multiponto
  - **|caixa de seleção|** Linha
  - **|caixa de seleção|** Multilinha
  - **|caixa de seleção|** Polígono
  - **|caixa de seleção|** Multipolígono
- *Validade geométrica*. Dependendo dos tipos de geometria, você pode escolher entre:
  - **|caixa de seleção|** *Autointerseções*
  - **|caixa de seleção|** *Nós duplicados*
  - **|caixa de seleção|** *Autocontatos*
  - **|caixa de seleção|** *Polígono com menos de 3 nós.*
- *Propriedades da geometria*. Dependendo dos tipos de geometria, diferentes opções estão disponíveis:
  - **|caixa de seleção|** *Polígonos e multipolígonos não podem conter buracos*
  - **|caixa de seleção|** *Os objetos multipartes devem consistir em mais de uma parte*
  - **|caixa de seleção|** *As linhas não devem ter oscilações*

- *Condições geométricas*. Permite adicionar alguma condição para validar as geometrias com:
  - **|caixa de seleção|** *Comprimento mínimo do segmento (unidades de mapa)* **|selecione o número|**
  - **|caixa de seleção|** *Ângulo mínimo entre o segmento (graus)* **|selecione o número|**
  - **|caixa de seleção|** *Área mínima do polígono (unidades do mapa sqr.)* **|selecione o número|**
  - **|caixa de seleção|** *Sem polígonos sliver* com um *Máxima finura* **|selecione o número|** e uma **|caixa de seleção|** *Máx. area (unidades do mapa sqr.)* **|selecione o número|**
- *Verificações de topologia*. Dependendo dos tipos de geometria, muitas opções diferentes estão disponíveis:
  - **|caixa de seleção|** *Verifica se há duplicatas*
  - **|caixa de seleção|** *Verifica se há recursos dentro de outros recursos*
  - **|caixa de seleção|** *Verifica se há sobreposições menores que* **|selecione o número|**
  - **|caixa de seleção|** *Procura lacunas menores que* **|selecione o número|**
  - caixa de seleção| *Os pontos devem ser cobertos por linhas*
  - **|caixa de seleção|** *Os pontos devem estar dentro de um polígono*
  - **|caixa de seleção|** *As linhas não devem cruzar outras linhas*
  - **|caixa de seleção|** *As linhas não devem cruzar com os recursos da camada* **|selecione fragmento|**
  - **|caixa de seleção|** *Os polígonos devem seguir os limites da camada* **|Selecione o fragmento|**
- *Tolerância*. Você pode definir a tolerância do check-in das unidades da camada do mapa.
- *Camada de vetor de saída* dá a opção de:
  - **|botão de rádio ligado|** *Modificar camada de entrada*
  - **|botão de rádio ligado|** *Criar novas camadas*

Quando estiver satisfeito com a configuração, você pode clicar no botão *Run*.

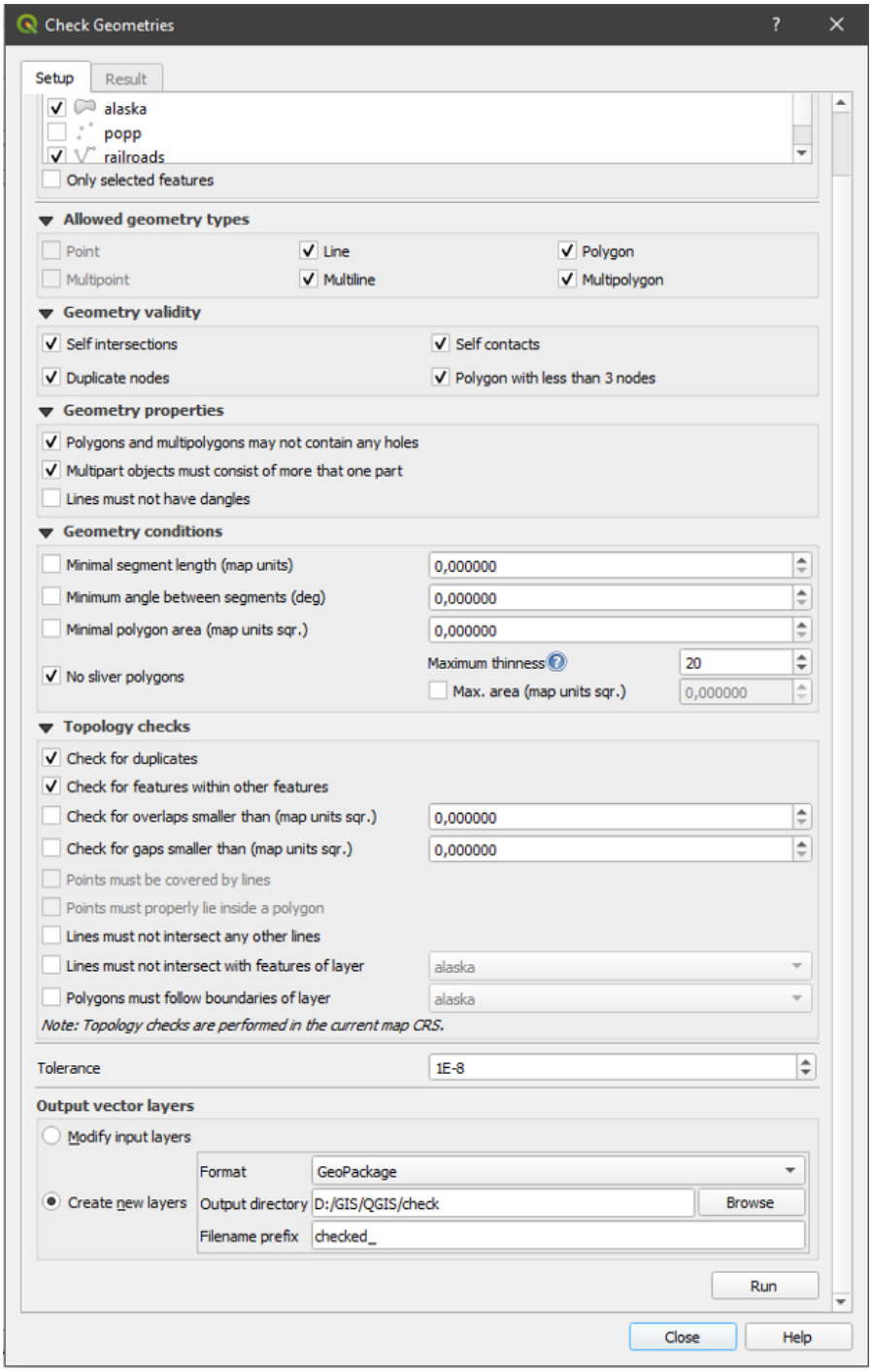


Figura25.8: O plug-in do verificador de geometria

O *Plugin do Verificador de Geometria* pode encontrar os seguintes erros:

- Autointerseções: um polígono com uma autointerseção
- Nós duplicados: dois nós duplicados em um segmento
- Buracos: buraco em um polígono
- Comprimento do segmento: um comprimento de segmento inferior a um limite
- Ângulo mínimo: dois segmentos com um ângulo inferior a um limite
- Área mínima: área do polígono inferior a um limite

- Polígono prateado: este erro vem de um polígono muito pequeno (com área pequena) com um perímetro grande
- Recursos duplicados
- Recurso dentro do recurso
- Sobreposições: sobreposição de polígonos
- Lacunas: lacunas entre polígonos

A figura a seguir mostra as diferentes verificações feitas pelo plugin.

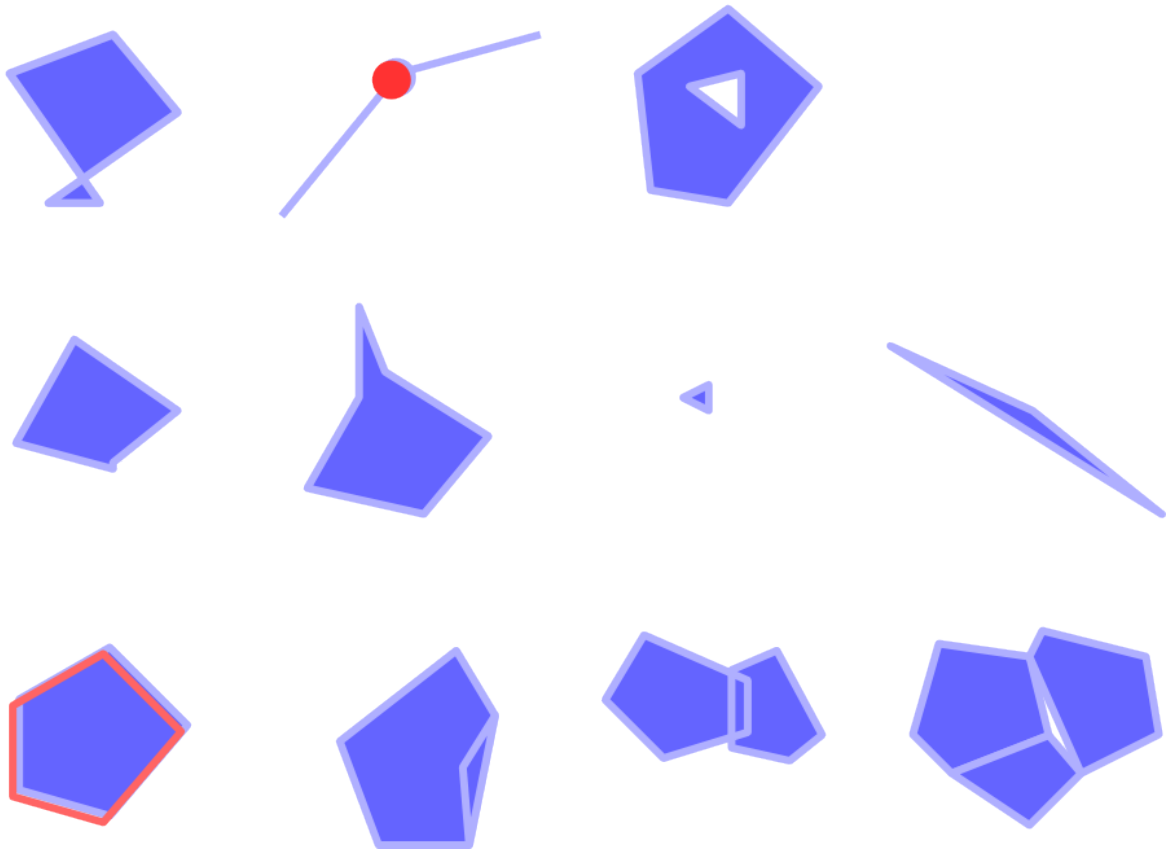


Figura25.9: Algumas verificações suportadas pelo plugin

Analisando os resultados

Os resultados aparecem na segunda aba (*Resultado*) e como uma camada de visão geral dos erros na tela (seu nome tem o prefixo padrão **:arquivo:`verificado_`**). Uma tabela lista o *Resultado da verificação de geometria* com um erro por linha e colunas contendo: o nome da camada, um ID, o tipo de erro, depois as coordenadas do erro, um valor (dependendo do tipo do erro) e finalmente a coluna de resolução que indica a resolução do erro. Na parte inferior desta tabela, você pode *Exportar* o erro em diferentes formatos de arquivo. Você também tem um contador com o número de erros totais e corrigidos.

