

APÊNDICE B

WORKFLOW DE TRANSPORTES

O objetivo do processo de correção de transportes é identificar e corrigir erros comuns em dados geoespaciais por meio de 16 rotinas para gerar as áreas com delimitadores e centroide após as correções topológicas, garantindo a execução correta.

As etapas iniciais do fluxo visam identificar geometrias inválidas, como vértices não compartilhados em interseções, sobreposições de feições e geometrias com topologia incorreta, entre outros erros que podem surgir devido a problemas na coleta, digitalização ou processamento de dados geoespaciais. A correção desses erros é essencial para evitar falhas na execução das etapas seguintes do processo de identificação de erros, tornando a etapa de identificação de geometrias inválidas uma prioridade na correção.

Os comandos em JSON funcionam como instruções de como o algoritmo provedor deve se comportar definindo seus parâmetros.

1- IDENTIFICAÇÃO DE GEOMETRIAS INVÁLIDAS:

a) Descrição: Este processo identifica geometrias inválidas nas camadas especificadas.

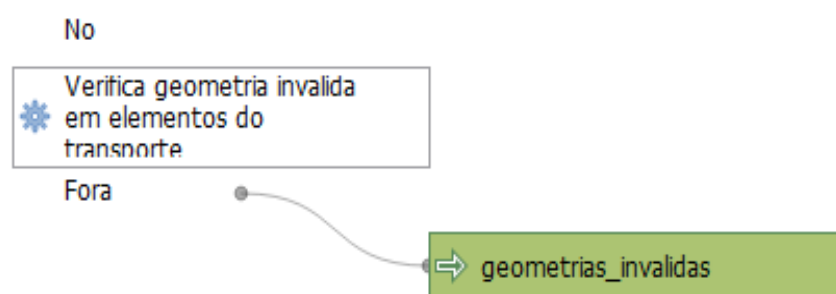
b) Arquivo: identifica_geometrias_invalidas_transportes_carta_orto.model3.

c) Algoritmos:

dsgtools: batchrunalgorithm,

dsgtools: identifyandfixinvalidgeometries

d) Composição do Modelo:



O modelo é composto por uma etapa:

Verifica geometria inválida: Identifica geometrias inválidas em todas as camadas de entrada, utilizando o algoritmo "*identifyandfixinvalidgeometries*".

e) Parâmetros:

Execução de algoritmos em lote

Properties Comments

Description: Verifica geometria invalida em elementos do transporte

Nomes das camadas de entrada separados por vírgula
123 infra_ferrovia_l,infra_mobilidade_urbana_l,infra_travessia_hidroviaria_l,infra_via_deslocamento_l

Nome do algoritmo com seu provedor
123 dsgtools:identifyandfixinvalidgeometries

Nome da chave de entrada
123 INPUT

Dicionário de parâmetros em JSON
123 {
 "SELECTED": false,
 "IGNORE_CLOSED": false,
 "TYPE": false,
 "FLAGS": "TEMPORARY_OUTPUT"
}

Nome dos parâmetros de saída da camada [opcional]
123 FLAGS

Execução de saída em lote
123 geometrias_invalidas

Dependencies
0 dependências selecionadas

OK Cancel Help

Camadas de entrada:

Tipo de Geometria	Camadas de Entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

Parâmetros em JSON: definem o comportamento do algoritmo provedor durante sua execução.

Verifica geometria invalida em elementos do transporte:

```
{  
"SELECTED": false,  
"IGNORE_CLOSED": false,  
"TYPE": false,  
"FLAGS": "TEMPORARY_OUTPUT"  
}
```

O parâmetro *"SELECTED"* determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como *"true"*, a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

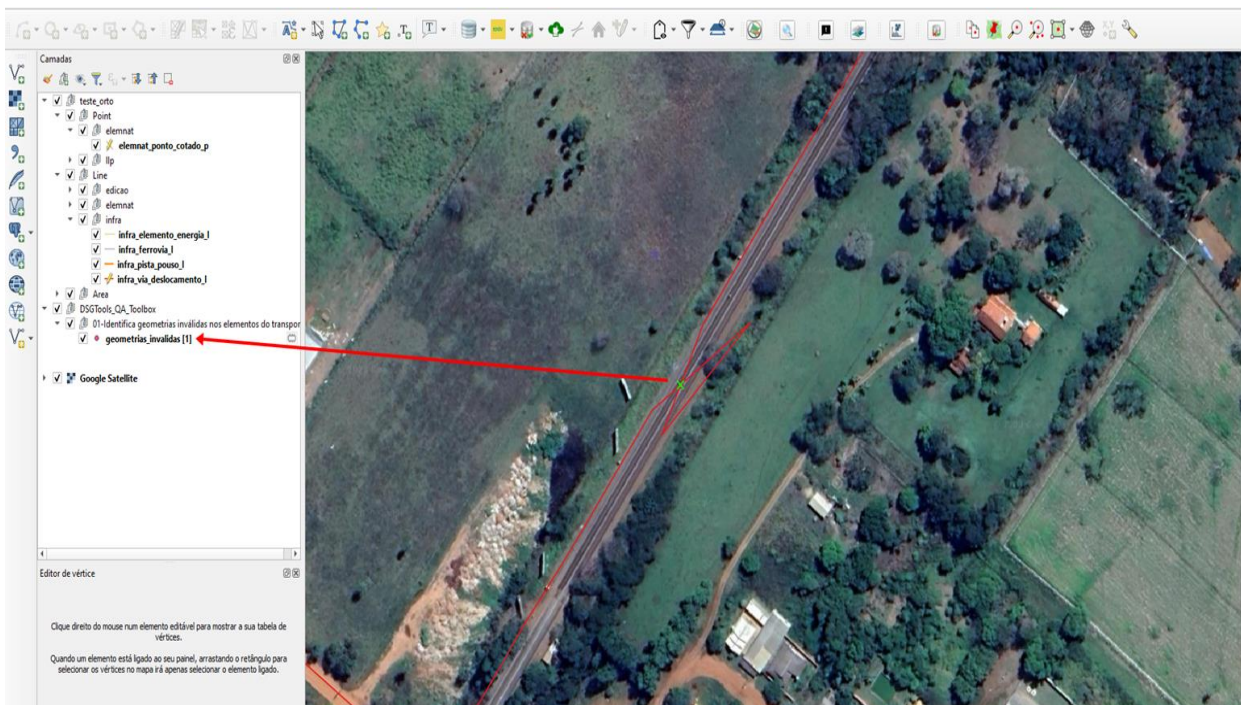
O parâmetro *"IGNORE_CLOSED"* determina se a rotina deve ignorar as geometrias fechadas, como os polígonos. Quando marcado como *"true"*, as geometrias fechadas não são consideradas na validação. Caso contrário, todas as geometrias são avaliadas.

O parâmetro *"TYPE"* é usado para definir o tipo de geometria que será validada. Quando marcado como *"true"*, a validação é realizada apenas nas geometrias do tipo especificado (linha, ponto ou polígono). Caso contrário, todas as geometrias são avaliadas.

O parâmetro *"FLAGS"* é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação. Neste caso, o valor *"TEMPORARY_OUTPUT"* especifica que uma camada temporária será gerada para armazenar informações sobre as geometrias inválidas encontradas durante a validação.

f) Nome da camada de *flags* gerada: geometrias_invalidas.

g) Resultado do processo e exemplo de erro:



2-IDENTIFICA GEOMETRIAS COM MAIS DE UMA PARTE:

a) Descrição: O algoritmo em questão identifica geometrias multipartidas (geométricas composta por múltiplas partes) em diferentes tipos de feições (linha, ponto e polígono) dentro de camadas específicas.

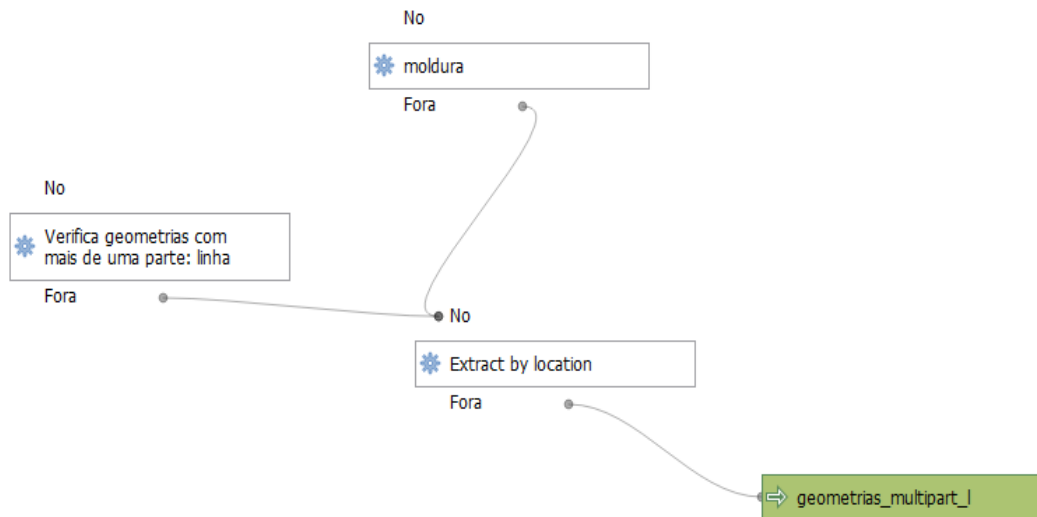
b) Arquivo: identifica_multipart_transportes_carta_orto.model3

c) Algoritmos:

dsgtools:batchrunalgorithm,
dsgtools:identifymultigeometries,
dsgtools:stringcsvtofirstlayerwiththelementsalgorithm

'native:extractbylocation'

d) Composição do Modelo:



O modelo é composto por três etapas:

Verifica geometrias com mais de uma parte na camada de linha: Identifica geometrias com mais de uma parte na camada de linha usando o algoritmo *identifymultigeometries*.

Moldura: Converte uma string csv em uma camada de entidades usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

ExtractByLocation: Extrai as geometrias com mais de uma parte que estão dentro da moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Execução de algoritmos em lote

Properties Comments

Description: Verifica geometrias com mais de uma parte: linha

Nomes das camadas de entrada separados por vírgula
123 infra_ferrovia_l,infra_mobilidade_urbana_l,infra_travessia_hidroviaria_l,infra_via_deslocamento_l

Nome do algoritmo com seu provedor
123 dsgtools:identifymultigeometries

Nome da chave de entrada
123 INPUT

Dicionário de parâmetros em JSON
123 {
 "SELECTED": false,
 "FLAGS": "TEMPORARY_OUTPUT"
}

Nome dos parâmetros de saída da camada [opcional]
123 FLAGS

Execução de saída em lote
[Digite o nome se este for um resultado final]

Dependencies
0 dependências selecionadas

OK Cancel Help


Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

Parâmetros em JSON:

```
{  
  "SELECTED": false,  
  "FLAGS": "TEMPORARY_OUTPUT"  
}
```

Na sequência, o modelo usa o algoritmo do QGIS *"native:extractbylocation"* para extrair apenas as geometrias que intersectam com a camada "moldura", no modo interseção (parâmetro *"PREDICATE"* = [0]). Esse parâmetro significa que somente as geometrias que têm intersecção com a moldura serão extraídas.




 Extrair por localização
 ×

Properties
 Comments

Description


Extrair feições de


 A usar saída de algoritmo


Onde as feições (predicado geométrico)

1 opções selecionadas
 ...

Ao comparar com as feições do


 A usar saída de algoritmo

Extraída (localização)



✖

Dependencies

...

OK

Cancel

Help

O objetivo é restringir as análises apenas às geometrias que estão dentro de uma área de interesse, no caso a moldura.

f) Nome da camada de flags: geometrias_multipart_1

g) Resultado do processo e exemplo de erro:



3- IDENTIFICAR FEIÇÕES DUPLICADAS

a) Descrição: Essa etapa tem como objetivo identificar feições duplicadas em diferentes camadas do projeto, que podem ter sido criadas acidentalmente ou durante o processo de validação. Para isso, o modelo utiliza uma lista negra de atributos (*ATTRIBUTE_BLACKLIST*) que não serão considerados na comparação das feições, a fim de evitar falsos positivos. Essa rotina identifica apenas camadas do tipo linha, e as respectivas feições duplicadas são sinalizadas com flags.

b) Arquivo: identifica_feicoes_duplicadas_transportes_carta_orto.model3

c) Algoritimos:

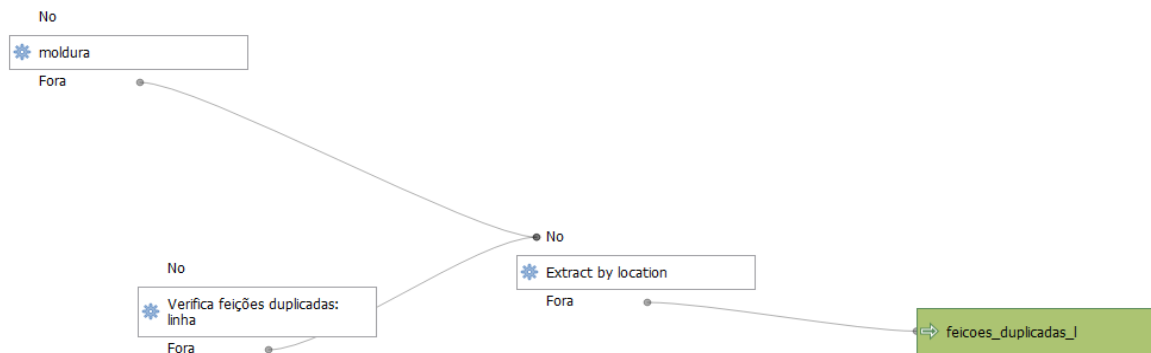
dsgtools:identifyduplicatedfeatures

dsgtools:batchrunalgorithm

native:extractbylocation

dsgtools:stringcsvtofirstlayerwiththelementsalgorithm

d) Composição do Modelo:




O modelo é composto por três etapas:

Moldura: Converte uma string csv em uma camada de entidades usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

Verifica feições duplicadas: Identifica feições duplicadas nas camadas de linha usando o algoritmo *identifyduplicatedfeatures*.