Você pode selecionar uma linha para ver o local do erro. Você pode alterar esse comportamento selecionando outra ação entre **|botão de rádio ligado|** *Erro* (padrão), **|botão de rádio desligado|** *Recurso*, **|botão de rádio desligado|** *Não mova*, e **|caixa de seleção|** *Realçar recursos selecionados*.

Abaixo da ação de zoom ao clicar na linha da tabela, você pode:

- **|do recurso selecionado|** *Show selected features in attribute table*
- **|sucesso|** *Corrigir erros selecionados usando a resolução padrão*

- |sucesso| *Corrigir erros selecionados, solicitar método de resolução* Você verá uma janela para escolher o método de resolução entre os quais:
  - Mesclar com o polígono vizinho com a aresta compartilhada mais longa
  - Mesclar com o polígono vizinho com maior área
  - Mesclar com o polígono vizinho com valor de atributo idêntico, se houver, ou deixar como está
  - Excluir recurso
  - Nenhuma ação
- |opções| *Configurações de resolução de erros* permite que você altere o método de resolução padrão dependendo do tipo de erro

> **Dica**
>
> **Corrigir vários erros**
>
> Você pode corrigir vários erros selecionando mais de uma linha na tabela com a ação *CTRL + clique*.

Finalmente, você pode escolher qual *Atributo usar ao mesclar recursos por valor de atributo*.

25.2.3 Cliente do Catálogo Metabusca

Introducción

MetaSearch is a QGIS plugin to interact with metadata catalog services, supporting both the OGC API - Records and OGC Catalog Service for the Web (CSW) standards.

O MetaSearch fornece uma abordagem fácil e intuitiva e uma interface amigável para pesquisar catálogos de metadados no QGIS.

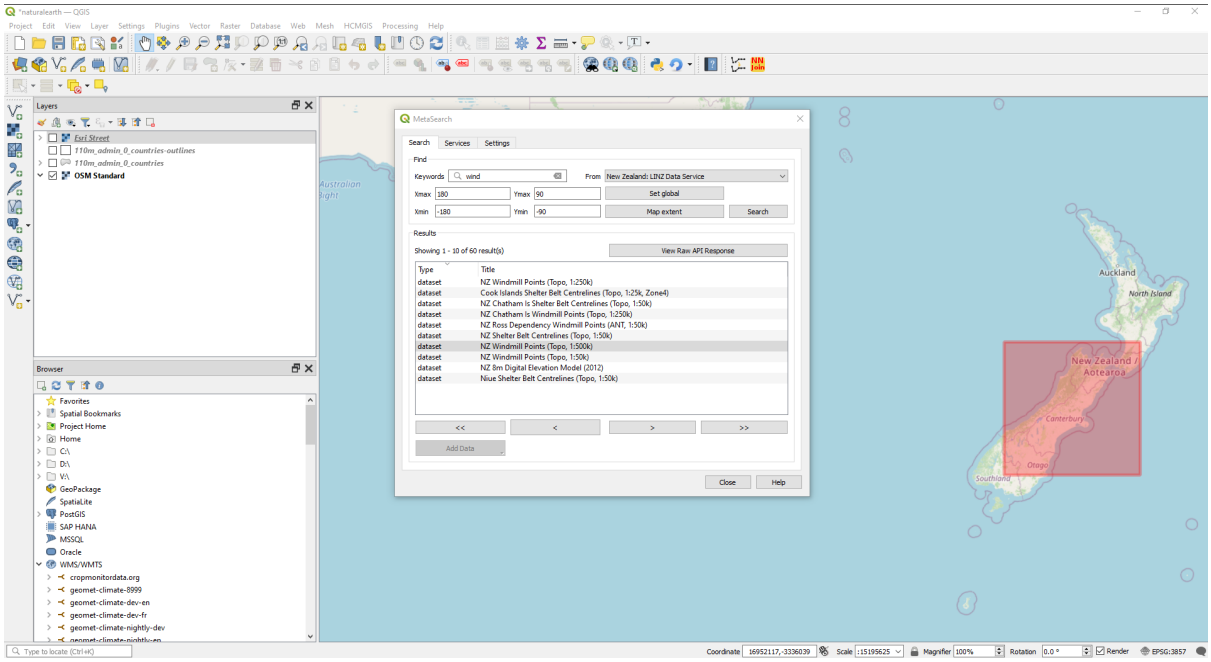


Figura25.10: Pesquisa e resultados de Serviços no MetaSearch

Trabalhando com Catálogos de Metadados no QGIS

O MetaSearch é incluído por padrão no QGIS, com todas as suas dependências, e pode ser ativado a partir do QGIS Gerenciador de plug-ins.

OGC API - Records

OGC API - Records is an OGC (Open Geospatial Consortium) standard for the discovery of geospatial resources on the Web.

CSW (Serviço de Catálogo para a Web)

CSW (Serviço de Catálogo para a Web) é uma especificação OGC (Consórcio Geoespacial Aberto) que define interfaces comuns para descobrir, navegar e consultar metadados sobre dados, serviços e outros recursos potenciais.

Startup

Para iniciar o MetaSearch, clique no botão ícone ou selecione :selecionar menu:*Web* –> *MetaSearch* –> *MetaSearch* através do menu principal do QGIS. A caixa de diálogo MetaSearch aparecerá. A GUI principal consiste em três abas: *Serviços*, *Procurar* e *Definições*.

Gerenciando Serviços de Catálogo

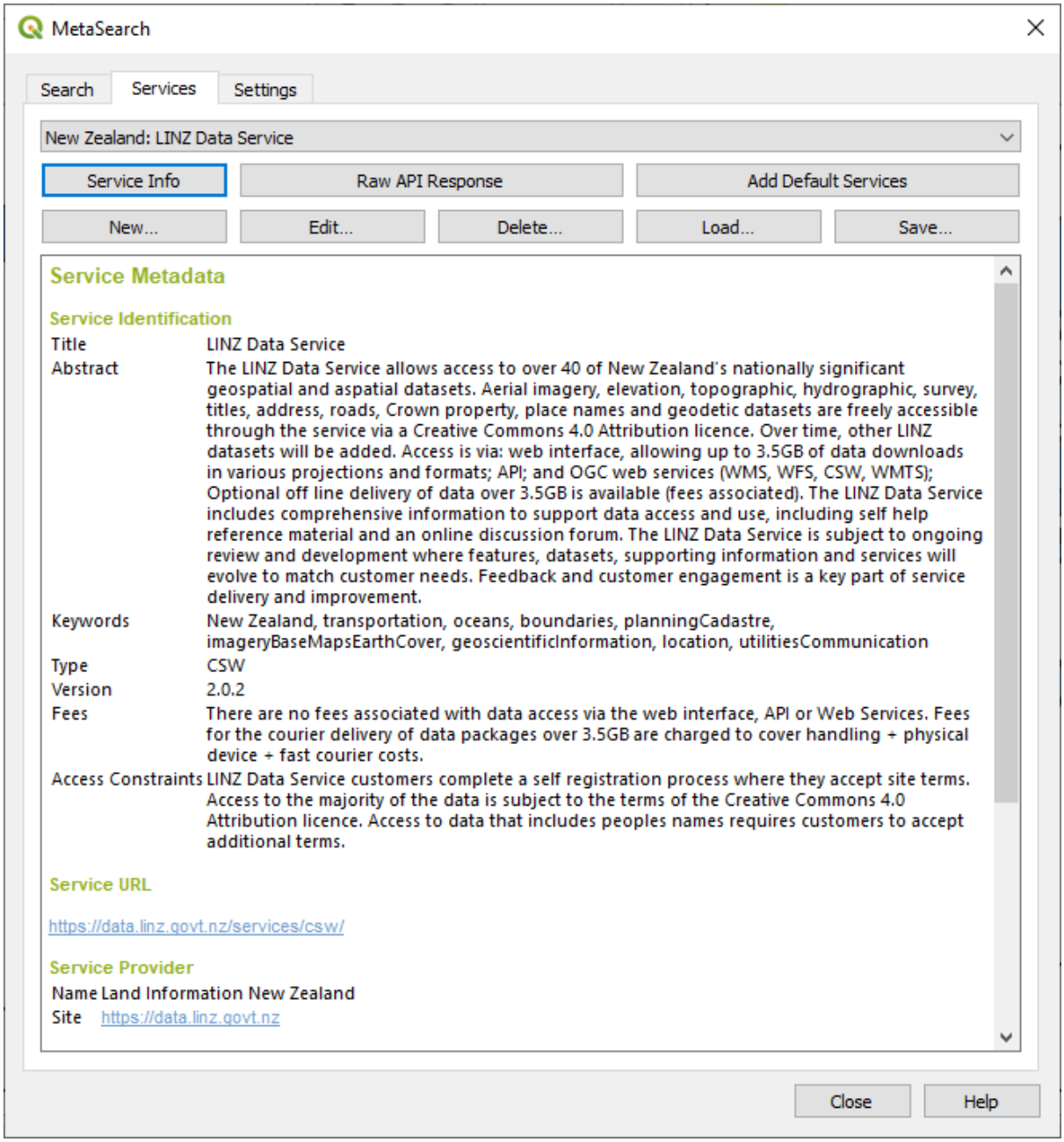


Figura25.11: Gerenciando Serviços de Catálogo

A aba *Serviços* permite ao usuário gerenciar todos os serviços de catálogo disponíveis. O MetaSearch fornece uma lista padrão de Serviços de Catálogo, que podem ser adicionados pressionando o botão *Adicionar serviços padrão*.

Para localizar todas as entradas listadas do Serviço de Catálogo, clique na caixa de seleção suspensa.

Para adicionar uma entrada do Serviço de Catálogo:

1. Clique no botão *Novo*
2. Enter a *Name* for the service, as well as the *URL* (endpoint). Note that for OGC CSW 2.0.2 Catalogs, only the base URL is required (not a full GetCapabilities URL). For OGC API - Records Catalogs, the URL should be the path to the collection endpoint
3. If the Catalog requires authentication, enter the appropriate *User name* and *Password* credentials.

4. Clique em *OK* para adicionar o serviço à lista de entradas.

Para editar uma entrada existente do Serviço de Catálogo:

1. Selecione a entrada que deseja editar
2. Clique no botão *Editar*
3. E modifique os valores *Nome* ou *URL*
4. Haz clic en *Aceptar*.

Para excluir uma entrada do Serviço de Catálogo, selecione a entrada que deseja excluir e clique no botão *Delete*. Você será solicitado a confirmar a exclusão da entrada.

MetaSearch permite carregar e salvar conexões em um arquivo XML. Isso é útil quando você precisa compartilhar configurações entre aplicativos. Abaixo está um exemplo do formato de arquivo XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<qgsCSWConnections version="1.0">
    <csw type="OGC CSW 2.0.2" name="Data.gov CSW" url="https://catalog.data.gov/
↪csw-all"/>
    <csw type="OGC CSW 2.0.2" name="Geonorge - National CSW service for Norway"␣
↪url="https://www.geonorge.no/geonetwork/srv/eng/csw"/>
    <csw type="OGC CSW 2.0.2" name="Geoportale Nazionale - Servizio di ricerca␣
↪Italiano" url="http://www.pcn.minambiente.it/geoportal/csw"/>
    <csw type="OGC CSW 2.0.2" name="LINZ Data Service" url="http://data.linz.govt.
↪nz/feeds/csw"/>
    <csw type="OGC CSW 2.0.2" name="Nationaal Georegister (Nederland)" url="http://
↪www.nationaalgeoregister.nl/geonetwork/srv/eng/csw"/>
    <csw type="OGC CSW 2.0.2" name="RNDT - Repertorio Nazionale dei Dati␣
↪Territoriali - Servizio di ricerca" url="http://www.rndt.gov.it/RNDT/CSW"/>
    <csw type="OGC CSW 2.0.2" name="UK Location Catalogue Publishing Service" url=
↪"http://csw.data.gov.uk/geonetwork/srv/en/csw"/>
    <csw type="OGC CSW 2.0.2" name="UNEP/GRID-Geneva Metadata Catalog" url="http://
↪metadata.grid.unep.ch:8080/geonetwork/srv/eng/csw"/>
</qgsCSWConnections>
```

Para carregar uma lista de entradas:

1. Clique no botão *Carregar*. Uma nova janela aparecerá.
2. Clique no botão *Navegar* e navegue até o arquivo XML de entradas que deseja carregar.
3. Clique em *Abrir*. A lista de entradas será exibida.
4. Selecione as entradas que deseja adicionar da lista e clique em *Carregar*.

Click the *Service Info* button to display information about the selected Catalog Service such as service identification, service provider and contact information. If you would like to view the raw API response, click the *Raw API Response* button. A separate window will open displaying server information in raw JSON or XML format.

Pesquisando serviços de catálogo

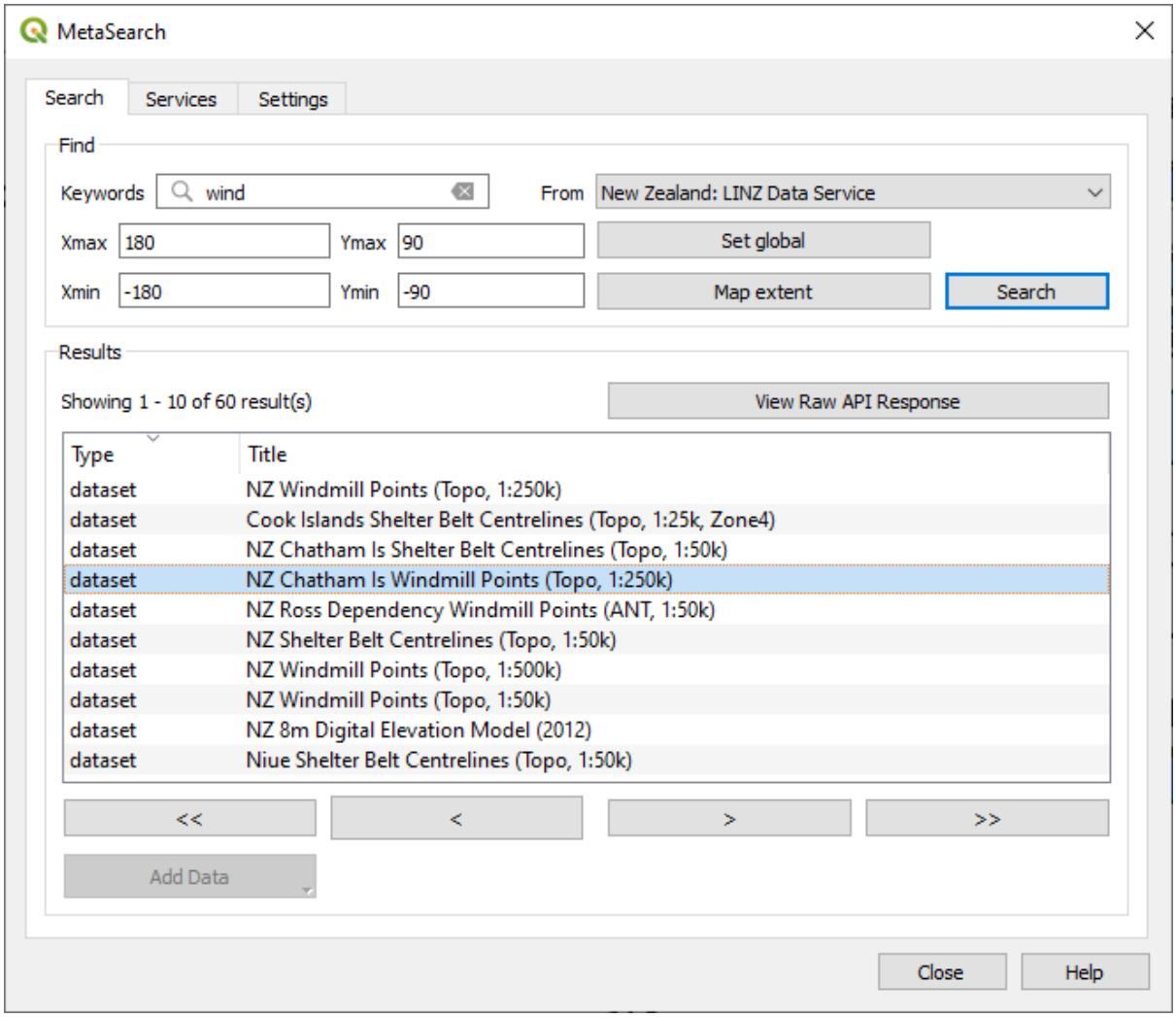


Figura25.12: Pesquisando serviços de catálogo

A guia *Procurar* permite que o usuário consulte os Serviços de Catálogo para dados e serviços, defina vários parâmetros de pesquisa e visualize os resultados.

Os seguintes parâmetros de pesquisa estão disponíveis:

- *Palavras-chave*: palavras-chave de pesquisa de texto livre;
- *A partir de*: o Serviço de Catálogo para realizar a consulta;
- **Caixa delimitadora**: a área espacial de interesse a ser filtrada, definida por *Xmax*, *Xmin*, *Ymax* e *Ymin*. Clique em *Definir global* para fazer uma busca global, clique em *Extensão do mapa* para fazer uma busca na área visível, ou insira os valores manualmente.

Clicar no botão *Procurar* pesquisará o Catálogo de Metadados selecionado. Os resultados da pesquisa são exibidos em uma lista e podem ser classificados clicando no cabeçalho da coluna. Você pode navegar pelos resultados da pesquisa com os botões direcionais abaixo dos resultados da pesquisa.

Selecione um resultado e:

- Click the *View Raw API Response* button to open a window with the service response in raw JSON or XML format.

- Se o registro de metadados tiver uma caixa delimitadora associada, uma área de cobertura da caixa delimitadora será exibida no mapa.
- Clique duas vezes no registro para exibir os metadados do registro com quaisquer links de acesso associados. Clicar em um link abre o link no navegador da Web do usuário.
- If the record is a supported web service (WMS/WMTS, WFS, WCS, ArcGIS REST Service, etc.) or a GIS File, the *Add Data* button will be enabled. When clicking this button, MetaSearch will verify if this is a valid OWS. The service will then be added to the appropriate QGIS connection list, and the appropriate connection dialog will appear. If you choose *Add GIS File* the linked GIS file will be added to the current project.

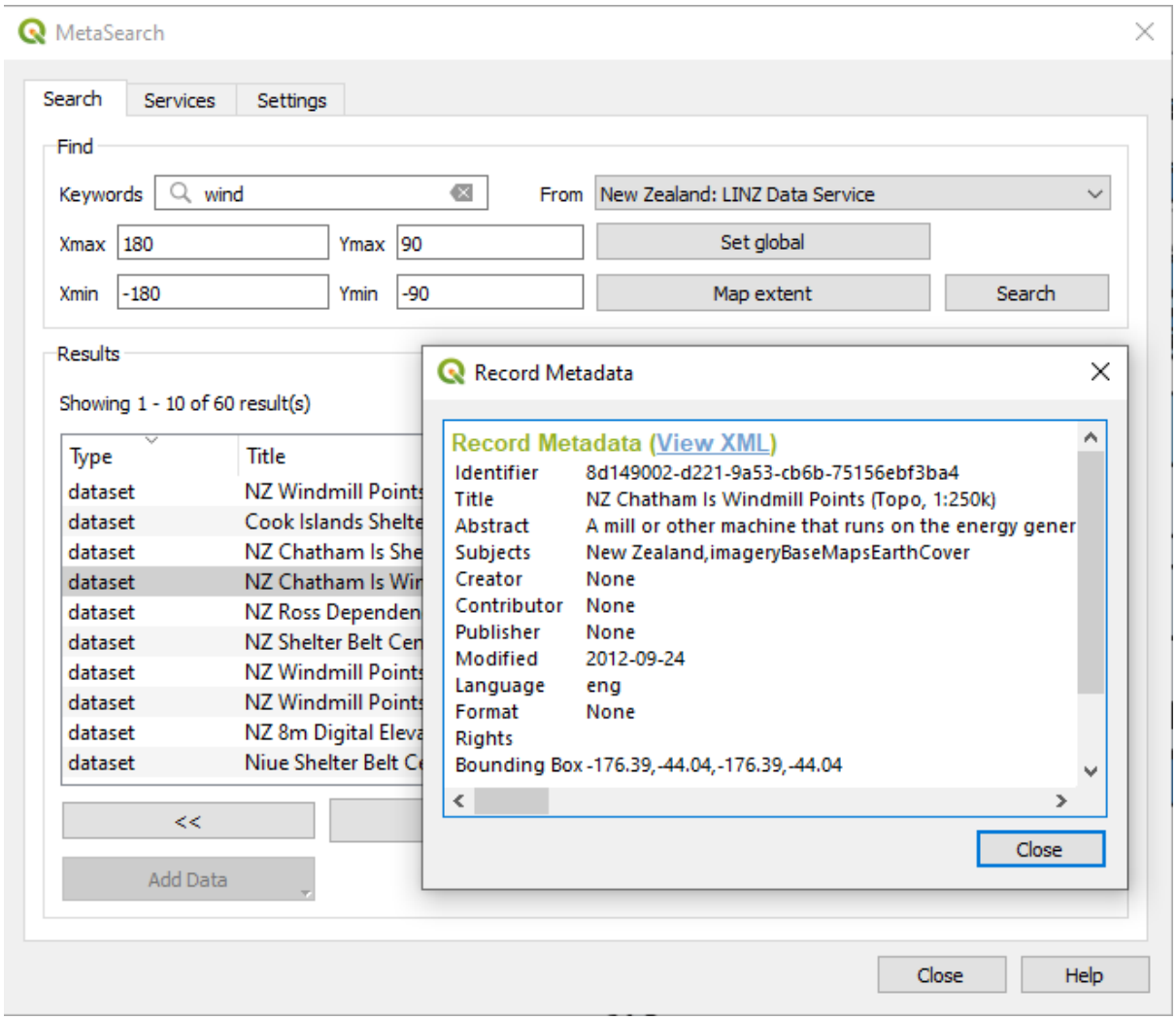


Figura25.13: Exibição de registro de metadados

Configurações

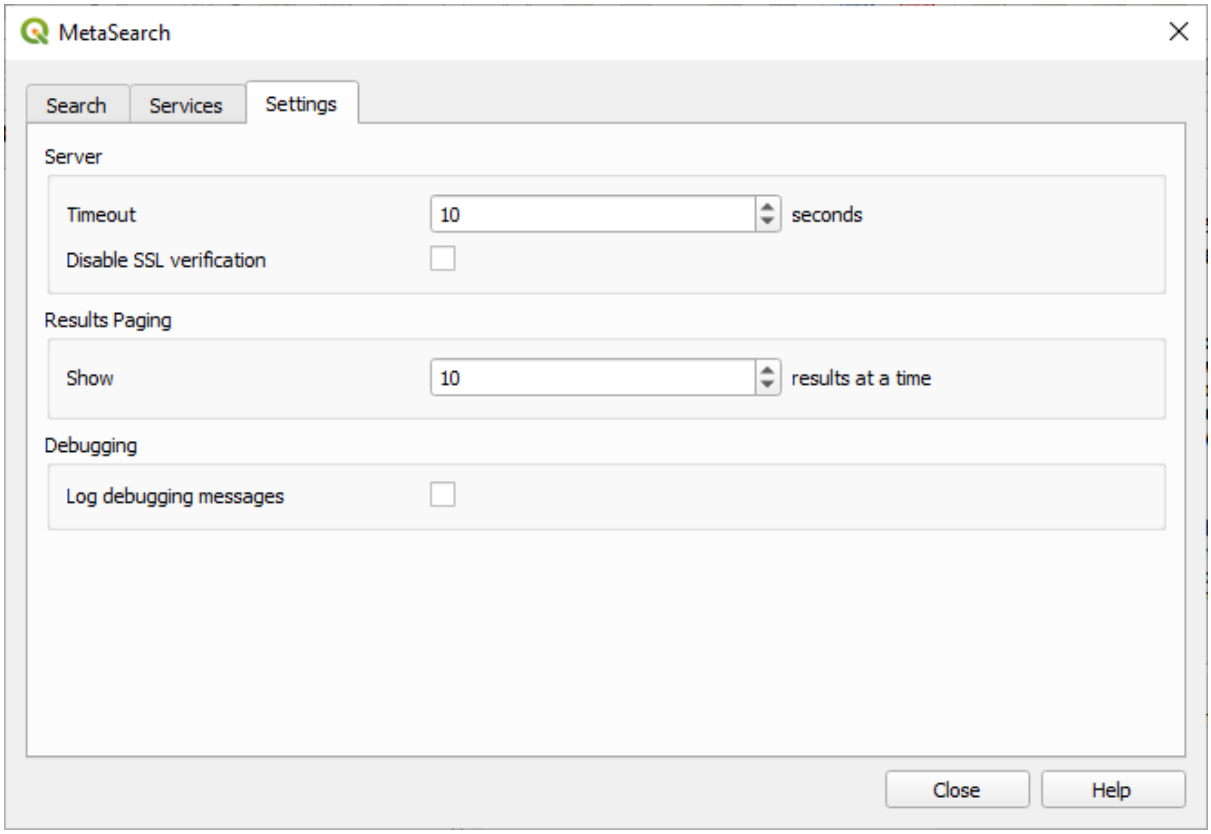


Figura25.14: Configurações de metapesquisa

Você pode ajustar o MetaSearch com o seguinte *Definições*:

- *Tempo limite do servidor*: ao pesquisar catálogos de metadados, o número de segundos para bloquear a tentativa de conexão. O valor padrão é 10.
- *Desativar verificação SSL*: opção para desativar a verificação SSL.
- *Paginação de resultados*: ao pesquisar catálogos de metadados, o número de resultados a serem exibidos por página. O valor padrão é 10.
- *Log debugging messages*: option to enable the logging of debugging messages in the *Log Messages Panel*.

Catalog Server Errors

In some cases, the Catalog will work in a web browser, but not in MetaSearch. This may be due to the Catalog server's configuration/setup. Catalog server providers should ensure URLs are consistent and up to date in their configuration (this is common in HTTP -> HTTPS redirection scenarios). Please see the pycsw FAQ item for a deeper explanation of the issue and fix. Although the FAQ item is pycsw specific it can also apply in general to other Catalog servers.

25.2.4 Plugin de edição fora de rede

For data collection, it is a common situation to work with a laptop or a cell phone offline in the field. Upon returning to the network, the changes need to be synchronized with the master datasource (e.g., a PostgreSQL database). If several persons are working simultaneously on the same datasets, it is difficult to merge the edits by hand, even if people don't change the same features.

O **|Cópia de edição fora de rede|** O plug-in Edição fora de rede automatiza a sincronização copiando o conteúdo da fonte de dados para um banco de dados SpatiaLite ou GeoPackage e armazenando as edições fora de rede em tabelas dedicadas. Após estar conectado à rede novamente, é possível aplicar as edições fora de rede ao conjunto de dados mestre.

Para usar o plug-in:

1. Open a project with some vector layers (e.g., from an Esri Shapefile, PostgreSQL or WFS-T datasource).