ExtractByLocation: Extrai as feições duplicadas que estão dentro da moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

 Execução de algoritmos em lote

Properties

Comments

Description

Verifica feições duplicadas: linha

Nomes das camadas de entrada separados por vírgula

123

infra_ferrovia_,infra_mobilidade_urbana_,infra_travessia_hidroviaria_,infra_via_deslocamento_

Nome do algoritmo com seu provedor

123

dsgtools:identifyduplicatedfeatures

Nome da chave de entrada

123

INPUT

Dicionário de parâmetros em JSON

123


```
{
  "SELECTED": false,
  "ATTRIBUTE_BLACK_LIST":
["id","texto_edicao","label_x","label_y","justificativa_txt","tamanho_txt","visivel","carta_simbolizacao","simbolizar_carta_mini","simb_rot","rotular_carta_mini","espacamento","tamanho_txt","estilo_fonte","cor","cor_buffer","tamanho_buffer","observacao","length_otf"],
  "IGNORE_VIRTUAL_FIELDS": true,
  "IGNORE_PK_FIELDS": true,
  "FLAGS": "TEMPORARY_OUTPUT"
}
```

Nome dos parâmetros de saída da camada [opcional]

123

FLAGS

Execução de saída em lote



[Digite o nome se este for um resultado final]

Dependencies

0 dependências selecionadas

...

OK

Cancel

Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_1, infra_mobilidade_urbana_1, infra_travessia_hidroviaria_1, infra_via_deslocamento_1

Parâmetros em JSON:

```
{
"SELECTED": false,
"ATTRIBUTE_BLACK_LIST":
["id","texto_edicao","label_x","label_y","justificativa_txt","tamanho_txt","visivel","carta_simboliz
acao","simbolizar_carta_mini","simb_rot","rotular_carta_mini","espacamento","tamanho_txt","esti
lo_fonte","cor","cor_buffer","tamanho_buffer","observacao","length_otf"],
"IGNORE_VIRTUAL_FIELDS": true,
"IGNORE_PK_FIELDS": true,
"FLAGS": "TEMPORARY_OUTPUT"
}
```

f) Nome da camada de flags: feicoes_duplicadas_1

g) Resultado do processo e exemplo de erros:



4- IDENTIFICA VÉRTICE NÃO COMPARTILHADO NAS INTERSECÇÕES:

a) **Descrição:** Este algoritmo identifica vértices não compartilhados nas intersecções entre camadas de linha.

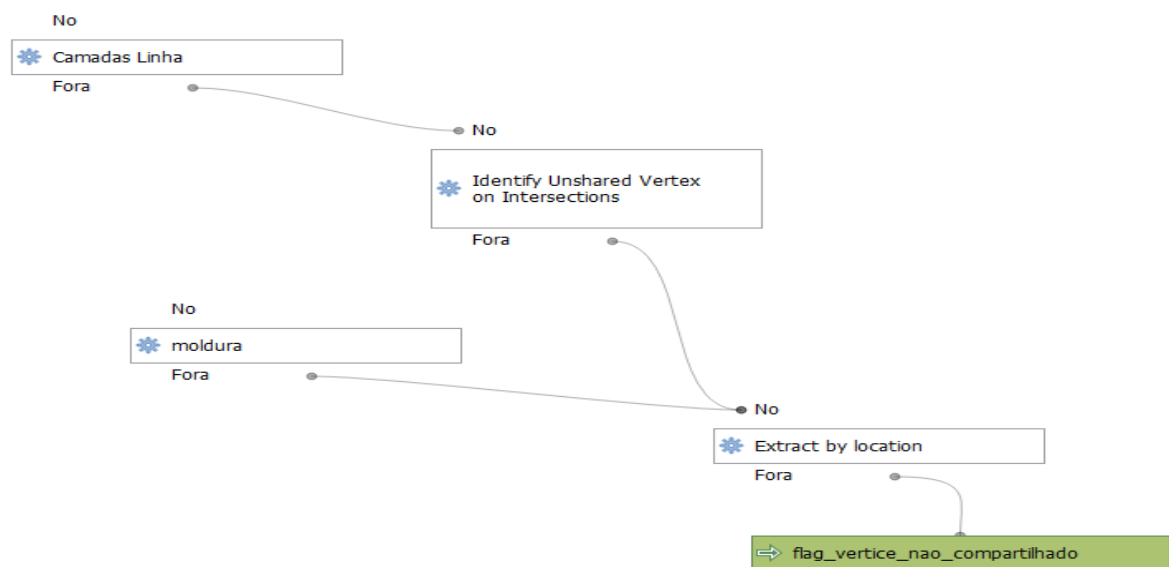
b) **Arquivo:**

identifica_vertice_nao_compartilhado_nas_interseccoes_transportes_carta_orto.model3

c) Algoritimos:

dsgtools:stringcsvtofirstlayerwithelementsalgorithm
dsgtools:stringcsvtolayerlistalgorithm
dsgtools:identifyunsharedvertexonintersectionsalgorithm
native:extractbylocation

d) Composição do Modelo:



O modelo é composto por quatro etapas:

Camadas Linha: Converte uma string csv em uma lista de camadas de linha usando o algoritmo *stringcsvtolayerlistalgorithm*.

Moldura: Converte uma string csv em uma camada de entidades usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

IdentifyUnsharedVertexOnIntersections: Identifica vértices não compartilhados nas intersecções entre as camadas de linha usando o algoritmo *identifyunsharedvertexonintersectionsalgorithm*.

ExtractByLocation: Extrai os vértices não compartilhados que estão dentro da moldura usando o algoritmo *extractbylocation*.

e) **Parâmetros:**

Identificar Vértices Não Compartilhados em Interseções

Properties Comments

Description Identify Unshared Vertex on Intersections

Point Layers [opcional]

123 0 selecionar entradas ...

Camadas do tipo linha [opcional]

A usar saída de algoritmo "Lista múltipla de camadas" do algoritmo "Camadas Linha"

Camadas do tipo polígono [opcional]

123 0 selecionar entradas ...

Processar apenas feições selecionadas

123 Não

Identificar Vértices Não Compartilhados em Interseções Flags

[Digite o nome se este for um resultado final]

Dependencies

0 dependências selecionadas ...

OK Cancel Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_1, infra_mobilidade_urbana_1, infra_travessia_hidroviaria_1, infra_via_deslocamento_1

f) **Nome da camada de flags:** *Flag_vertice_nao_compartilhado*

g) Resultado do processo e exemplos de erros:



5- IDENTIFICAR VÉRTICE NÃO COMPARTILHADO NOS SEGMENTOS COMPARTILHADOS:

a) Descrição: Este algoritmo identifica vértices não compartilhados em segmentos compartilhados entre camadas de linha, e dependendo do seu raio, identifica vértices muito próximos.

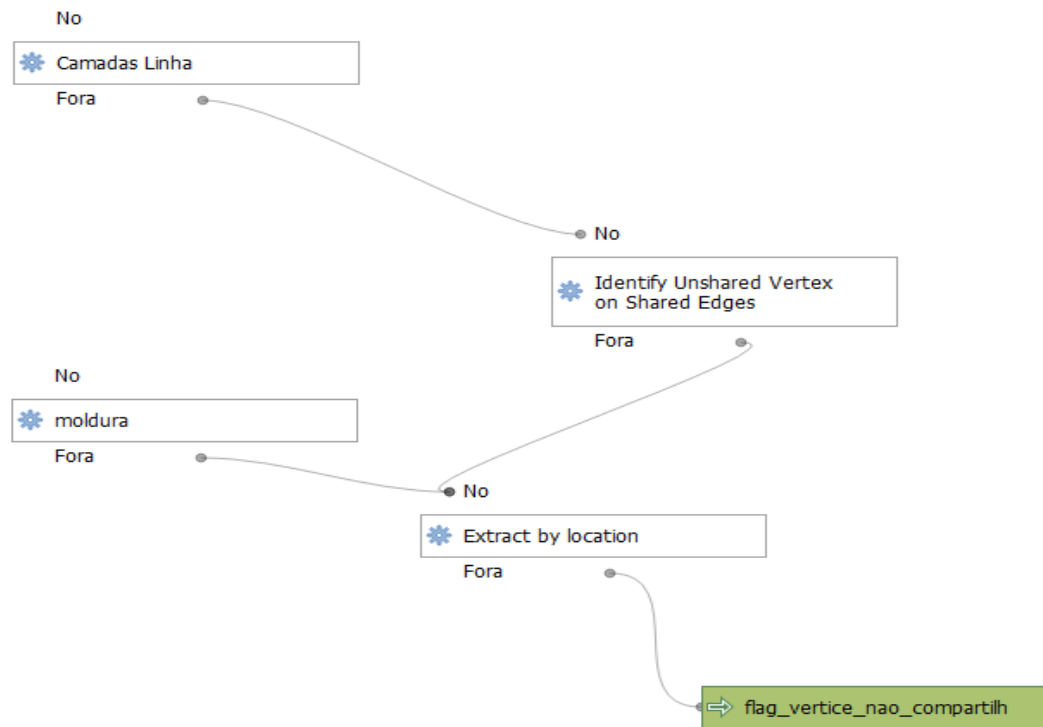
b) Arquivo:

identifica_vertice_nao_compartilhado_nos_segmentos_compartilhados_transportes_carta_orto.mxd

c) Algoritmos:

dsgtools:stringcsvtolayerlistalgorithm
dsgtools:identifyunsharedvertexonintersectionsalgorithm
dsgtools:stringcsvtofirstlayerwiththelementsalgorithm
native:extractbylocation

d) Composição do Modelo:



O modelo é composto por quatro etapas:

Camadas Linha: Converte uma string csv em uma lista de camadas de linha usando o algoritmo *stringcsvtolayerlistalgorithm*.

Moldura: Converte uma string csv em uma camada de entidades usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

IdentifyUnsharedVertexOnSharedEdges: Identifica vértices não compartilhados nos segmentos compartilhados entre as camadas de linha e polígono usando o algoritmo *identifyunsharedvertexonsharededgesalgorithm*.

ExtractByLocation: Extrai os vértices não compartilhados que estão dentro da moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Identificar Vértices Não Compartilhados em Bordas Compartilhadas

Properties Comments

Description Identify Unshared Vertex on Shared Edges

Camadas do tipo linha [opcional]
☐ A usar saída de algoritmo "Lista múltipla de camadas" do algoritmo "Camadas Linha"

Camadas do tipo polígono [opcional]

Processar apenas feições selecionadas

Raio de busca

Identificar Vértices Não Compartilhados em Bordas Compartilhadas Flags
☐ [Digite o nome se este for um resultado final]

Dependencies

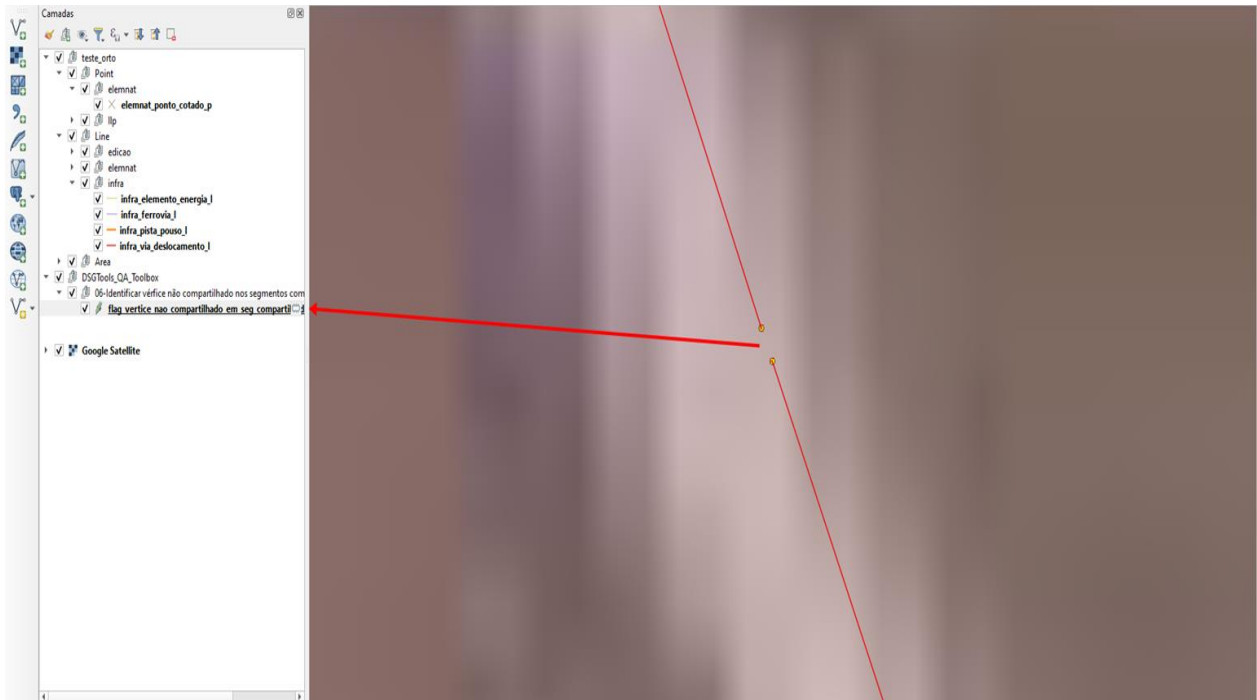
OK Cancel Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

f) Nome da camada de flags: *Flag_vertice_nao_compartilhado_em_seg_compartilhado*.

g) Resultado do Processo e exemplos de erros:



6- IDENTIFICAR VÉRTICE PRÓXIMO DE ARESTA:

a) Descrição: Este algoritmo identifica vértices próximos de arestas em uma lista de camadas de linha.

b) Arquivo: identifica_vertice_proximo_de_aresta_transportes_carta_orto.model3

c) Algoritmo:

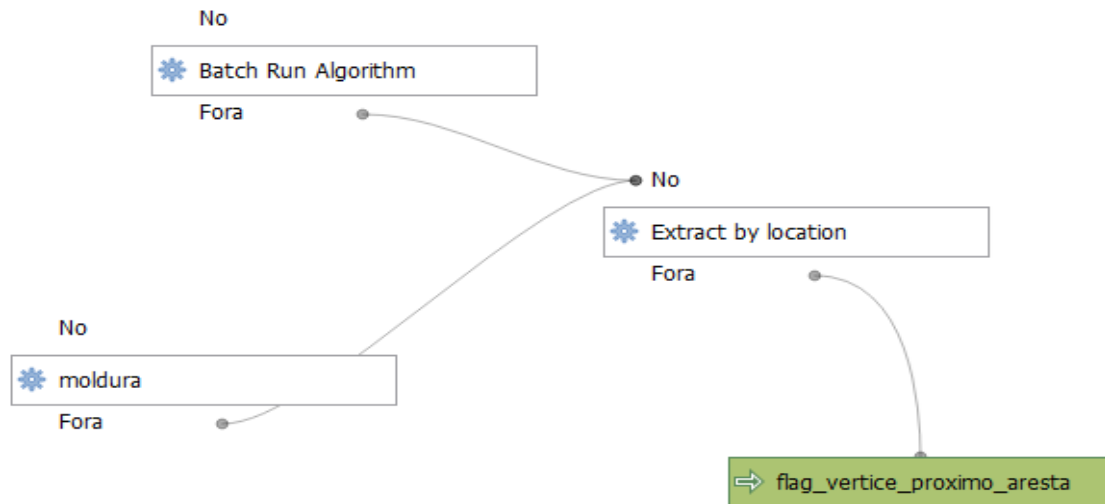
dsgtools:identifyvertexnearedges

dsgtools:batchrunalgorithm

dsgtools:stringcsvtofirstlayerwiththelementsalgorithm

native:extractbylocation

d) Composição do Modelo:



O modelo é composto por três etapas:

BatchRunAlgorithm: Executa o algoritmo *identifyvertexnearedges* em uma lista de camadas de linha.

Moldura: Converte uma string csv em uma camada de entidades usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

ExtractByLocation: Extrai os vértices próximos de arestas que estão dentro da moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Execução de algoritmos em lote

Properties Comments

Description Batch Run Algorithm

Nomes das camadas de entrada separados por vírgula

123 infra_ferrovia_l,infra_mobilidade_urbana_l,infra_travessia_hidroviaria_l,infra_via_deslocamento_l

Nome do algoritmo com seu provedor

123 dsgtools:identifyvertexnearedges

Nome da chave de entrada

123 INPUT

Dicionário de parâmetros em JSON

123 {
 "SELECTED": false,
 "SEARCH_RADIUS": 1e-05,
 "FLAGS": "TEMPORARY_OUTPUT"
}

Nome dos parâmetros de saída da camada [opcional]

123 FLAGS

Execução de saída em lote

➡ [Digite o nome se este for um resultado final]

Dependencies

0 dependências selecionadas

OK Cancel Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

Parâmetros em JSON:

```
{  
"SELECTED": false,  
"SEARCH_RADIUS": 1e-05,  
"FLAGS": "TEMPORARY_OUTPUT" }
```

"SELECTED": false - Esta propriedade é uma chave que tem um valor booleano associado a ela. O valor "false" indica que o item não está selecionado ou marcado.

"SEARCH_RADIUS": 1e-05 - Esta propriedade é uma chave que tem um valor numérico associado a ela. O valor "1e-05" é uma notação científica que representa o número decimal 0.00001. Esse valor pode ser usado como um raio de busca para encontrar objetos dentro de uma determinada distância.

"FLAGS": "TEMPORARY_OUTPUT" - Esta propriedade é uma chave que tem uma string associada a ela. A string "TEMPORARY_OUTPUT" pode ser um sinalizador ou uma opção que é usada para indicar que um determinado resultado ou saída é temporária e não deve ser armazenado permanentemente.

f) Nome da camada de *flags*: *Flag_vertice_nao_compartilhado_em_seg_compartilhado*

g) Resultado do processo e exemplo de erros:



7- IDENTIFICAR GEOMETRIAS COM DENSIDADE INCORRETA DE VÉRTICES

a) Descrição: O objetivo deste algoritmo é realizar a identificação de geometrias que apresentam uma densidade de vértices considerada inadequada. Essa identificação é baseada em uma distância específica dentro de uma tolerância pré-estabelecida. O propósito principal é reduzir a quantidade de vértices presentes nas geometrias, buscando simplificar e otimizar sua representação.

b) Arquivo:

identifica_geometrias_com_densidade_incorreta_de_vertices_transportes_carta_orto.model3

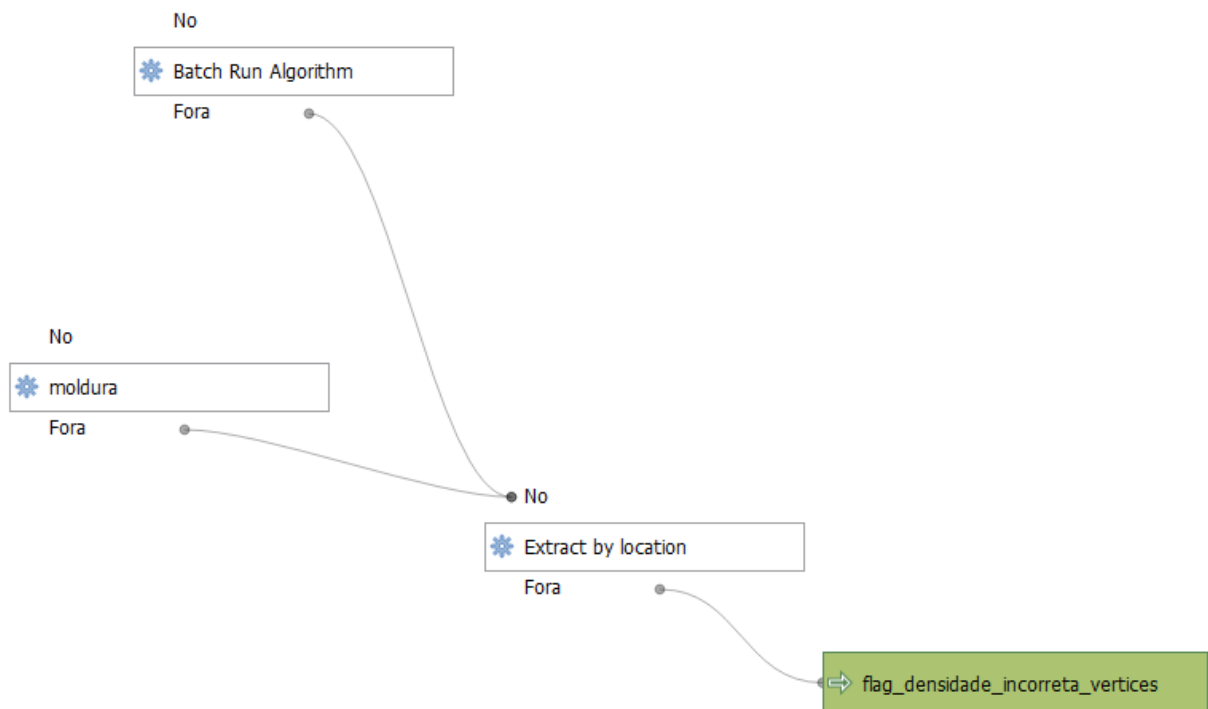
c) Algoritmo:

dsgtools:batchrunalgorithm

dsgtools:stringcsvtofirstlayerwithelementsalgorithm

native:extractbylocation

d) Composição do Modelo:



O modelo é composto por três etapas:

BatchRunAlgorithm: Executa o algoritmo *identifygeometrieswithlargevertexdensityalgorithm* com os parâmetros descritos conforme a seção e.

Moldura: Converte uma *string* csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

ExtractByLocation: Extrai os pontos com densidade incorreta que estão dentro da moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Execução de algoritmos em lote

Properties Comments

Description Batch Run Algorithm

Nomes das camadas de entrada separados por vírgula

123 infra_ferrovia_l,infra_mobilidade_urbana_l,infra_travessia_hidroviaria_l,infra_via_deslocamento_l

Nome do algoritmo com seu provedor

123 dsgtools:identifygeometrieswithlargevertexdensityalgorithm

Nome da chave de entrada

123 INPUT

Dicionário de parâmetros em JSON

123 {
 "SELECTED": false,
 "SEARCH_RADIUS": 1e-05,
 "FLAGS": "TEMPORARY_OUTPUT"
}

Nome dos parâmetros de saída da camada [opcional]

123 FLAGS

Execução de saída em lote

➡ [Digite o nome se este for um resultado final]

Dependencies

0 dependências selecionadas

OK Cancel Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

Parâmetros em JSON:

```
{  
  "SELECTED": false,  
  "SEARCH_RADIUS": 1e-05,  
  "FLAGS": "TEMPORARY_OUTPUT"
```

}

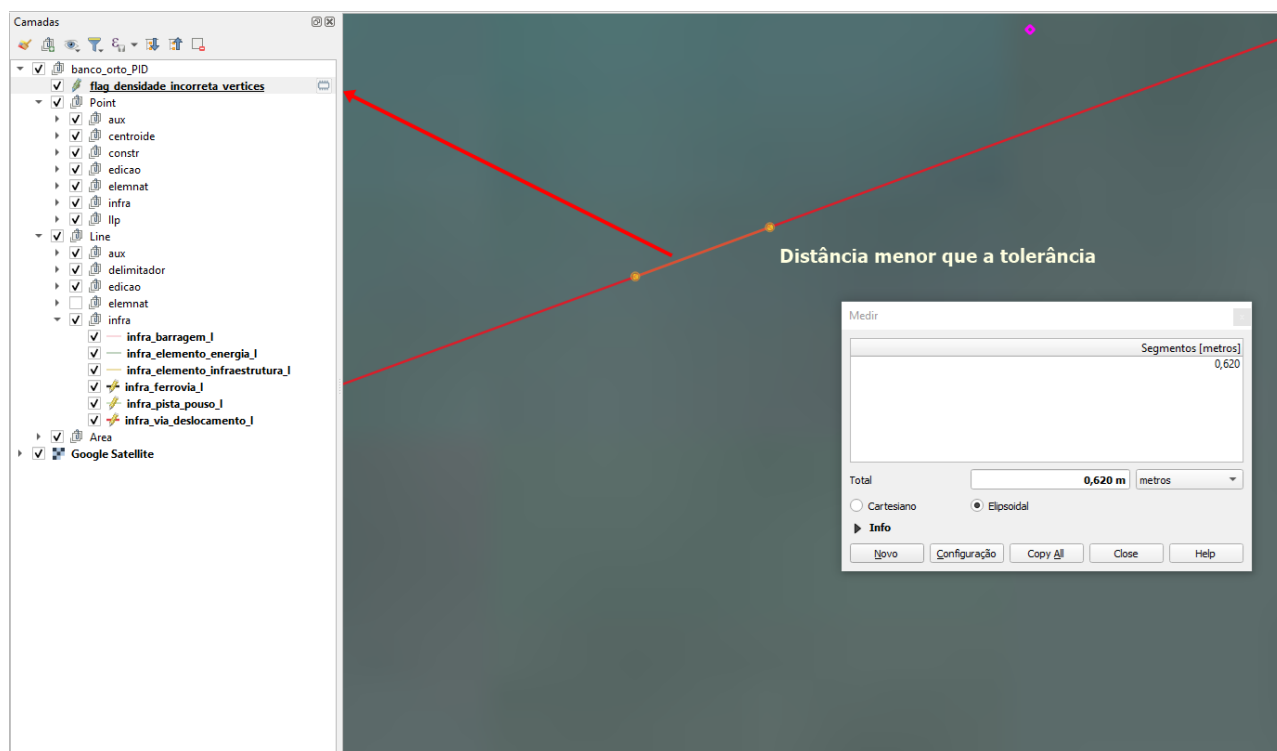
O parâmetro **"SELECTED"** determina se a identificação de erros deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como **"true"**, a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

O parâmetro **"SEARCH RADIUS"**: do algoritmo determina a distância máxima que será percorrida para encontrar vértices próximos um dos outros. No modelo, o valor padrão do raio de busca é definido como 0,00001 graus no sistema de coordenadas geográfico que corresponde a um valor aproximado de 1 metro. No entanto, é importante lembrar que esse valor pode ser ajustado de acordo com as necessidades específicas, considerando a escala e precisão dos dados utilizados no processamento.