2. Assumindo que você já ativou o plugin (veja núcleo_e_plugins externos) vá para :seleção de menu:*Database* –> *Edição fora da rede* –> **|Cópia de edição fora da rede|** *Converter para projeto fora da rede*. A caixa de diálogo epônimo é aberta.
3. Selecione o *Tipo de armazenamento*. Pode ser do tipo *GeoPackage* ou *SpatiaLite*
4. Use o botão *Procurar* para indicar a localização do banco de dados no qual armazenar os *Dados fora da rede*. Pode ser um arquivo existente ou um para criar.
5. Na seção *Selecionar camadas remotas*, marque as camadas que deseja salvar. O conteúdo das camadas é salvo nas tabelas do banco de dados.

   Nota

   Como os formatos de banco de dados de destino não têm suporte de lista nativo, o plug-in de edição fora de rede transforma campos de lista {fragmento, número} em campos de fragmentos onde os valores são separados por vírgulas. Isso permite ler e editar o conteúdo desses campos quando estiver fora de rede.

   Se você quiser lidar com o campo da camada original e da camada fora de rede, você pode confiar nas funções de expressão experimentar() 1 ` e :ref:`variedade 2, por exemplo:

   ```
   try(array_contains("field",1),array_contains(string_to_array("field"),1))
   ```

6. Você pode marcar **|caixa de seleção|** *Sincronize somente feições selecionadas se uma seleção estiver presente* permitindo somente salvar e trabalhar em um subconjunto. Pode ser inestimável no caso de grandes camadas.

   Isso é tudo!
7. Salve seu projeto e leve-o para o campo.
8. Edite as camadas fora de rede.
9. Após estar conectado novamente, carregue as alterações usando :seleção de menu:*Base de dados* –> *Edição fora de rede* –> **|Sincronização de edição fora de rede|** *Sincronizar*.

Nota

As camadas usadas fora de rede são marcadas com o **|indicador fora de rede|** ícone no painel *Camadas*.

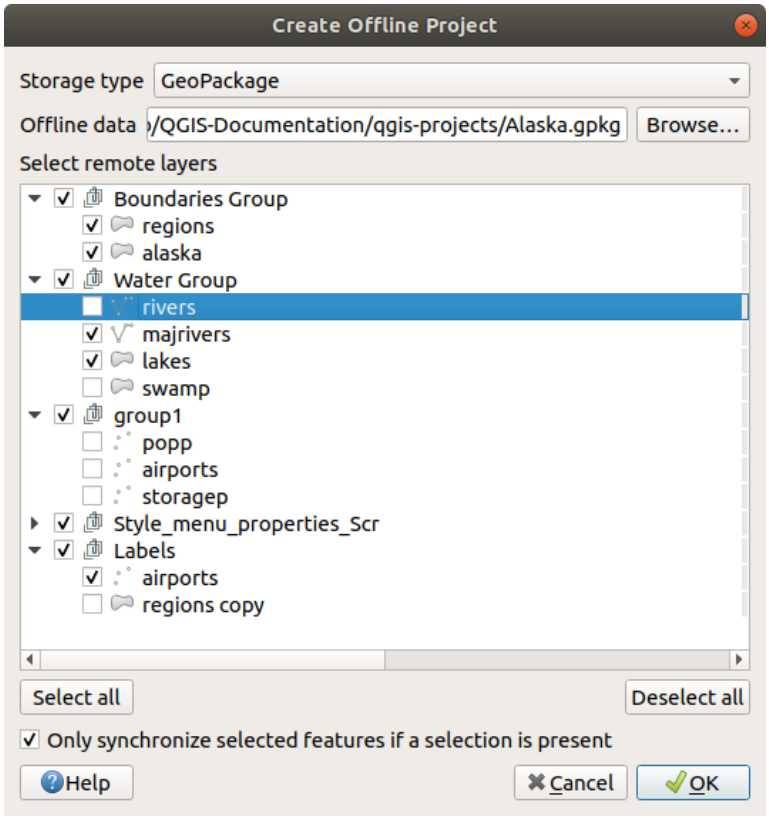


Figura25.15: Criar um projeto fora de rede

25.2.5 Complementos Verificador de Topologia

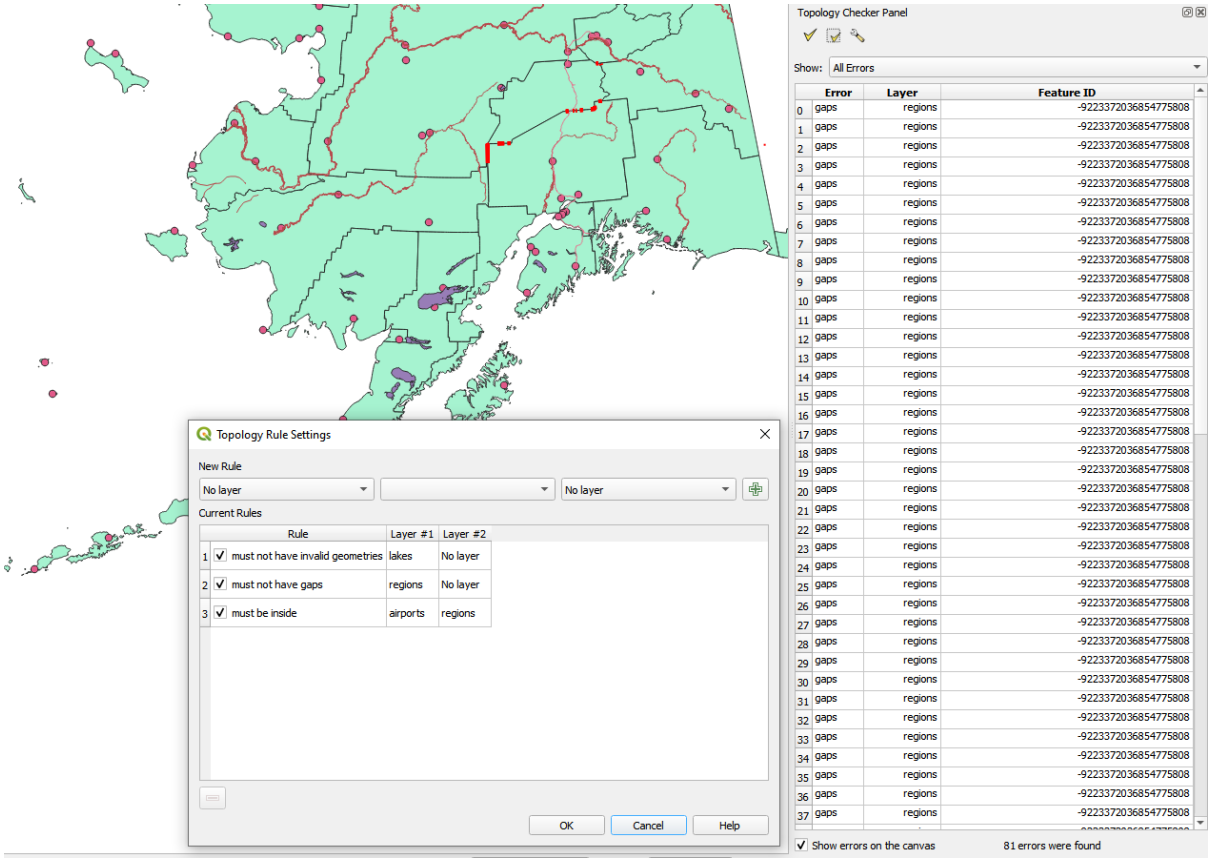


Figura25.16: O complemento verificador de topologia

A topologia descreve as relações entre pontos, linhas e polígonos que representam as feições de uma região geográfica. Com o complemento Verificador de topologia, você pode examinar seus arquivos vetoriais e verificar a topologia com várias regras de topologia. Essas regras verificam com as relações espaciais se seus recursos 'Igual', 'Contêm', 'Cobrir', são 'Coberto por', 'Cruzamento', são 'Desarticular', 'Intersecção', 'Sobreposição', 'Tocar' ou estão 'Dentro ' uns aos outros. Depende de suas perguntas individuais quais regras de topologia você aplica aos seus dados vetoriais (por exemplo, normalmente você não aceitará ultrapassagens em camadas de linha, mas se elas retratarem ruas sem saída, você não as removerá de sua camada vetorial).

O QGIS possui um recurso de edição topológica integrado, que é ótimo para criar novos recursos sem erros. Mas os erros de dados existentes e os erros induzidos pelo usuário são difíceis de encontrar. Este complemento ajuda você a encontrar esses erros por meio de uma lista de regras.

To enable the *Topology checker* plugin:

1. Go to *Plugins* menu
2. Open *Manage and Install plugins* and choose *Installed*
3. Enable *Topology checker*
4. Close the *Plugin manager* dialog. A *Topology checker* entry is added to the *Vector* menu.

After you enabled *Topology checker* open it and choose *Configure* to create your topology rules.

Em **camadas de pontos**, as seguintes regras estão disponíveis:

- *Must be covered by*: Here you can choose a vector layer from your project. Points that aren't covered by the given vector layer occur in the 'Error' field.

- *Must be covered by endpoints of*: Here you can choose a line layer from your project.
- *Must be inside*: Here you can choose a polygon layer from your project. The points must be inside a polygon. Otherwise, QGIS writes an 'Error' for the point.
- *Must not have duplicates*: Whenever a point is represented twice or more, it will occur in the 'Error' field.
- *Must not have invalid geometries*: Checks whether the geometries are valid.
- *Must not have multi-part-geometries*: All multi-part points are written into the 'Error' field.

Em **camadas de linha**, as seguintes regras estão disponíveis:

- *End points must be covered by*: Here you can select a point layer from your project.
- *Must not have dangles*: This will show the overshoots in the line layer.
- *Must not have duplicates*: Whenever a line feature is represented twice or more, it will occur in the 'Error' field.
- *Must not have invalid geometries*: Checks whether the geometries are valid.
- *Must not have multi-part geometries*: Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. All multi-part lines are written into the 'Error' field.
- *Must not have pseudos*: A line geometry's endpoint should be connected to the endpoints of two other geometries. If the endpoint is connected to only one other geometry's endpoint, the endpoint is called a pseudo node.

Em **camadas de polígonos**, as seguintes regras estão disponíveis:

- *Must contain*: Polygon layer must contain at least one point geometry from the second layer.
- *Must not have duplicates*: Polygons from the same layer must not have identical geometries. Whenever a polygon feature is represented twice or more it will occur in the 'Error' field.
- *Must not have gaps*: Adjacent polygons should not form gaps between them. Administrative boundaries could be mentioned as an example (US state polygons do not have any gaps between them…).
- *Must not have invalid geometries*: Checks whether the geometries are valid. Some of the rules that define a valid geometry are:
  - Os anéis de polígono devem fechar.
  - Os anéis que definem os furos devem estar dentro dos anéis que definem os limites externos.
  - Os anéis não podem se cruzar (eles não podem se tocar nem se cruzar).
  - Os anéis não podem tocar outros anéis, exceto em um ponto.
- *Must not have multi-part geometries*: Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. For example, a country consisting of multiple islands can be represented as a multi-polygon.
- *Must not overlap*: Adjacent polygons should not share common area.