O parâmetro **"FLAGS"** é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de onde estão os erros. Neste caso, o valor **"TEMPORARY_OUTPUT"** especifica que uma camada temporária será gerada para armazenar informações sobre a densidade incorreta de vértice.

f) Nome da camada de flags: flag_densidade_incorreta_vertices

g) Resultado do processo e exemplo de erros:



8– IDENTIFICAR ÂNGULO PEQUENOS

a) Descrição: Este algoritmo identifica ângulos pequenos em uma lista de camadas de linha.

b) Arquivo: identifica_angulos_pequenos_transportes_carta_orto.model3

c) Algoritmo:

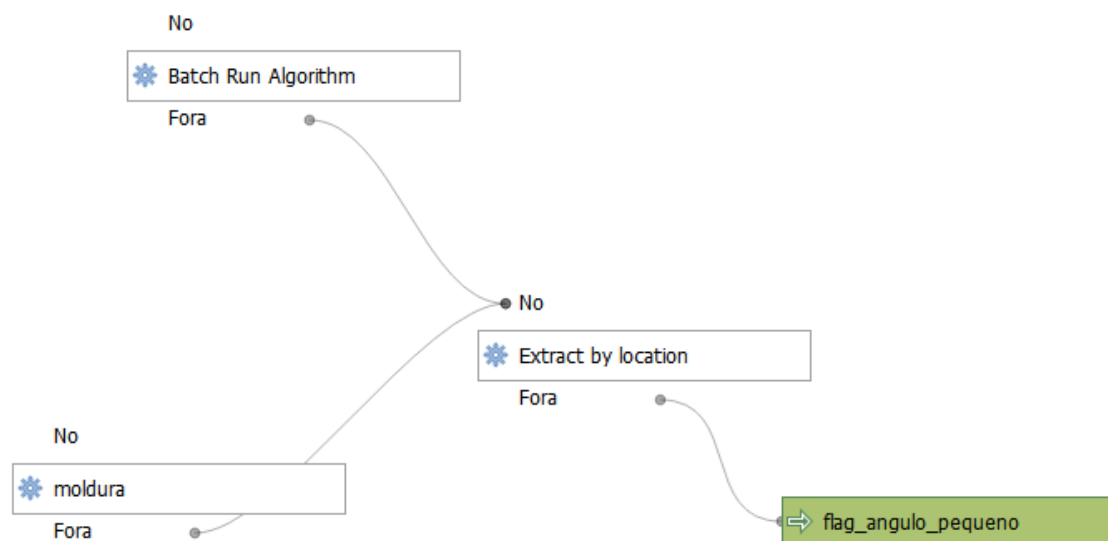
dsgtools:identifyoutofboundsangles

dsgtools:batchrunalgorithm

dsgtools:stringcsvtofirstlayerwithelementsalgorithm

native:extractbylocation

d) Composição do Modelo:



O modelo é composto por três etapas:

BatchRunAlgorithm: Executa o algoritmo *identifyoutofboundsangles* com os parâmetros selecionados abaixo.

Moldura: Converte uma *string* csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

ExtractByLocation: Extrai os ângulos pequenos que estão dentro da moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Execução de algoritmos em lote

Properties Comments

Description Batch Run Algorithm

Nomes das camadas de entrada separados por vírgula

123 infra_ferrovia_l,infra_mobilidade_urbana_l,infra_travessia_hidroviaria_l,infra_via_deslocamento_l

Nome do algoritmo com seu provedor

123 dsgtools:identifyoutofboundsangles

Nome da chave de entrada

123 INPUT

Dicionário de parâmetros em JSON

123 {
 "SELECTED": false,
 "TOLERANCE": 10,
 "FLAGS": "TEMPORARY_OUTPUT"
}

Nome dos parâmetros de saída da camada [opcional]

123 FLAGS

Execução de saída em lote

[Digite o nome se este for um resultado final]

Dependencies

0 dependências selecionadas

OK Cancel Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

Parâmetros em JSON:

```
{
"SELECTED": false,
"TOLERANCE": 10,
"FLAGS": "TEMPORARY_OUTPUT"
}
```

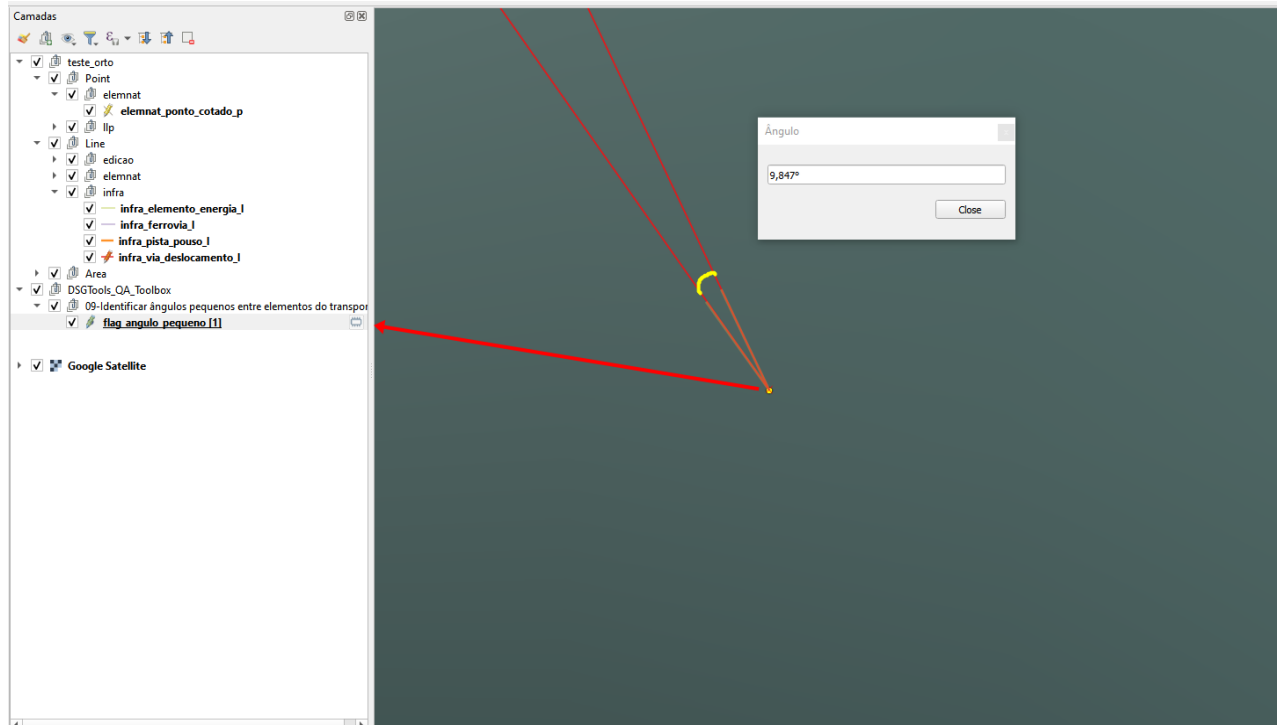
O parâmetro **"SELECTED"** determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como "true", a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

O parâmetro **"TOLERANCE"**: Um ângulo de tolerância de 10 graus indica que qualquer ângulo abaixo desse valor será considerado pequeno. Essa medida é usada para identificar irregularidades na geometria, tais como curvas de nível muito acentuadas ou dobras em trechos de drenagem. Entretanto, é importante mencionar que para outras aplicações, uma tolerância maior ou menor pode ser mais apropriada.

O parâmetro **"FLAGS"** é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação. Neste caso, o valor **"TEMPORARY_OUTPUT"** especifica que uma camada temporária será gerada para armazenar informações sobre os ângulos pequenos encontradas durante a validação.

f) Nome da camada de flags: *flag_angulo_pequeno*

g) Resultado do processo e exemplo de erros:



9- IDENTIFICA ÂNGULOS PEQUENOS ENTRE CAMADAS:

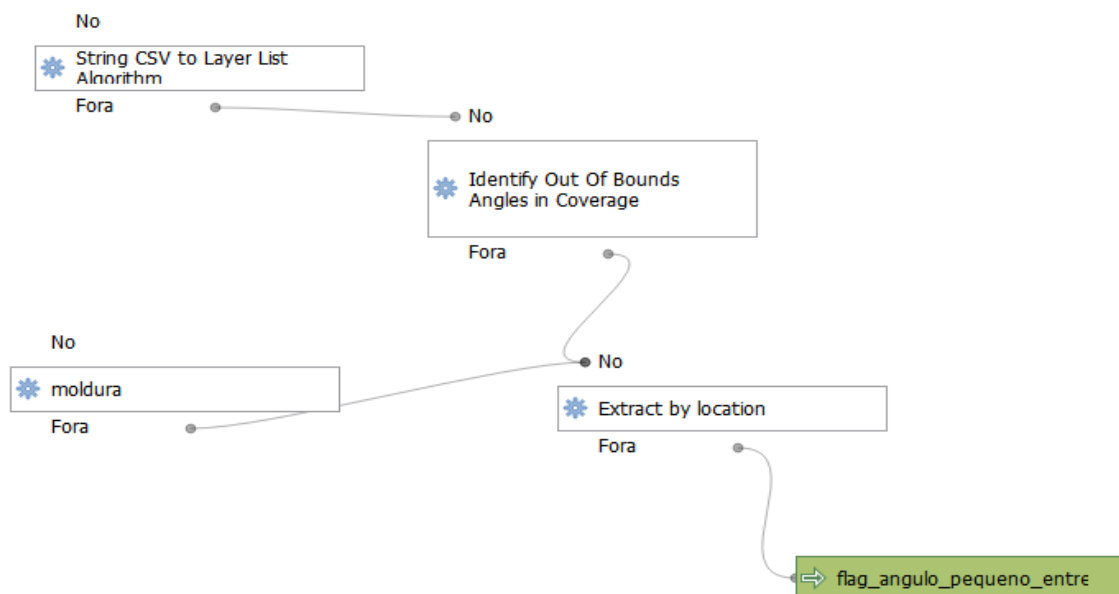
a) Descrição: Este algoritmo funciona de maneira similar à rotina anterior, identificando os ângulos pequenos em uma lista de camadas de linha. A diferença é que as camadas são unificadas, a fim de se comportarem como uma única camada.

b) Arquivo: identifica_angulos_pequenos_entre_camadas_transportes_carta_orto.model3

c) Algoritmo:

dsgtools:stringcsvtolayerlistalgorithm
dsgtools:stringcsvtofirstlayerwiththelementsalgorithm
dsgtools:identifyoutofboundsanglesincoverage
native:extractbylocation

d) Composição do Modelo:



O modelo é composto por quatro etapas:

StringCsvToLayerListAlgorithm: Converte uma *string* CSV em uma lista de camadas.

IdentifyOutOfBoundsAnglesInCoverage: Identifica ângulos pequenos em cada camada de entrada, com base no parâmetro *TOLERANCE*.

Moldura: Converte uma *string* csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwiththelementsalgorithm*.

ExtractByLocation: Extrai os ângulos pequenos que estão dentro da área delimitada pela moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Identificar Ângulos Fora de Limites na Cobertura Terrestre

Properties | Comments

Description: Identify Out Of Bounds Angles in Coverage

Camada de entrada
☐ A usar saída de algoritmo "Lista múltipla de camadas" do algoritmo "String CSV to Layer List Algorithm"

Processar apenas feições selecionadas
☒ Não

Minimum angle (in degrees)

Identificar Ângulos Fora de Limites na Cobertura Terrestre Flags

Dependencies

OK Cancel Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

O parâmetro *"SELECTED"* determina se a validação deve ser realizada apenas nas

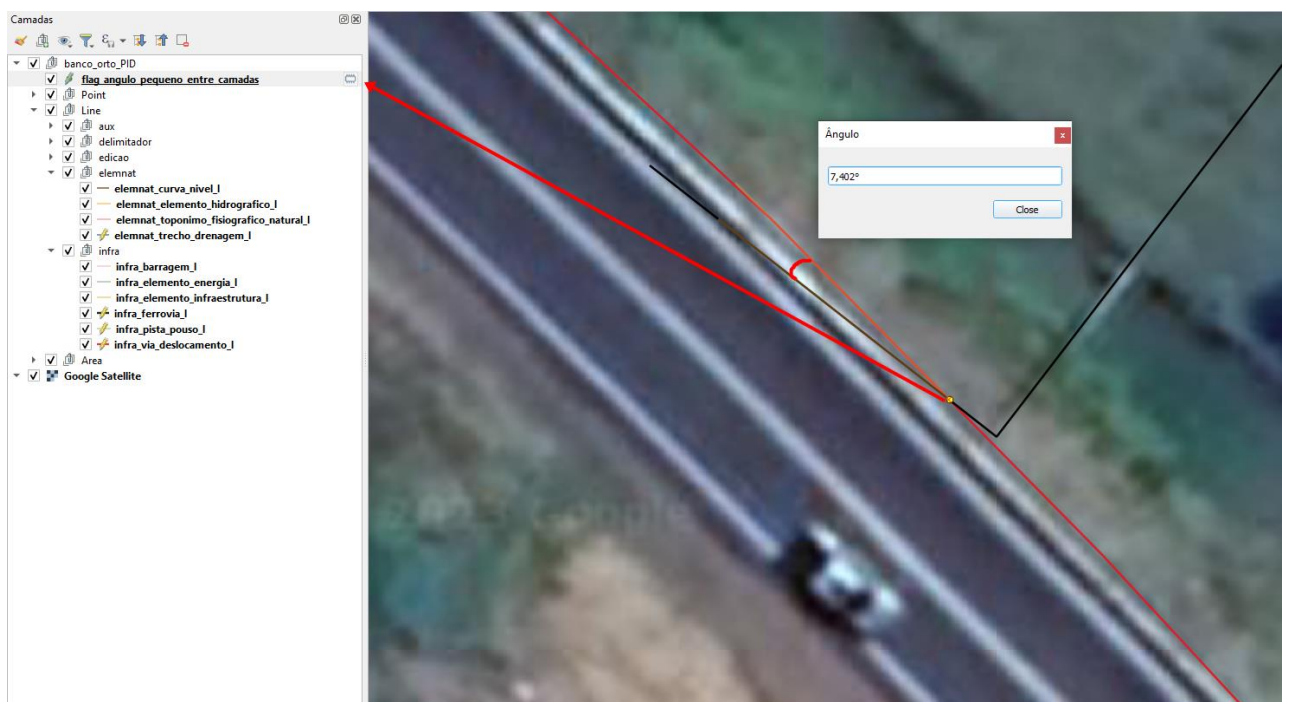
geometrias selecionadas pelo usuário. Quando marcado como "true", a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

O parâmetro **"TOLERANCE"**: Um ângulo de tolerância de 10 graus indica que qualquer ângulo abaixo desse valor será considerado pequeno. Essa medida é usada para identificar irregularidades na geometria, tais como curvas de nível muito acentuadas ou dobras em trechos de drenagem. Entretanto, é importante mencionar que para outras aplicações, uma tolerância maior ou menor pode ser mais apropriada.