- *Must not overlap with*: Adjacent polygons from one layer should not share common area with polygons from another layer.

When you create a *New rule* click on the Add rule to include it to the *Current rules*. You can enable or disable individual rules by clicking on the checkbox. Right-clicking over a rule provides the following options:

- *Select All* the rules
- *Activate* or *Deactivate* the selected rules
- *Toggle activation* of selected rules
- *Delete* selected rules. This can also be achieved with the Delete selected rules button.

Press *OK* and then choose from the *Topology checker* panel:

- Validate All: applies the active rules to all the features of the involved layer(s)
- or Validate Extent: applies the active rules to the features of the involved layer(s), within the current map canvas. The button is kept pushed and the results will update as the map canvas extent changes.

Errors will show up in the table of results containing type of error, layer and feature ID. Use *Filter errors by rule* menu to filter the errors to a specific error type.

Check *Show errors on the canvas* to show error location on the canvas. Clicking a row in the table will zoom the map canvas to the concerned feature, where you can use *QGIS digitizing tools* to fix the error.

Abaixo está a lista de plugins do núcleo fornecidos com o QGIS. Eles não são necessariamente habilitados por padrão.

Ícone	Plugar	Descrição	Referência manual
	DB Manager	Gerencie seus bancos de dados no QGIS	*Plugin do gerenciador de banco de dados*
	Verificador de Geometria	Verifique e repare erros em geometrias vetoriais	*Plugin do Verificador de Geometria*
	GRASS 7	Funcionalidade GRASS	*Integração com SIG GRASS*
	Fornecedor de GRASS GIS	Funcionalidade de processamento GRASS GIS	*Integração com SIG GRASS*
	Cliente do Catálogo Metabusca	Interagir com serviços de catálogo de metadados (CSW)	*Cliente do Catálogo Metabusca*
	Edição off-line	Edição offline e sincronização com banco de dados	*Plugin de edição fora de rede*
	Processamento	Estrutura de processamento de dados espaciais	*QGIS estrutura de processamento*
	Verificador de topologia	Encontre erros topológicos em camadas vetoriais	*Complementos Verificador de Topologia*

Nota

To use the Core Plugins GRASS 7, GRASS GIS provider, they have to be configured. Informations can be found *here*.

25.3 Terminal Python QGIS

Como você verá mais adiante neste capítulo, o QGIS foi projetado com uma arquitetura de plugin. Plugins podem ser escritos em Python, uma linguagem muito famosa no mundo geoespacial.

QGIS brings a Python API (see PyQGIS Developer Cookbook for some code sample) to let the user interact with its objects (layers, feature or interface). QGIS also has a Python console.

The QGIS Python Console is an interactive shell for Python command executions. It also has a Python file editor that allows you to edit and save your Python scripts. Both console and editor are based on PyQScintilla2 package. To open the console go to *Plugins* ► *Python Console* (`Ctrl+Alt+P`) or click on the Python Console icon in the *Plugins* toolbar.

25.3.1 O console interativo

The console is a Python interpreter that allows you to execute Python commands. Modules from QGIS (analysis, core, gui, server, processing, 3d) and Qt (QtCore, QtGui, QtNetwork, QtWidgets, QtXml) as well as Python's math, os, re and sys modules are already imported and can be used directly.

The interactive console is composed of a toolbar, an input area and an output one.

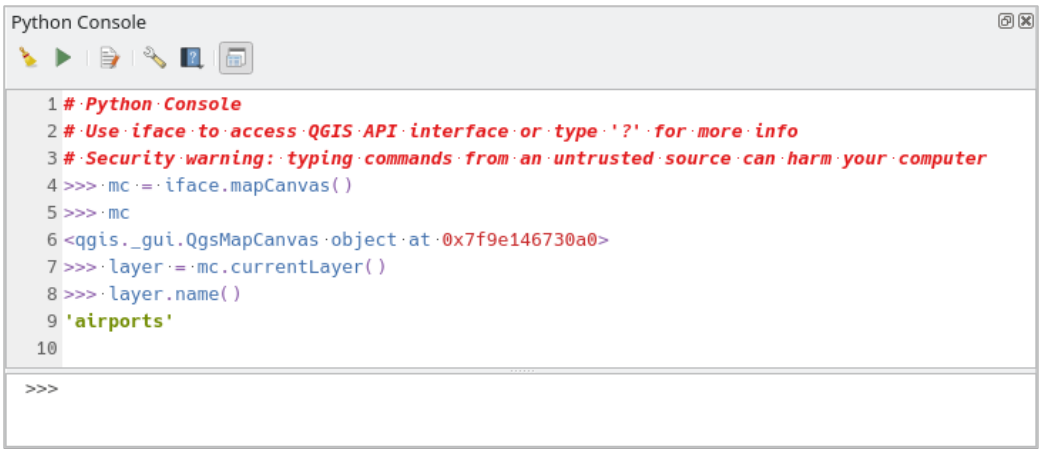


Figura25.17: O Console Python

Barra de Ferramentas

The toolbar proposes the following tools:

- Clear Console to wipe the output area;
- Run Command available in the input area: same as pressing `Enter`;
- Show Editor: toggles *O Editor de Código* visibility;
- Options...: opens a dialog to configure *console properties*;
- Help... provides a menu to access various documentation:
  - *Python Console Help* (the current page)
  - PyQGIS API documentation
  - PyQGIS Cookbook
- Dock Code Editor to dock or undock the panel in QGIS interface

Input area

The Console input area main features are:

- Code completion, highlighting syntax and calltips for the following APIs:
  - Python
  - PyQGIS
  - PyQt5
  - QScintilla2
  - osgeo-gdal-ogr

  - osgeo-geos

- `Ctrl+Alt+Space` para visualizar a lista de auto-preenchimento se habilitado no :ref:`console_options';
- Execute trechos de código da área de entrada digitando e pressionando `Enter` ou *Executar Comando*;
- Execute code snippets from the output area using the *Enter Selected* from the contextual menu or pressing `Ctrl+E`;
- Navegue pelo histórico de comandos da área de entrada utilizando as teclas de seta `Cima` e `Baixo` e execute o comando desejado;
- `Ctrl+Shift+Space' para visualizar o histórico de comandos: clicando duas vezes em uma linha, o comando será executado. O diálogo :guilabel:`Histórico de Comando` também pode ser acessado a partir do menu de contexto da área de entrada;
- Save and clear the command history. The history will be saved into the `console_history.txt` file under the active *user profile* folder;
- Type the following special commands:
  - `?` to show a help of the Python Console
  - `_api` to open QGIS C++ API documentation or `_api(object)` for a specific object documentation (in QGIS C++ API or Qt API documentation)
  - `_pyqgis` to open QGIS Python API documentation or `_pyqgis(object)` for a specific object documentation (in QGIS Python API or Qt API documentation)
  - `_cookbook` to open PyQGIS Cookbook.
  - `!` followed by a command to execute Shell commands from the Python Console. The console will start a subprocess, and forward its output to the Python Console Output. While the subprocess is running, the Python Console Input switches to STDIN mode and forwards entered character to the child process. This makes it possible to send confirmation when the child program asks for it. When the Console is in STDIN mode, pressing `Ctrl+C` kills the subprocess. It is also possible to affect the result of a command to a variable with the syntax `var = !cmd`

```
>>> !echo QGIS Rocks!
QGIS Rocks

>>> !gdalinfo --version
GDAL 3.10.3, released 2025/04/01

>>> !pip install black
# Install black python formatter using pip (if available)

>>> sql_formats = !ogrinfo --formats | grep SQL
>>> sql_formats
['SQLite -vector- (rw+v): SQLite / Spatialite', '  MSSQLSpatial -vector-␣
↪(rw+): Microsoft SQL Server Spatial Database', '  PostgreSQL -vector-␣
↪(rw+): PostgreSQL/PostGIS', '  MySQL -vector- (rw+): MySQL', '  PGDUMP -
↪vector- (w+v): PostgreSQL SQL dump']
```

Dica

Reutilize comandos executados do painel de saída

You can execute code snippets from the output panel by selecting some text and pressing `Ctrl+E`. No matter if selected text contains the interpreter prompt (`>>>`, `...`).

25.3.2 O Editor de Código

Use the Show Editor button in the Interactive Console to enable the editor widget. It allows editing and saving Python scripts and offers advanced functionalities to manage your code. Depending on the *enabled settings*, it provides various capabilities for easier code writing, such as code completion, highlighting syntax and calltips for supported APIs. Automatic indentation, parenthesis insertion, code commenting and syntax checking are also available.

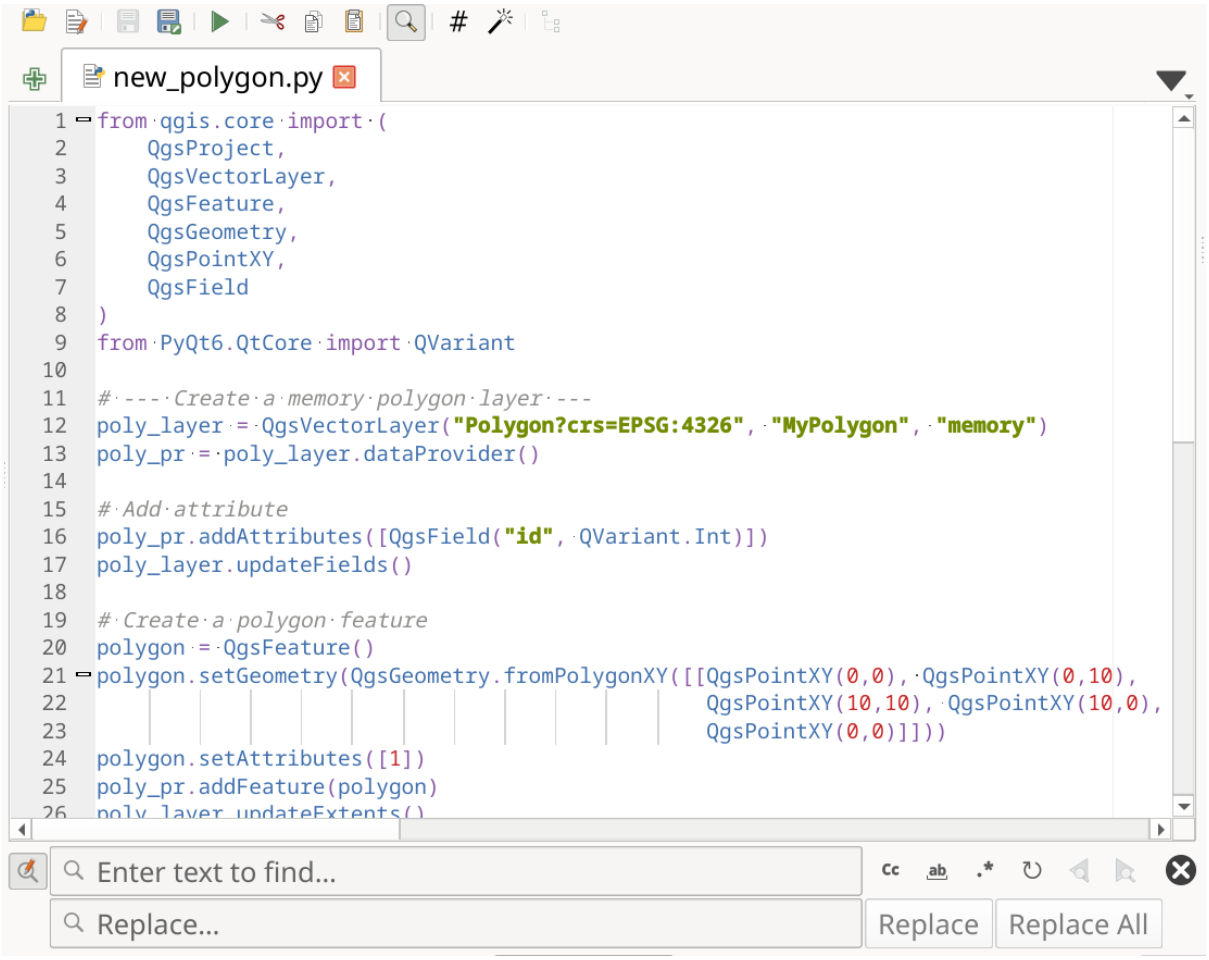


Figura25.18: O editor do console Python

The code editor area allows to simultaneously work on different scripts, each in a specific tab. Press New editor to add a new tab. You can run partially or totally a script from the *Code Editor* and output the result in the Interactive Console output area.

> **Dica**
>
> Press `Ctrl+Space` to view the auto-completion list.

At the top of the dialog, a toolbar provides access to a few commands. Right-clicking the editor area provides some more options. All available tools are described in the following table.

Tabela25.1: Python Console Editor Tools

Tool name	Function	Location
Open script…	Loads a Python file in the code editor dialog, as a new tab	Barra de Ferramentas
Open in external editor…	Opens a saved Python script in the default external program set for Python file editing	
Save (`Ctrl+S`)	Saves the current script	
Save as… (`Ctrl+Shift+S`)	Saves the current script as a new file	
Run script (`Ctrl+Shift+E`)	Executes the whole script in the Interactive console (this creates a byte-compiled file with the extension `.pyc`)	Toolbar & Contextual menu
Context help (`F1`)	Attempts to display help on the selected string (class, method, object,…) in its corresponding API documentation	Contextual menu
Run selected (`Ctrl+E`)	Executes in the Interactive console the lines selected in the script	Toolbar & Contextual menu
Cut (`Ctrl+X`)	Cuts selected text to the clipboard	
Copy (`Ctrl+C`)	Copies selected text to the clipboard	
Paste (`Ctrl+V`)	Pastes a cut or copied text	
Find text	Allows to search and replace a text in the script. • Use the default Desktop Environment shortcut to find next/previous (`Ctrl+G` and `Shift+Ctrl+G`); • Encontrar automaticamente a primeira correspondência ao digitar na caixa de busca; • Defina a string de busca inicial para a seleção ao abrir a busca; • Enable the *Use Regular Expressions* option in the search bar to use regular expressions instead of plain text matching; • Replace the matching text with the text in the replace box. When using a regular expression, you can reference captured groups (e.g. via `\1`) in the replacement text. Use *Replace All* to replace all matches at once. • Pressionando `Esc` fecha a barra de busca.	
Toggle comment (`Ctrl+:`)	Comments out or uncomment selected lines, by adding or removing # character (potentially followed by a space) at the start of the lines. If the selection contains both comments and actual commands, adds # at the start of each non empty line. Pressing again the button reverts to the initial state.	
Reformat code (`Ctrl+Alt+F`)	Allows to manually apply various formatting rules (sort imports, indentation, line length,…) to the code, following *user-defined settings*. This may require installation of additional Python modules.	
Object inspector…	Shows and hides a dedicated browser with a tree structure for classes and functions available in the script. Click an item fo a quick access to its definition. The tool requires an activation from the *Python settings - Run and Debug*.	

continua na próxima página

Tabela 25.1 – continuação da página anterior

Tool name	Function	Location
Hide editor	Hides the Python code editor block. To make it visible again, press Show editor button from the interactive console toolbar.	Contextual menu
Check syntax (`Ctrl+4`)	Browses the code and reports syntax errors, such as missing parenthesis, colons, wrong indentation,…	
Undo (`Ctrl+Z`)	Undoes the latest action	
Redo (`Ctrl+Shift+Z`)	Reverts undone actions to a more recent	
Select all (`Ctrl+A`)	Selects the whole script	
Share on GitHub	Shares the script as a *Secret Gist* or *Public Gist* on GitHub, provided a *GitHub access token*.	
Options…	Opens the *Python settings* dialog.	

CAPÍTULO 26

Apêndices

26.1 Apêndice C: Formatos de arquivo QGIS

26.1.1 QGS/QGZ - O Formato de Arquivo de Projeto QGIS

O formato **QGS** é um formato XML para armazenar projetos QGIS. O formato **QGZ** é um arquivo compactado (zip) que contém um arquivo QGS e um arquivo QGD. O arquivo **QGD** é o banco de dados sqlite associado do projeto qgis que contém dados auxiliares para o projeto. Se não houver dados auxiliares, o arquivo QGD estará vazio.

Um arquivo QGIS contém tudo o que é necessário para armazenar um projeto QGIS, incluindo:

- título do projeto
- SRC do projeto
- a árvore de camadas
- configurações de encaixe
- relações
- a extensão da tela do mapa
- modelos do projeto
- legend
- docas de visualização de mapa (2D e 3D)
- as camadas com links para os conjuntos de dados subjacentes (fontes de dados) e outras propriedades da camada, incluindo extensão, SRS, junções, estilos, renderizador, modo de mescla, opacidade e muito mais.
- propriedades do projeto

As figuras abaixo mostram as marcações de nível superior em um arquivo QGS e a marca expandida “Camadas do Projeto”.

Figura26.1: As marcações de nível superior em um arquivo QGS