O parâmetro **"FLAGS"** é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação. Neste caso, o valor **"TEMPORARY_OUTPUT"** especifica que uma camada temporária será gerada para armazenar informações sobre os ângulos pequenos encontradas durante a validação.

f) Nome da camada de flags: Flag_angulo_pequeno_entre_camadas

g) Resultado do processo e exemplo de erros:



10 – IDENTIFICAR Z (Entrada tá varada)

a) Descrição: O ângulo Z é a diferença entre ângulos formados por três pontos consecutivos. O modelo tem por objetivo encontrar esses ângulos em linhas e polígonos.

b) Arquivo: identifica_z_transportes_carta_orto.model3

c) Algoritmo:

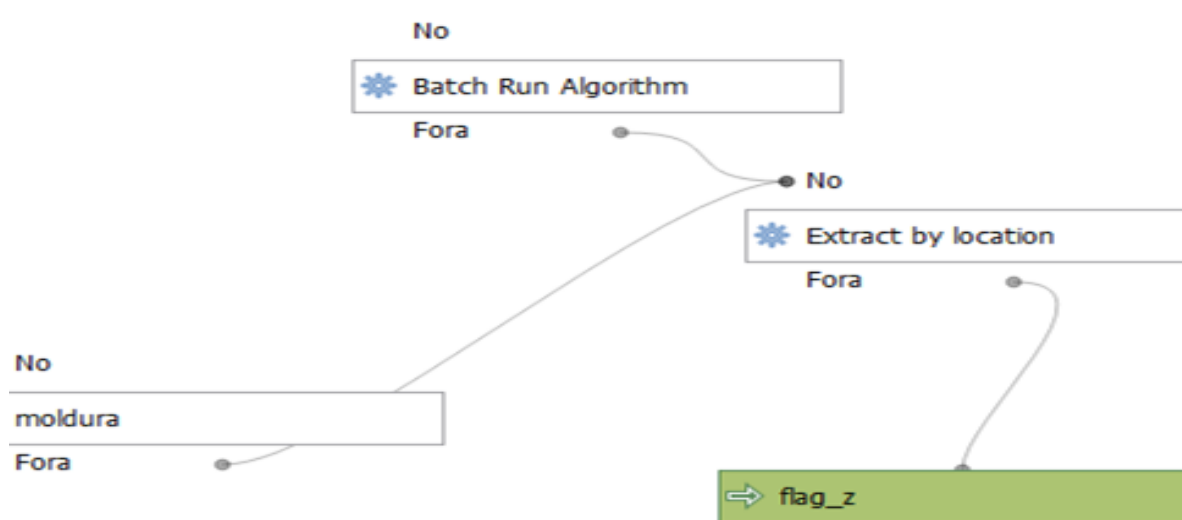
dsgtools:identifyanglesbetweenfeatures

dsgtools:batchrunalgorithm

dsgtools:stringcsvtofirstlayerwithelementsalgorithm

native:extractbylocation

d) Composição do Modelo:



O modelo é composto por três etapas:

Batchrunalgorithm: executa o algoritmo *identifyoutofboundsangles* com os parâmetros selecionados abaixo.

Moldura: Converte uma string csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

ExtractByLocation: Extrai os ângulos pequenos que estão dentro da área delimitada pela moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_1, infra_mobilidade_urbana_1, infra_travessia_hidroviaria_1, infra_via_deslocamento_1

Parâmetros em JSON:

```
{ "ANGLE": 300,  
"OUTPUT": "TEMPORARY_OUTPUT" }
```

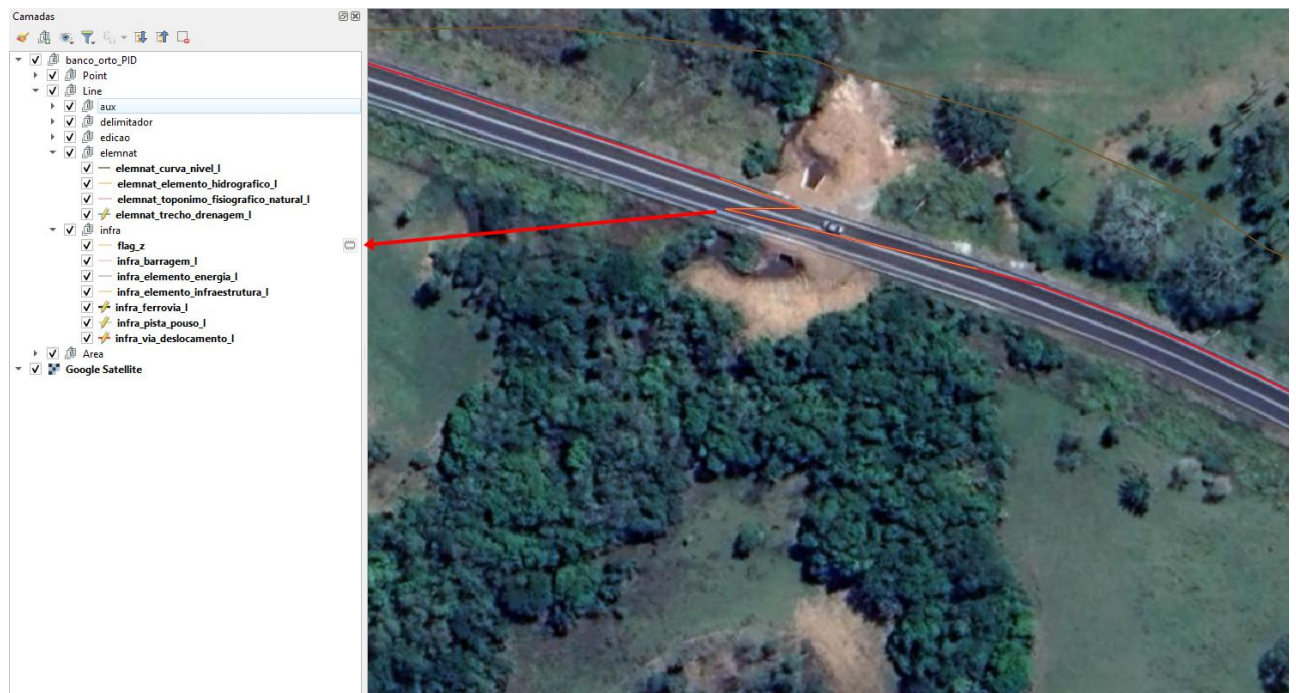
O Parâmetro **"ANGLE"** representa a medida de ângulo em graus, onde qualquer ângulo menor do que 300 graus será considerado pequeno. Essa medida é usada para identificar irregularidades na geometria, entretanto, é importante mencionar que para outras aplicações, uma tolerância maior ou menor pode ser mais apropriada.

O Parâmetro **"OUTPUT"** é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação. Neste caso, o valor **"TEMPORARY_OUTPUT"**

especifica que uma camada temporária será gerada para armazenar informações sobre os ângulos inválidos encontradas durante a validação.

f) nome da camada de flags: Flag_z

g) Resultado do processo e exemplo de erros:



11 - IDENTIFICAR *OVERLAPS* DENTRO DA MESMA CAMADA:

a) Descrição: Este algoritmo identifica sobreposições dentro da mesma camada em uma lista de camadas de linha.

b) Arquivo: identifica_overlaps_linhas_transportes_carta_orto.model3

c) Algoritmo:

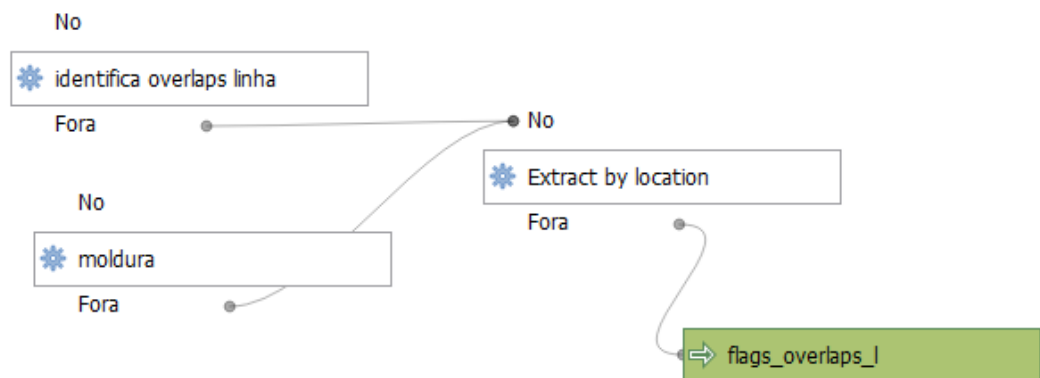
dsgtools:identifyoverlaps

dsgtools:batchrunalgorithm

dsgtools:stringcsvtofirstlayerwithelementsalgorithm

native:extractbylocation

d) Composição do Modelo:



O modelo é composto por três etapas:

Batchrunalgorithm: Executa o algoritmo *identifyoverlaps* com os parâmetros selecionados abaixo para as camadas de linha.

Moldura: Converte uma *string* csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

ExtractByLocation: Extrai os ângulos pequenos que estão dentro da área delimitada pela moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Execução de algoritmos em lote

Properties Comments

Description: identifica overlaps linha

Nomes das camadas de entrada separados por vírgula: infra_ferrovia_l,infra_mobilidade_urbana_l,infra_travessia_hidroviaria_l,infra_via_deslocamento_l

Nome do algoritmo com seu provedor: dsgtools:identifyoverlaps

Nome da chave de entrada: INPUT

Dicionário de parâmetros em JSON: {"SELECTED": false, "FLAGS": "TEMPORARY_OUTPUT"}

Nome dos parâmetros de saída da camada [opcional]: FLAGS

Execução de saída em lote

OK Cancel Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

Parâmetros em JSON:

```
{ "SELECTED": false,  
  "FLAGS": "TEMPORARY_OUTPUT" }
```

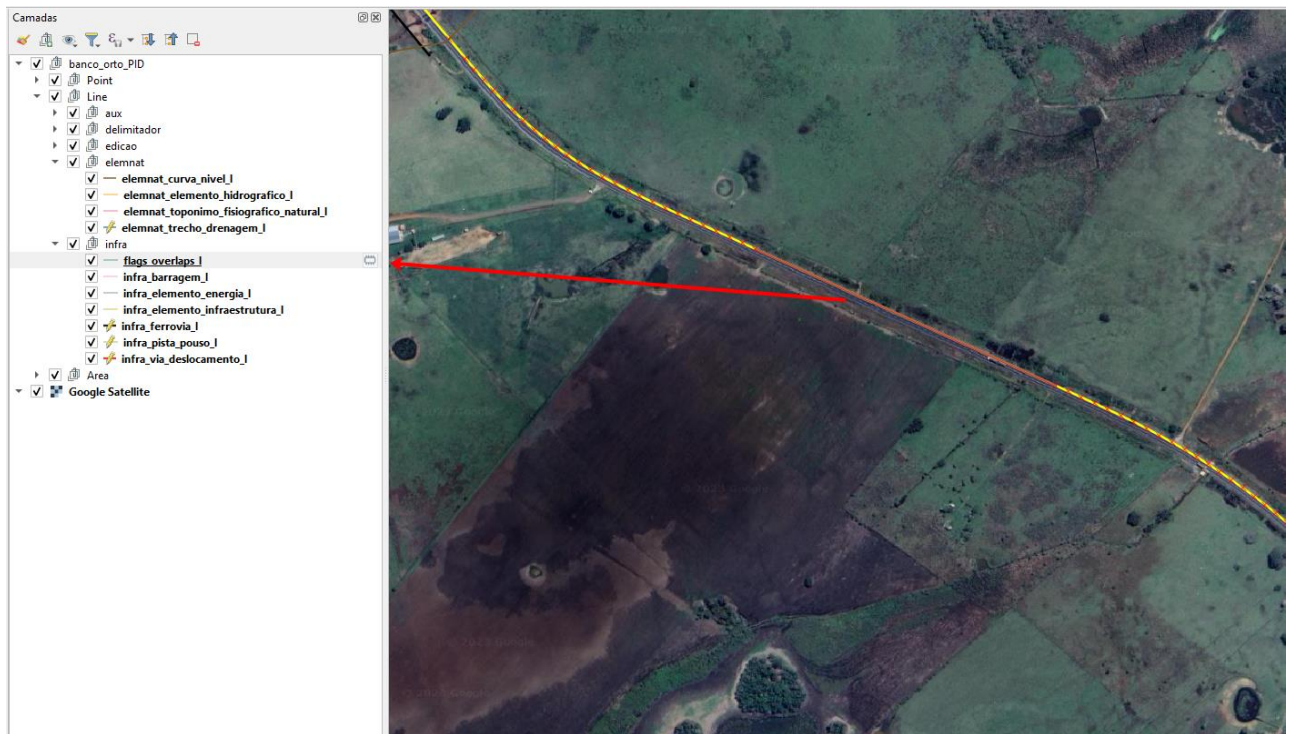
O parâmetro **"SELECTED"** determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como **"true"**, a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

"FLAGS" é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação. Neste caso, o valor **"TEMPORARY_OUTPUT"** especifica

que uma camada temporária será gerada para armazenar informações sobre os overlaps encontradas durante a validação.

f) nome da camada de *flags*: flags_overlaps_1

g) Resultado do processo e exemplo de erros:



12- IDENTIFICA *UNDERSHOOT* COM MOLDURA E CONEXÃO DE LINHAS:

a) Descrição: Este algoritmo identifica pontas soltas dentro da mesma camada em uma lista de camadas de linha.

b) Arquivo: identifica_undershoot_moldura_conexao_linhas_transportes.model3

c) Algoritmo:

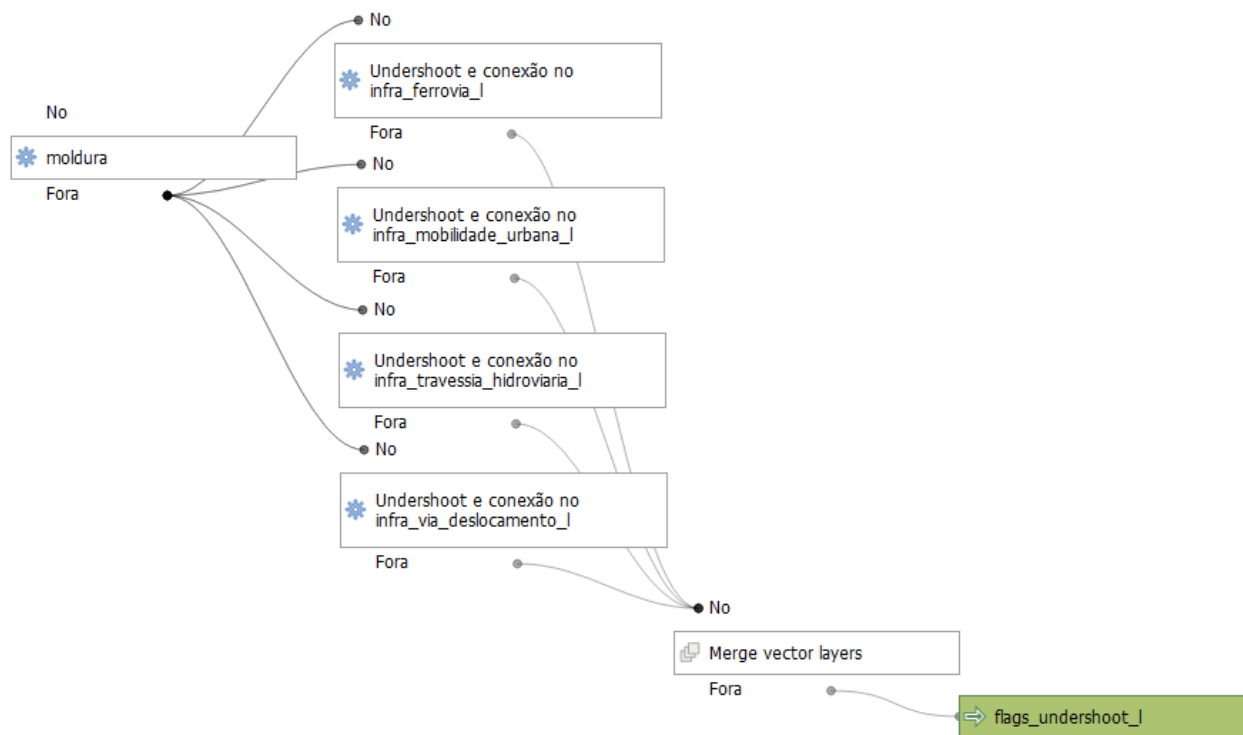
dsgtools:stringcsvtofirstlayerwithelementsalgorithm

dsgtools:identifydangles

native:mergevectorlayers

dsgtools:batchrunalgorithm

d) Composição do Modelo:



O modelo é composto por três etapas:

Moldura: Converte uma *string* csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

Batchrunalgorithm: Executa os algoritmos *identifydangles* para as camadas de linha com os parâmetros descritos abaixo.

Merge Vector Layers: Realiza uma operação de unir as camadas de saída utilizando o algoritmo *mergevectorlayers*

e) Parâmetros:

Identificar Pontas Soltas

Properties

Comments

Description

Undershoot e conexão no infra_ferrovia_l

Camada de entrada

123 infra_ferrovia_l

Processar apenas feições selecionadas

123 Não

Input is a boundary layer (every line must be connected to an element of either the input layer or the filters)

123 Não

Raio de busca

123 0,000010

Camada de filtro do tipo linha [opcional]

123 0 selecionar entradas

Camadas de filtro do tipo polígono [opcional]

A usar saída de algoritmo

"Loaded layer" do algoritmo "moldura"

Ignorar pontas soltas em linhas não segmentadas

123 Sim

Geographic Boundary (this layer only filters the output dangles) [opcional]

A usar saída de algoritmo

"Loaded layer" do algoritmo "moldura"

Identificar Pontas Soltas Flags

[Digite o nome se este for um resultado final]

Dependencies

0 dependências selecionadas

OK

Cancel

Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

O parâmetro **"INPUT"** determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário.

O parâmetro **"SELECTED"** determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como **"True"**, a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

O parâmetro **"INPUT IS A BOUNDARY LAYER"** Se a opção **"False"** estiver definida no algoritmo, isso significa que a moldura não precisa estar conectada a elementos da camada de entrada ou dos filtros. É importante ressaltar que o campo `elemnat_curva_nivel` a opção precisa ser **"True"**, por se tratar de um contorno e ter um comportamento particular.

O Parâmetro: **"SEARCH RADIUS"** do algoritmo determina a distância máxima que será percorrida para encontrar pontas soltas em relação aos segmentos de linha e polígonos que se cruzam nas intersecções. No modelo, o valor padrão do raio de busca é definido como 0,00001 unidades do sistema de coordenadas utilizado no processamento. No entanto, é importante lembrar que esse valor pode ser ajustado de acordo com as necessidades específicas, considerando a escala e precisão dos dados utilizados no processamento

O parâmetro **" CAMADA DE FILTRO"** tem como proposta identificar pontas soltas próximas às camadas especificadas. Essas pontas soltas serão adicionadas à regra de verificação de proximidade durante a busca por feições próximas às analisadas, a fim de minimizar a ocorrência de falsos positivos.

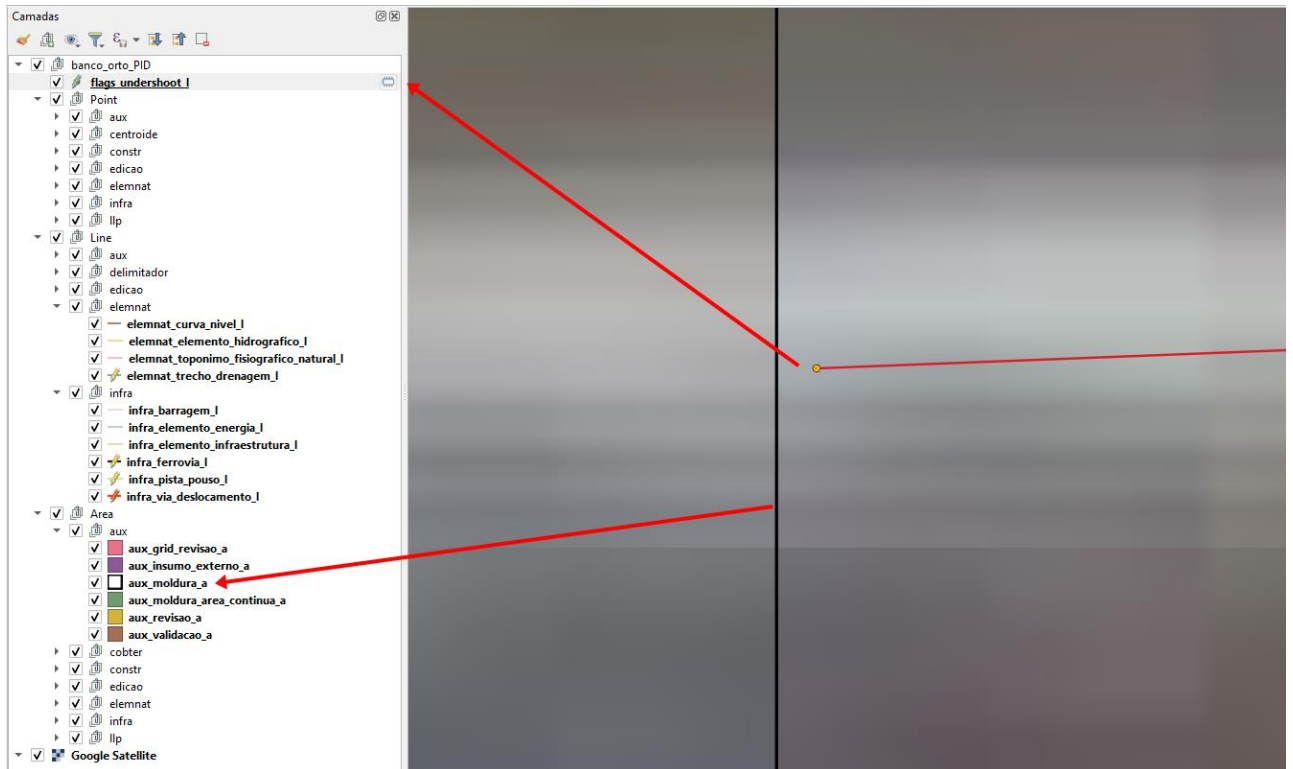
O parâmetro **"IGNORAR PONTAS SOLTAS EM LINHAS NÃO SEGMENTADAS"** quando marcado como **"True"** permite que o algoritmo ignore pontas soltas em linhas não segmentadas. Ou seja, ele não considera como *undershoots* aquelas pontas soltas que não estão conectadas a outros segmentos de linha. Ao ignorar essas pontas, o algoritmo pode concentrar-se apenas nas linhas que realmente importam e fornecer resultados mais precisos.

O parâmetro **"GEOGRAPHIC BOUNDARY"** se refere à delimitação geográfica para a execução do processamento em uma região determinada. A "moldura" representa a área delimitada em questão, isso evita que sejam incluídas informações irrelevantes ou imprecisas de outras áreas que estão fora da delimitação.

O parâmetro **"FLAGS"** é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação.

f) nome da camada de flags: `flags_undershoot_l`

g) Resultado do processo e exemplo de erros:



13- IDENTIFICA LINHAS SEGMENTADAS COM MESMO CONJUNTO DE ATRIBUTOS:

a) Descrição: Este algoritmo identifica linhas segmentadas com mesmo conjunto de atributos em uma lista de camadas de linha.

b) Arquivo:

identifica_linhas_segmentadas_com_mesmo_conjunto_de_atributos_transportes_carta_orto.model

3

c) Algoritmo:

dsgtools:stringcsvtolayerlistalgorithm

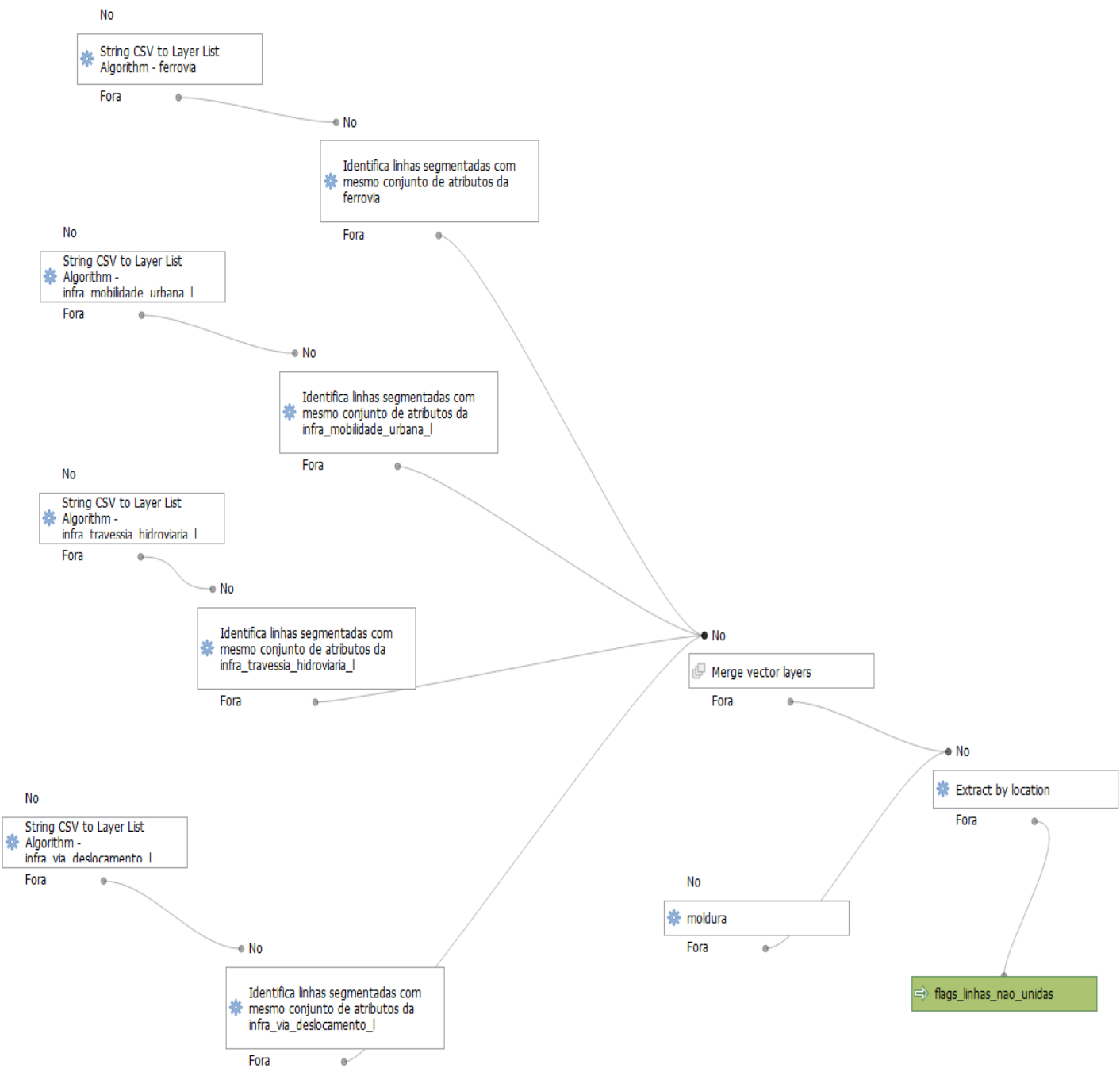
dsgtools:identifyunmergedlineswithsameattributeset

dsgtools:stringcsvtofirstlayerwiththelementsalgorithm

native:mergevectorlayers

native:extractbylocation

d) Composição do Modelo:



O modelo é composto por cinco etapas:

String CSV to Layer List Algorithm: Converte uma *string* csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

Identificar linhas Segmentadas: Identifica as linhas segmentadas com os parâmetros definidos abaixo utilizando o algoritmo *identifysplitlines*.