```
−<projectlayers>
 −<maplayer styleCategories="AllStyleCategories" readOnly="0" autoRefreshTime="0" autoRefreshEnabled="0" refreshOnNotifyEnabled="0" maxScale="0"
  geometry="Polygon" labelsEnabled="0" type="vector" simplifyDrawingHints="1" hasScaleBasedVisibilityFlag="0" simplifyDrawingTol="1"
  simplifyMaxScale="1" minScale="1e+8" simplifyAlgorithm="0" simplifyLocal="1" refreshOnNotifyMessage="">
  +<extent></extent>
   <id>watersheds_b62efa19_8809_4406_b6ec_2951ac4c94c5</id>
  −<datasource>
     ./QGIS-Training-Data-2.0/exercise_data/processing/generalize/watersheds.shp
   </datasource>
  +<keywordList></keywordList>
   <layername>watersheds</layername>
  +<srs></srs>
  +<resourceMetadata></resourceMetadata>
   <provider encoding="UTF-8">ogr</provider>
   <vectorjoins/>
   <layerDependencies/>
   <dataDependencies/>
   <legend type="default-vector"/>
   <expressionfields/>
  +<map-layer-style-manager current="default"></map-layer-style-manager>
   <auxiliaryLayer/>
  +<flags></flags>
  +<renderer-v2 symbollevels="0" enableorderby="0" type="singleSymbol" forceraster="0"></renderer-v2>
  +<customproperties></customproperties>
   <blendMode>0</blendMode>
   <featureBlendMode>0</featureBlendMode>
   <layerOpacity>1</layerOpacity>
  +<SingleCategoryDiagramRenderer diagramType="Histogram" attributeLegend="1"></SingleCategoryDiagramRenderer>
  +<DiagramLayerSettings priority="0" linePlacementFlags="18" dist="0" showAll="1" placement="1" obstacle="0" zIndex="0"></DiagramLayerSettings>
  +<geometryOptions removeDuplicateNodes="0" geometryPrecision="0"></geometryOptions>
  +<fieldConfiguration></fieldConfiguration>
  +<aliases></aliases>
   <excludeAttributesWMS/>
   <excludeAttributesWFS/>
  +<defaults></defaults>
  +<constraints></constraints>
  +<constraintExpressions></constraintExpressions>
   <expressionfields/>
  +<attributeactions></attributeactions>
  +<attributetableconfig actionWidgetStyle="dropDown" sortExpression="" sortOrder="0"></attributetableconfig>
  +<conditionalstyles></conditionalstyles>
   <editform tolerant="1"/>
   <editforminit/>
   <editforminitcodesource>0</editforminitcodesource>
   <editforminitfilepath/>
  +<editforminitcode></editforminitcode>
   <featformsuppress>0</featformsuppress>
   <editorlayout>generatedlayout</editorlayout>
  +<editable></editable>
  +<labelOnTop></labelOnTop>
   <widgets/>
   <previewExpression>ID</previewExpression>
   <mapTip/>
 </maplayer>
</projectlayers>
```


Figura26.2: O nível superior expandido marca as camadas do projeto de um arquivo QGS

26.1.2 QLR - O arquivo de definição de camada QGIS

Um arquivo de Definição de Camada (QLR) é um arquivo XML que contém um ponteiro para a fonte de dados da camada, além das informações de estilo QGIS para a camada.

O caso de uso desse arquivo é simples: ter um único arquivo para abrir uma fonte de dados e trazer todas as informações de estilo relacionadas. Os arquivos QLR também permitem mascarar a fonte de dados subjacente em um arquivo fácil de abrir.

Um exemplo de uso de QLR é para abrir camadas do MS SQL. Em vez de ter que ir para a caixa de diálogo de conexão do MS SQL, conectar, selecionar, carregar e finalmente aplicar o esyilo, você pode simplesmente adicionar um arquivo .qlr que aponte para a camada correta do MS SQL com todo o estilo necessário incluído.

No futuro, um arquivo .qlr pode conter uma referência a mais de uma camada.

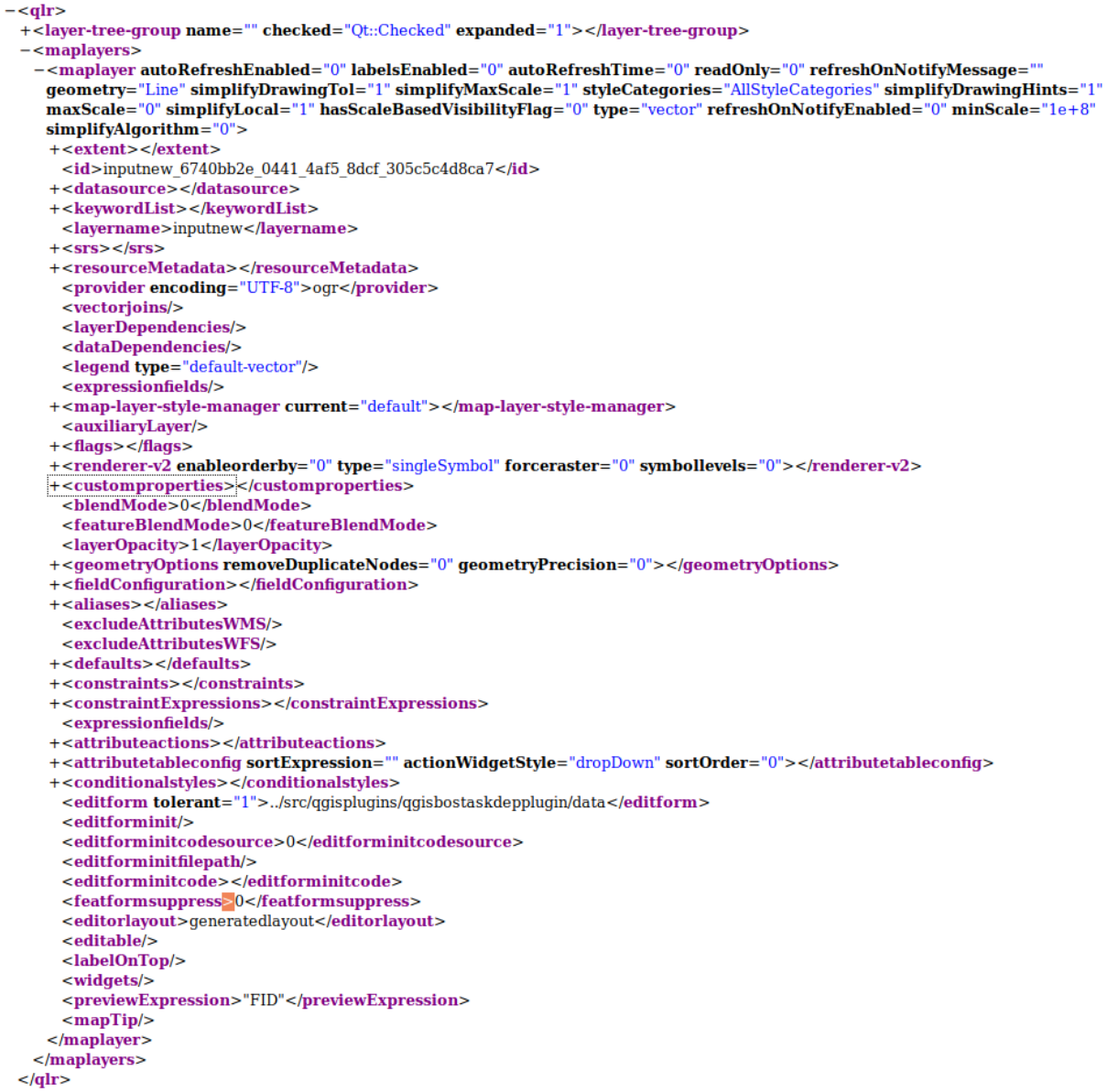


Figura26.3: As marcações de nível superior de um arquivo QLR

26.1.3 QML - O Formato de Arquivo de Estilo QGIS

QML é um formato XML para armazenar o estilo da camada.

Um arquivo QML contém todas as informações que o QGIS pode manipular para a renderização de geometrias de feições, incluindo definições de símbolos, tamanhos e rotações, inclusão de rótulos, opacidade e modo de mescla e muito mais.

A figura abaixo mostra as marcações de nível superior de um arquivo QML (com apenas `renderizador_v2` e sua marcação `símbolo` expandida).

```
−<qgis version="3.4.13-Madeira" styleCategories="AllStyleCategories" readOnly="0" maxScale="0"
 labelsEnabled="0" simplifyDrawingHints="1" hasScaleBasedVisibilityFlag="0" simplifyDrawingTol="1"
 simplifyMaxScale="1" minScale="1e+8" simplifyAlgorithm="0" simplifyLocal="1">
 +<flags></flags>
 −<renderer-v2 symbollevels="0" enableorderby="0" type="singleSymbol" forceraster="0">
   −<symbols>
     +<symbol clip_to_extent="1" name="0" alpha="1" type="fill" force_rhr="0"></symbol>
    </symbols>
    <rotation/>
    <sizescale/>
  </renderer-v2>
 +<customproperties></customproperties>
  <blendMode>0</blendMode>
  <featureBlendMode>0</featureBlendMode>
  <layerOpacity>1</layerOpacity>
 +<SingleCategoryDiagramRenderer diagramType="Histogram" attributeLegend="1">
  </SingleCategoryDiagramRenderer>
 +<DiagramLayerSettings priority="0" linePlacementFlags="18" dist="0" showAll="1" placement="1"
  obstacle="0" zIndex="0">
  </DiagramLayerSettings>
 +<geometryOptions removeDuplicateNodes="0" geometryPrecision="0"></geometryOptions>
 +<fieldConfiguration></fieldConfiguration>
 +<aliases></aliases>
  <excludeAttributesWMS/>
  <excludeAttributesWFS/>
 +<defaults></defaults>
 +<constraints></constraints>
 +<constraintExpressions></constraintExpressions>
  <expressionfields/>
 +<attributeactions></attributeactions>
 +<attributetableconfig actionWidgetStyle="dropDown" sortExpression="" sortOrder="0">
  </attributetableconfig>
 +<conditionalstyles></conditionalstyles>
  <editform tolerant="1"/>
  <editforminit/>
  <editforminitcodesource>0</editforminitcodesource>
  <editforminitfilepath/>
 +<editforminitcode></editforminitcode>
  <featformsuppress>0</featformsuppress>
  <editorlayout>generatedlayout</editorlayout>
 +<editable></editable>
 +<labelOnTop></labelOnTop>
  <widgets/>
  <previewExpression>ID</previewExpression>
  <mapTip/>
  <layerGeometryType>2</layerGeometryType>
</qgis>
```

Figura26.4: As marcações de nível superior de um arquivo QML (somente a marcação renderizador_v2 com sua marcação de símbolo é expandida)

26.2 Apêndice D: sintaxe de script QGIS R

Contributed by Matteo Ghetta - funded by Scuola Superiore Sant'Anna

Escrever scripts R scripts em Processamento é um pouco complicado em função da sintaxe específica.

Um script de Processamento R começa com a definição de suas **Entradas** e **Saídas**, cada uma precedida de caracteres tralha duplos (##).

Pode-se especificar o grupo para colocar o algoritmo antes das entradas. Se o grupo já existe, o algoritmo será adicionado a ele e, se não, o grupo será criado. No exemplo abaixo, o nome do grupo é "Meu grupo":