Merge Vector Layers: Realiza uma operação de unir as camadas de saída utilizando o algoritmo *mergevectorlayers*.

Moldura: Converte uma *string* csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*

ExtractByLocation: Extrai os ângulos pequenos que estão dentro da área delimitada pela moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Identify Unmerged Lines With Same Attribute Set

Properties

Comments

Description

Identifica linhas segmentadas com mesmo conjunto de atributos da ferrovia

Input layer

123

infra_ferrovia_1

Process only selected features

123

Não

Fields to ignore [opcional]

123

ificativa_txt;tamanho_txt;visivel;carta_simbolizacao;simbolizar_carta_mini;simb_rot;rotular_carta_mini;espacamento;tamanho_txt;estilo_fonte;cor;cor_buffer;tamanho_

Ignore virtual fields

123

Sim

Ignore primary key fields

123

Sim

Point Filter Layers [opcional]

123

0 selecionar entradas

Line Filter Layers [opcional]

A usar saída de algoritmo

"Lista múltipla de camadas" do algoritmo "String CSV to Layer List Algorithm - ferrovia"

Identify Unmerged Lines With Same Attribute Set Flags

[Digite o nome se este for um resultado final]

Dependencies

0 dependências selecionadas

OK

Cancel

Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_1, infra_mobilidade_urbana_1, infra_travessia_hidroviaria_1, infra_via_deslocamento_1

O parâmetro **"SELECTED"** determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como **"true"**, a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

No parâmetro **"FIELDS TO IGNORE"** é definido uma lista negra de atributos (**ATTRIBUTE_BLACKLIST**) que não serão considerados na comparação das feições, a fim de evitar falsos positivos.

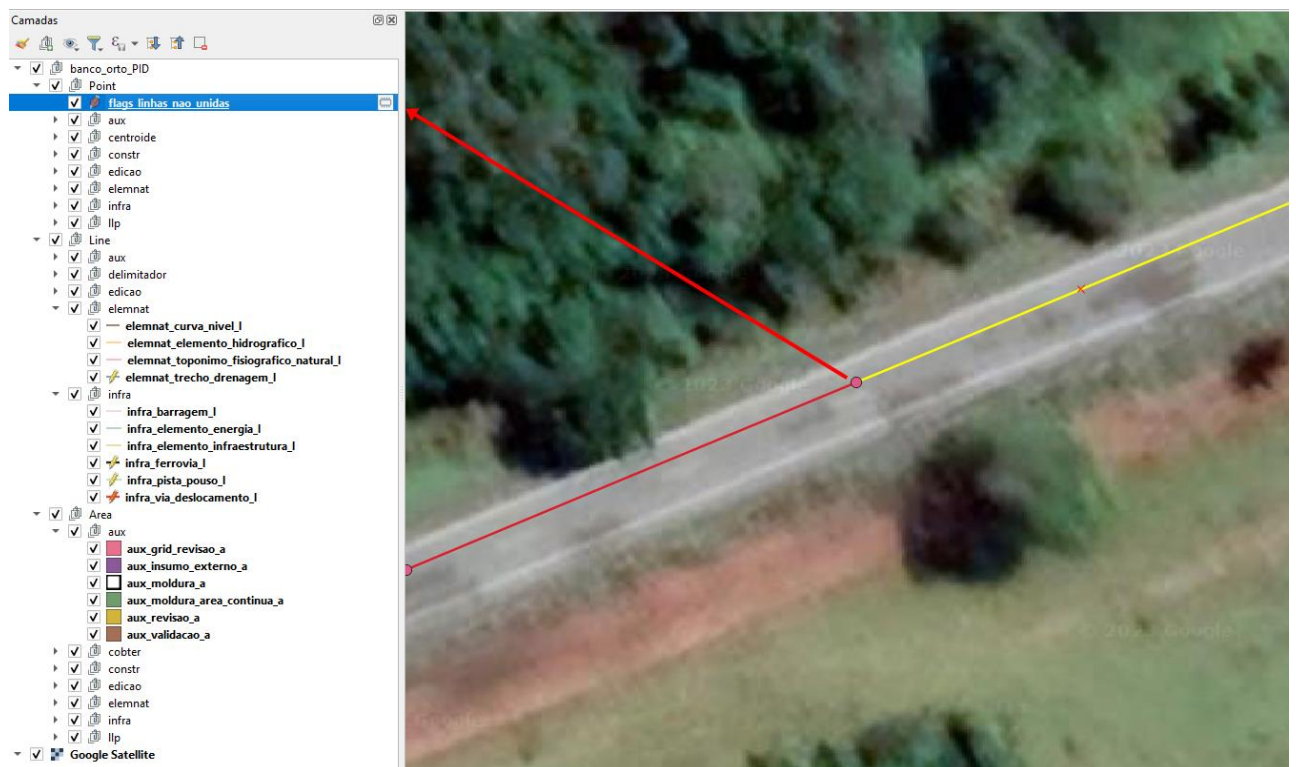
O parâmetro **"IGNORE_PK_FIELDS"** é um booleano que indica se os campos da chave primária das camadas de entrada devem ser ignorados na comparação. (apesar de estar marcado como **"true"**, o campo de chave primária **"id"** já está sendo ignorado na lista negra de atributos).

O parâmetro **"IGNORE_VIRTUAL_FIELDS"** é um booleano que indica se os campos virtuais (calculados dinamicamente) devem ser ignorados na comparação.

O parâmetro: **"FLAGS"** é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação.

f) nome da camada de *flags*: flags_linhas_nao_unidas

g) Resultado do processo e exemplo de erros:



14 - IDENTIFICAR LINHAS NÃO SEGMENTADAS NAS INTERSECÇÕES:

a) Descrição: O algoritmo em questão tem como finalidade detectar linhas não segmentadas nas intersecções de camadas. Essa tarefa é realizada através da identificação de geometrias não conectadas que não compartilham vértices. A partir dessa análise, é possível determinar se as linhas com vértices não conectados em sua extensão devem ser classificadas como **"dangles"** (pontas soltas) e, portanto, descartadas, ou se devem ser consideradas como não segmentadas dentro das camadas especificadas em uma lista de camadas de linha.

b) Arquivo: identificar_linhas_nao_segmentadas_nas_interseccoes_transportes.model3

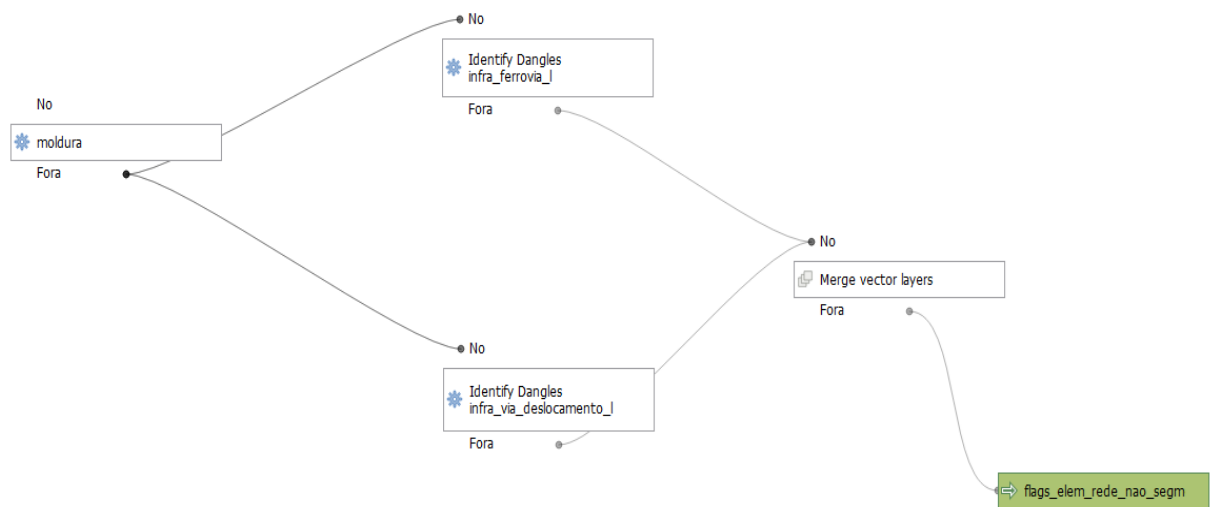
c) Algoritmo:

`dsgtools:stringcsvtofirstlayerwithelementsalgorithm`

`dsgtools:identifydanglesalgorithm`

`native:mergevectorlayers`

d) Composição do Modelo:



O modelo é composto por três etapas:

Moldura: Converte uma string csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

Identify Dangles: Identifica pontas soltas na camada de *infra_ferrovia_l* e *infra_via_deslocamento* com o algoritmo *identifydanglesalgorithm*

Marger Vector Layers: Realiza uma operação de unir as camadas de saída utilizando o algoritmo *mergevectorlayers*.

e) Parâmetros:

Identificar Pontas Soltas

Properties Comments

Description Identify Dangles infra_ferrovia_1

Camada de entrada
123 infra_ferrovia_1

Processar apenas feições selecionadas
123 Não

Input is a boundary layer (every line must be connected to an element of either the input layer or the filters)
123 Não

Raio de busca
123 0,000100

Camada de filtro do tipo linha [opcional]
123 0 selecionar entradas

Camadas de filtro do tipo polígono [opcional]
A usar saída de algoritmo "Loaded layer" do algoritmo "moldura"

Ignorar pontas soltas em linhas não segmentadas
123 Não

Geographic Boundary (this layer only filters the output dangles) [opcional]
A usar saída de algoritmo "Loaded layer" do algoritmo "moldura"

Identificar Pontas Soltas Flags
[Digite o nome se este for um resultado final]

Dependencies
0 dependências selecionadas

OK Cancel Help

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_1 infra_via_deslocamento_1

O parâmetro **"INPUT"** determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário.

O parâmetro **"SELECTED"** determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como **"true"**, a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

O parâmetro **"INPUT IS A BOUNDARY LAYER"** Se a opção **"False"** estiver definida no algoritmo, isso significa que a moldura não precisa estar conectada a elementos da camada de entrada ou dos filtros.

O Parâmetro: **"SEARCH RADIUS"** do algoritmo determina a distância máxima que será percorrida para encontrar pontas soltas em relação aos segmentos de linha que se cruzam nas intersecções. No modelo, o valor padrão do raio de busca é definido como 0,00001 unidades do sistema de coordenadas utilizado no processamento. No entanto, é importante lembrar que esse valor pode ser ajustado de acordo com as necessidades específicas, considerando a escala e precisão dos dados utilizados no processamento.

O parâmetro **"CAMADA DE FILTRO"** usados para limitar as camadas que serão analisadas no algoritmo.

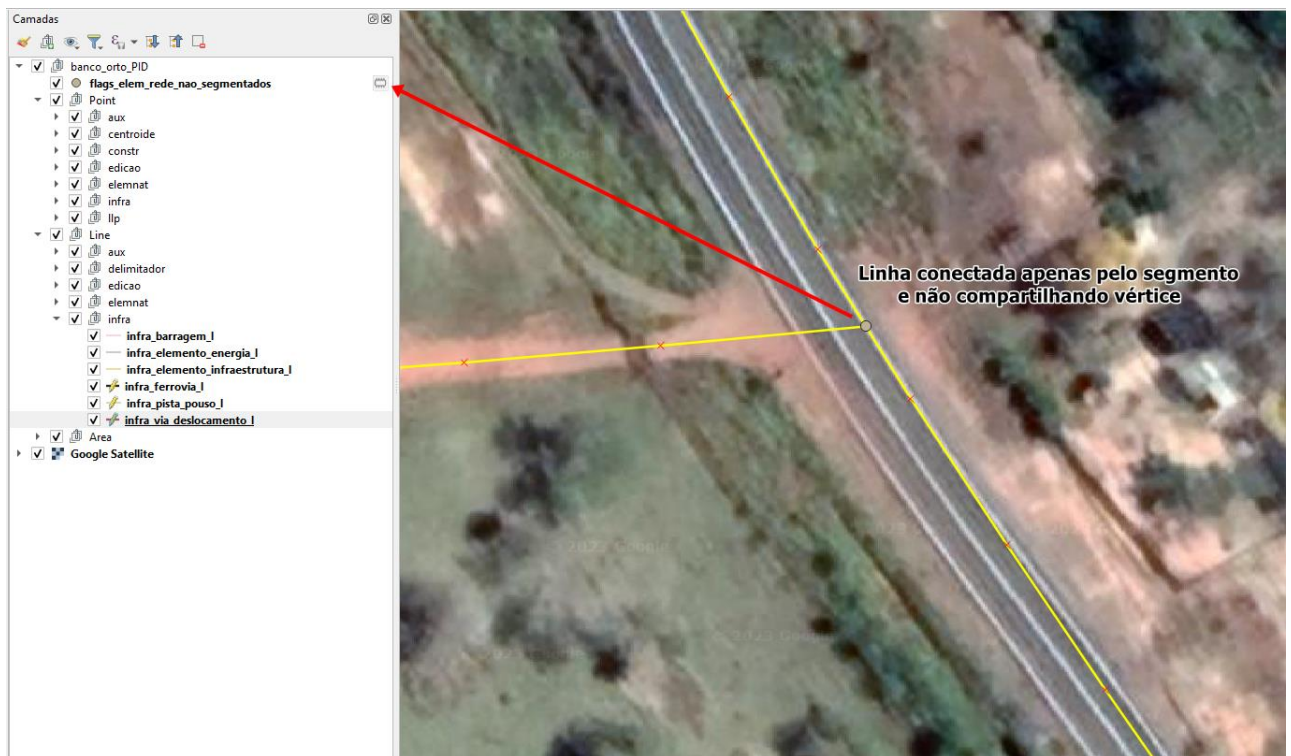
O parâmetro **"IGNORAR PONTAS SOLTAS EM LINHAS NÃO SEGMENTADAS"** quando marcado como **"False"** permite que o algoritmo encontre pontas soltas em linhas não segmentadas. Ou seja, considera como *undershoots* aquelas pontas soltas que não estão conectadas a outros segmentos de linha.