```
##Meu Grupo=grupo
```

26.2.1 Entradas

Todos os dados de entrada e parâmetros devem ser especificados. Existem vários tipos de entradas:

- vetor: `##Camada = vetor`
- campo vetorial: ##F = Camada de campo (onde *Layer* é o nome de uma camada vetorial de entrada à qual o campo pertence)
- raster: `##r = raster`
- tabela: `##t = tabela`
- número: `##Num = número`
- string: `##Str = string`
- Booleano: ##Bol = booleano
- elementos em um menu dropdown. Os items devem ser separados com ponto e vírgula `;`: `##tipo=pontos;linhas;pontos+linhas`

26.2.2 Saídas

Quanto às entradas, cada saída tem de ser definida no início do script:

- vetor: `##saída= vetor de saída`
- raster: `##saída= saída raster`
- tabela: `##saída= tabela de saída`
- gráfico: `##saída_gráfico_para_html` (##showplots em versões anteriores)
- Para mostrar a saída do R no *Visualizador de Resultados*, coloque > na frente do comando cuja saída você gostaria de mostrar.

26.2.3 Sumário de Sintaxe para scripts QGIS R

Vários tipos de parâmetros de entrada e saída são oferecidos.

Tipos de parâmetros de entrada

Parâmetros	Exemplo de sintaxe	Retornando objetos
vetor	Camada = vetor	sf objeto (ou objeto ObjetoDeQuadroDeDadosEspaciais, se ##carrega_vetor_usando_rgdal for especificado)
vetor de pontos	Camada = vetor de pontos	sf objeto (ou objeto ObjetoDeQuadroDeDadosEspaciais, se ##carrega_vetor_usando_rgdal for especificado)
vetor de linha	Camada = vetor de linha	sf objeto (ou objeto ObjetoDeQuadroDeDadosEspaciais, se ##carrega_vetor_usando_rgdal for especificado)
vetor de poligono	Camada = polígono de vetor	objeto sf (ou EspacialPolígonosQuadroDeDados objeto, se ##carrega_vetor_usando_rgdal é usado)
vetores multiplos	Camada = vetores múltiplos	objeto sf (ou QuadroDeDadosEspaciais objetos se ##carrega_vetor_usando_rgdal for especificado)
Tabela	Camada = tabela	quadro de conversão de csv, objeto padrão da função “read.csv”
campo	Campo = Camada Campo	nome do campo selecionado, e.g. “Area”
raster	Camada = imagem	Objeto RasterBrick, padrão do pacote de objetos “raster”
múltiplas imagens	Camada = Múltiplas imagens	Objeto RasterBrick, padrão do pacote de objetos “raster”
número	N = número	número escolhido inteiro ou flutuante
texto	S = texto	texto adicionado na caixa
texto longo	LS= texto longo	texto adicionado na caixa, pode ser maior que o texto normal
seleção	S = primeira seleção; segunda; terceira	texto do item selecionado escolhido no menu suspenso.
crs	C = crs	string do SRC resultante escolhido, no formato: `"EPSG: 4326"`
Extensão	E = extensão	Extensão do pacote de objeto “raster”, você pode extrair valores como “E@xmin”
ponto	P = ponto	Quando clicado no mapa, você terá as coordenadas do ponto
arquivo	F = arquivo	caminho do arquivo escolhido, e.g “/home/matteo/file.txt”
folder	F = pasta	caminho do arquivo escolhido, e.g “/home/matteo/Downloads”

Um parâmetro poder ser **OPCIONAL**, o que significa que pode ser ignorado.

Para definir uma entrada como opcional, adicione a string `opcional` **antes** da entrada, como por exemplo:

```
##Layer = vector
##Field1 = Field Layer
##Field2 = optional Field Layer
```

Tipos de parâmetros de saída

Parâmetros	Exemplo de sintaxe
vetor	Saída = vetor de saída
raster	Saída = imagem de saída
Tabela	Saída = tabela de saída
arquivo	Saída = arquivo de saída

Nota

Você pode salvar gráficos como `png` do *Visualizador de Resultado em Processamento*, ou salvar o gráfico diretamente da interface do algoritmo.

Cuerpo de lineas de codigo

O corpo do script segue a sintaxe R e o painel **Log** pode ajudar se houver algo errado com o seu script

Lembre que você precisa carregar todas as bibliotecas adicionais no script:

```
library(sp)
```

26.2.4 Exemplos

Exemplo com saída em vetor

Vamos pegar um algoritmo da coleção online que cria pontos aleatórios a partir da extensão de uma camada de entrada:

```
##Point pattern analysis=group
##Layer=vector polygon
##Size=number 10
##Output=output vector
library(sp)
spatpoly = as(Layer, "Spatial")
pts=spsample(spatpoly,Size,type="random")
spdf=SpatialPointsDataFrame(pts, as.data.frame(pts))
Output=st_as_sf(spdf)
```

Explicação (por linha no script):

1. `Análise de padrão pontual` é o grupo do algoritmo
2. `Camada` é a camada de entrada **vetorial**
3. `Tamanho` é um parâmetro **numérico** com um valor padrão de 10
4. `Saída` é a camada **vetorial** que será criada pelo algoritmo
5. `library(sp)` loads the **sp** library
6. spatpoly = como (Camada , “Espacial”)`` traduz para um objeto sp
7. Chame a função `spamostra` da biblioteca `sp` e execute-a usando a entrada definida acima (`Camada` e ``Tamanho`)
8. Crie um *EspacialPontosQuadroDeDados” usando o objeto `EspacialPontosQuadroDeDados` função
9. Crie a camada vetorial de saída usando a função `st_as_sf`

É isso! Agora você só precisa rodar o algoritmo com uma camada vetorial que você tenha na Legenda QGIS, escolhendo o número de ponto aleatório. A camada resultante será adicionada ao seu mapa.

Exemplo com saída raster

O script a seguir executará krigagem ordinária básica para criar um mapa raster de valores interpolados de um campo especificado da camada vetorial de ponto de entrada usando a função `autoKrige` do pacote R `automap`. Ele primeiro calculará o modelo de krigagem e, em seguida, criará um raster. O raster é criado com a função `raster` do pacote raster R:

```
##Basic statistics=group
##Layer=vector point
##Field=Field Layer
##Output=output raster
##load_vector_using_rgdal
require("automap")
```

(continua na próxima página)

(continuação da página anterior)

```
require("sp")
require("raster")
table=as.data.frame(Layer)
coordinates(table)= ~coords.x1+coords.x2
c = Layer[[Field]]
kriging_result = autoKrige(c~1, table)
prediction = raster(kriging_result$krige_output)
Output<-prediction
```

Usando `##carregar_vetor_usando_rgdal`, a camada vetorial de entrada será disponibilizada como um `EspacialQuadroDeDados`, então evitamos ter que traduzi-lo de um objeto `sf`.

Exemplo com saída em tabela

Vamos editar o algoritmo do `Sumário de Estatísticas` para que a saída seja um arquivo de dados em tabela (csv).

O corpo do script é o seguinte:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##Stat=Output table
Summary_statistics<-data.frame(rbind(
    sum(Layer[[Field]]),
    length(Layer[[Field]]),
    length(unique(Layer[[Field]])),
    min(Layer[[Field]]),
    max(Layer[[Field]]),
    max(Layer[[Field]])-min(Layer[[Field]]),
    mean(Layer[[Field]]),
    median(Layer[[Field]]),
    sd(Layer[[Field]]))),
  row.names=c("Sum:","Count:","Unique values:","Minimum value:","Maximum value:",
↪"Range:","Mean value:","Median value:","Standard deviation:"))
colnames(Summary_statistics)<-c(Field)
Stat<-Summary_statistics
```

A terceira linha especifica o **Campo vetorial** na entrada e a quarta linha informa ao algoritmo que a saída deve ser uma tabela.

A última linha pegará o objeto `Stat` criado no script e o converterá em uma tabela `csv`.

Exemplo com saída do console

Você pode usar o exemplo anterior e ao invés de criar uma tabela, pode imprimir o resultado no **Visualizador de Resultado**:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
Summary_statistics<-data.frame(rbind(
sum(Layer[[Field]]),
length(Layer[[Field]]),
length(unique(Layer[[Field]])),
min(Layer[[Field]]),
max(Layer[[Field]]),
max(Layer[[Field]])-min(Layer[[Field]]),
mean(Layer[[Field]]),
```

(continua na próxima página)

(continuação da página anterior)

```
median(Layer[[Field]]),
sd(Layer[[Field]])),row.names=c("Sum:","Count:","Unique values:","Minimum value:",
↪"Maximum value:","Range:","Mean value:","Median value:","Standard deviation:"))
colnames(Summary_statistics)<-c(Field)
>Summary_statistics
```

O script é exatamente o mesmo que foi usado acima exceto por duas edições:

1. nenhuma saída especificada (a quarta linha foi removida)
2. a última linha começa com >, informando ao Processamento que deve deixar o objeto disponível no visualizador de resultado

Exemplo com gráfico

Para criar plotagens, você deve usar o parâmetro `##saída_impressão_para_html` como no script a seguir:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##output_plots_to_html
####output_plots_to_html
qqnorm(Layer[[Field]])
qqline(Layer[[Field]])
```

O roteiro usa um campo (`Campo`) de uma camada vetorial (`Camada`) como entrada e cria um *QQ Plot* (para testar a normalidade da distribuição).

O gráfico é adicionado automaticamente ao Processando *Visualizador de Resultados*.

26.3 Apêndice E: Conexões de Rede do Aplicativo QGIS

Esta é uma lista das conexões de rede pré-configuradas/automatizadas que o QGIS faz. Algumas das conexões são iniciadas pelo usuário porque exigem uma ação do usuário antes que ocorram, outras acontecem automaticamente.

Nome	Finalidade	Modo UX	Servidor	Informações enviadas ao servidor	Informações armazenadas no servidor
qgis.org					
Ajuda da API Python	Documentação do navegador PyQGIS	Iniciado pelo usuário	https://qgis.org/pyqgis/3.44/search	IP, versão QGIS, OS	IP no log do servidor
version.qgis.					
Verificação de nova versão	Notifique o usuário sobre a disponibilidade de novas versões do QGIS	Automático	https://version.qgis.org	IP, versão QGIS, OS	IP no log do servidor
feed.qgis.					
Alimentação QGIS	Recuperar notícias QGIS do feed	Automático	https://feed.qgis.org	IP, versão QGIS, código de idioma, carimbo de data/hora do último download, sistema operacional	IP no log do servidor; A versão do QGIS, o sistema operacional e o IP são agregados e usados para coletar algumas estatísticas
plugins.qgis.o					
Verifique se há atualizações de plugins	Notifique o usuário sobre atualizações de plugins	iniciado pelo usuário/aut (configurável)	https://plugins.qgis.org	IP, versão QGIS, OS	IP no log do servidor
Lista de plug-ins	Recupere a lista de plugins	Iniciado pelo usuário	https://plugins.qgis.org	IP, versão QGIS, OS	IP no log do servidor
Instalação do plug-in	Baixe e instale um pacote de plugins	Iniciado pelo usuário	https://plugins.qgis.org	IP, versão QGIS, OS	Aumente o contador de downloads de plugins em um
Estilos	Listar estilos contribuídos pelo usuário	Iniciado pelo usuário	https://hub.qgis.org/styles	IP, versão QGIS, OS	Aumente o contador de downloads em um
terceiro					
Dados do terreno	Produza um DEM para vistas 3D	Iniciado pelo usuário	https://s3.amazonaws.com/elevation-tiles-pr terrarium/{z}/{x	IP, versão QGIS, OS	veja Amazon TOS
Geocodificador do Google Maps	Serviços de geocodificação	Iniciado pelo usuário	https://maps.googleapis.com/maps/api/geocode/json	IP, versão QGIS, OS	Veja os TOS da API do Google Maps
Geocodificador Nominatim	Serviços de geocodificação	Iniciado pelo usuário	https://nominatim.qgis.org/ui/search.html	IP, versão QGIS, OS	
Grade geodésica	Adicionar uma nova grade PROJ	Iniciado pelo usuário	https://cdn.proj.org	IP, versão PROJ	Os logs de acesso são excluídos permanentemente após um dia

CAPÍTULO 27

Referências Bibliográficas e Web

GDAL-SOFTWARE-SUITE. Geospatial data abstraction library. https://gdal.org/en/latest, 2013.

GRASS-PROJECT. Geographic resource analysis support system. https://grass.osgeo.org, 2013.

NETELER, M., AND MITASOVA, H. Open source gis: A grass gis approach, 2008.

OGR-SOFTWARE-SUITE. Geospatial data abstraction library. https://gdal.org/en/latest, 2013.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.1.1) implementation specification. https://portal.ogc.org/files/?artifact_id=1081&version=1&format=pdf, 2002.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.3.0) implementation specification. https://portal.ogc.org/files/?artifact_id=14416&format=pdf, 2004.

POSTGIS-PROJECT. Spatial support for postgresql. http://www.refractions.net/products/postgis/, 2013.