O parâmetro **"GEOGRAPHIC BOUNDARY"** se refere à delimitação geográfica para a execução do processamento em uma região determinada. A "moldura" representa a área delimitada em questão, isso evita que sejam incluídas informações irrelevantes ou imprecisas de outras áreas que estão fora da delimitação.

O parâmetro **"FLAGS"** é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação.

f) nome da camada de flags: *flags_undershoot_l*

g) Resultado do processo e exemplo de erros:



15- IDENTIFICAR ELEMENTOS PEQUENOS NA REDE:

a) Descrição: Este algoritmo realiza a identificação de elementos com tamanhos menor que o estabelecido na tolerância em uma camada pré-determinada, ao mesmo tempo em que detecta segmentos de transporte que se encontram desconectados da rede.

b) Arquivo: identificar_elementos_pequenos_na_rede_transportes.model3

c) Algoritmo:

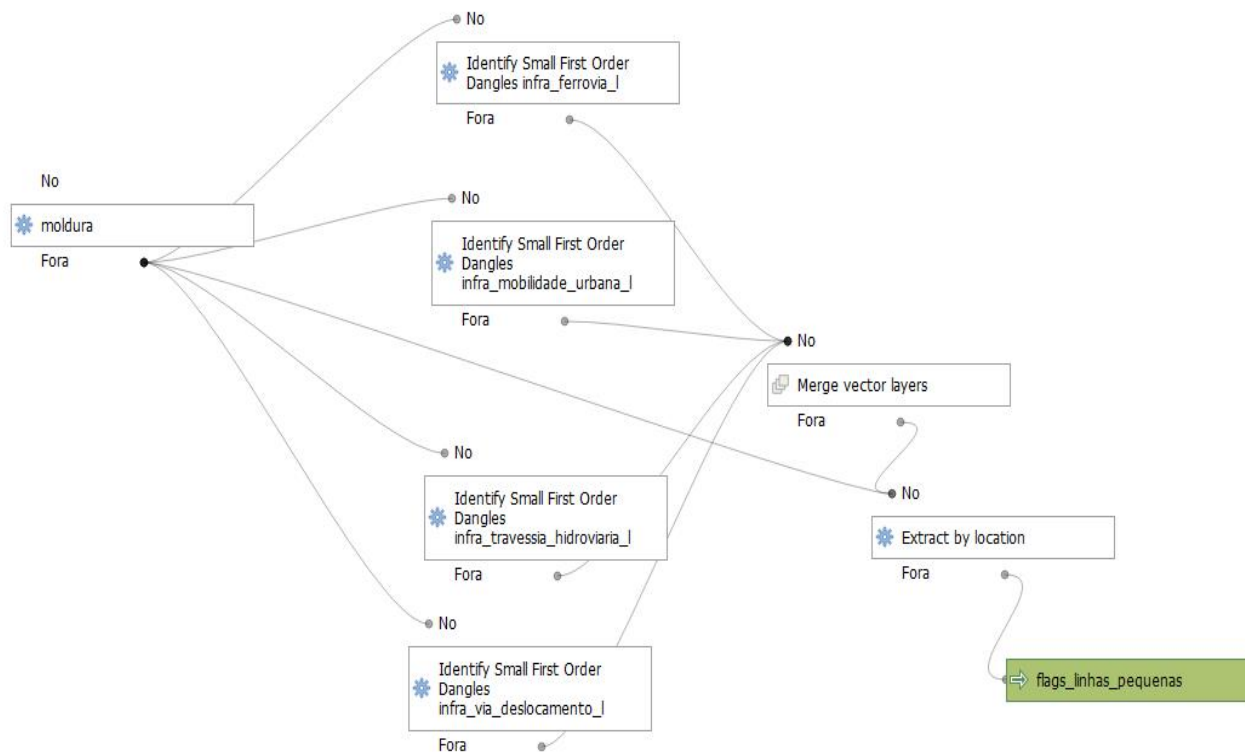
dsgtools:stringcsvtofirstlayerwiththelementsalgorithm

dsgtools:identifysmallfirstorderdangles

native:mergevectorlayers

native:extractbylocation

d) Composição do Modelo:



O modelo é composto por quatro etapas:

Moldura: Converte uma *string* csv em uma camada especificada usando o algoritmo *stringcsvtofirstlayerwithelementsalgorithm*.

Identify Small First Order Dangles: Procura por pequenos segmentos de linha que não se conectam a outros segmentos ou estão muito próximos de uma interseção, executando o algoritmo *identifysmallfirstorderdangles*.

Merge Vector Layers: Realiza uma operação de unir as camadas de saída utilizando o algoritmo *mergevectorlayers*.

ExtractByLocation: Extrai os elementos pequenos que estão dentro da área delimitada pela moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Identify Small First Order Dangles

Properties Comments

Description: Identify Small First Order Dangles infra_ferrovia_

Input layer: 123 infra_ferrovia_

Process only selected features: 123 Não

Minimum size: 123 0,000050

Search radius: 123 0,010000

Linestring Filter Layers [opcional]: 123 0 selecionar entradas

Polygon Filter Layers [opcional]: 123 0 selecionar entradas

Geographic Boundary (this layer only filters the output dangles) [opcional]: A usar saída de algoritmo "Loaded layer" do algoritmo "moldura"

Identify Small First Order Dangles Flags: [Digite o nome se este for um resultado final]

Dependencies: 0 dependências selecionadas

OK Cancel Help

O algoritmo "*Identify Small First Order Dangles*" é uma ferramenta utilizada para identificar feições que não estão conectadas na rede, também conhecidos como "*dangles*" ou pontas soltas. Ele busca por trechos de linha que possuem um único vértice que não é conectado a nenhum outro trecho. Para evitar a identificação de *dangles* não relevantes, o algoritmo se concentra em identificar apenas aqueles que têm um comprimento mínimo especificado.

O parâmetro "***SELECTED***" determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como "*true*", a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

O parâmetro "***MINIMUM_SIZE***" se concentra em identificar apenas *dangles* pequenos, ou seja, aqueles que têm um comprimento mínimo de 0,001 e estão localizados dentro de um raio de pesquisa definido pelo parâmetro "***SEARCH_RADIUS***" de 0,0001. permitindo identificar possíveis problemas na rede.

O parâmetro "***FLAGS***" é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação. Neste caso, o valor "***TEMPORARY_OUTPUT***"

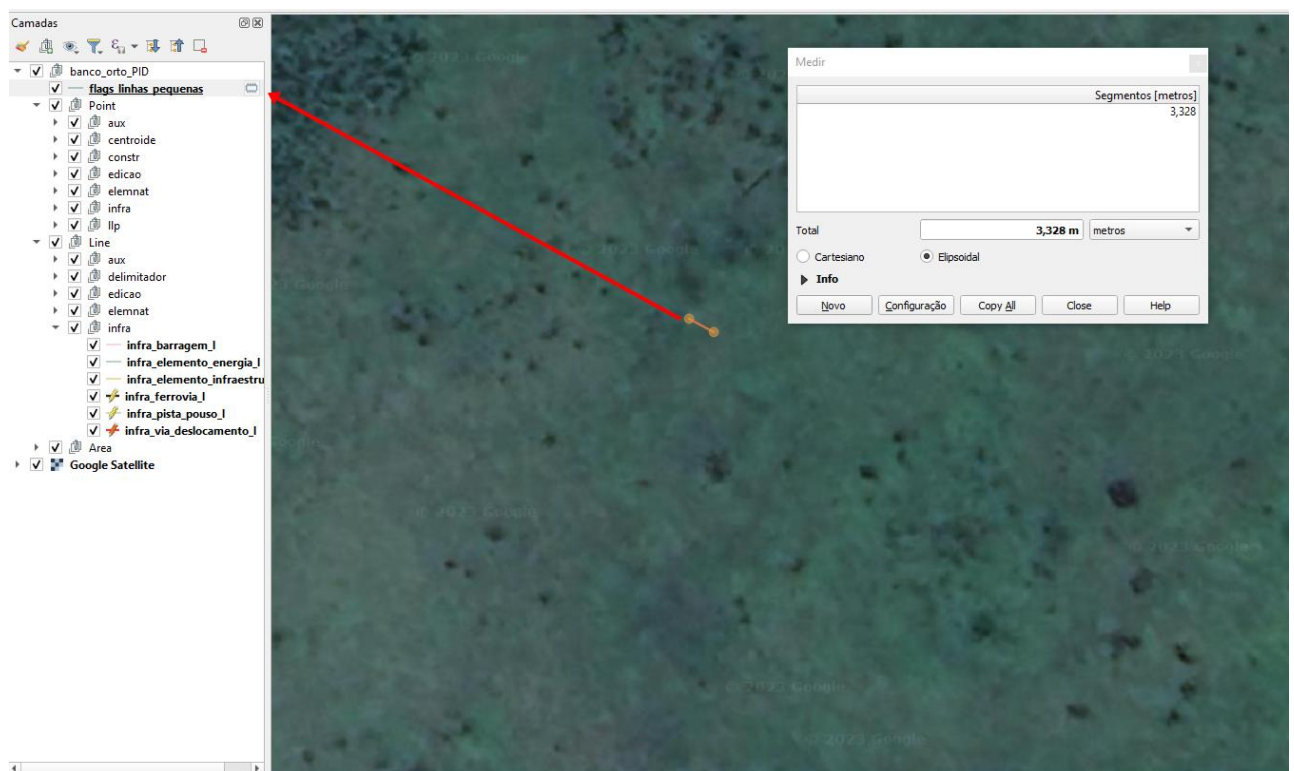
especifica que uma camada temporária será gerada para armazenar informações sobre os elementos pequenos e possíveis problemas encontrados na rede.

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_ferrovia_l, infra_mobilidade_urbana_l, infra_travessia_hidroviaria_l, infra_via_deslocamento_l

f) nome da camada de *flags*: flags_linhas_pequenas.

g) Resultado do processo e exemplo de erros:



16 - IDENTIFICAR ERROS NA CONSTRUÇÃO DAS REDES DE RODOVIÁRIAS E FERROVIÁRIAS:

a) Descrição: Tem como objetivo detectar e identificar erros relacionados à construção da rede de transportes através da detecção de ângulos agudos, pontas soltas e linhas não segmentadas desconectadas da rede.

b) Arquivo: identificar_erros_rede_transporte.model3

c) Algoritmos:

dsgtools:stringcsvtolayerlistalgorithm

dsgtools:identifyunmergedlineswithsameattributeset

dsgtools:stringcsvtofirstlayerwithelementsalgorithm

dsgtools:identifynetworkconstructionissues

native:mergevectorlayers

native:extractbylocation

d) Composição do Modelo:

Identify Unmerged Lines With Same Attribute Set: Responsável por identificar linhas não mescladas (*unmerged*) com o mesmo conjunto de atributos. Ele compara os atributos de todas as linhas dentro de uma camada especificada e verifica se duas ou mais linhas estão com o mesmo conjunto de atributos utilizando o algoritmo

dsgtools:identifyunmergedlineswithsameattributeset.

Merge Vector Layers: Realiza uma operação de unir as camadas de saída utilizando o algoritmo *mergevectorlayers*.

ExtractByLocation: Extrai os erros que estão dentro da área delimitada pela moldura usando o algoritmo *extractbylocation*.

e) Parâmetros:

Identify Network's Geometry Construction Issues

Properties Comments

Description Identify Network's Geometry Construction Issues infra_via_deslocamento_I

Input lines

A usar saída de algoritmo "Lista múltipla de camadas" do algoritmo "infra_via_deslocamento_I"

Process only selected features

123 Não

Input is a boundary layer (every line must be connected to an element of either the input layer or the filters)

123 Não

Search radius

123 0,000010

Linestring Filter Layers [opcional]

A usar saída de algoritmo "Lista múltipla de camadas" do algoritmo "infra_ferrovia_I"

Polygon Filter Layers [opcional]

123 0 selecionar entradas

Ignore dangle on unsegmented lines

123 Não

Geographic Boundary (this layer only filters the output dangles) [opcional]

A usar saída de algoritmo "Loaded layer" do algoritmo "moldura"

Identify Network's Geometry Construction Issues Flags

[Digite o nome se este for um resultado final]

Dependencies

0 dependências selecionadas

OK Cancel Help

O parâmetro **"SELECTED"** determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como **"true"**, a verificação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são verificadas.

O parâmetro **"INPUT IS A BOUNDARY LAYER"** Se a opção "False" estiver definida no algoritmo, isso significa que a moldura não precisa estar conectada a elementos da camada de entrada ou dos filtros.

O Parâmetro: **"SEARCH RADIUS"** do algoritmo determina a distância máxima que será percorrida para encontrar pontas soltas em relação aos segmentos de linha e polígonos que se cruzam nas intersecções. No modelo, o valor padrão do raio de busca é definido como 0,00001 unidades do sistema de coordenadas utilizado no processamento. No entanto, é importante lembrar que esse valor pode ser ajustado de acordo com as necessidades específicas, considerando a escala e precisão dos dados utilizados no processamento.

O parâmetro **"FILTER LAYER"** As camadas de filtro presentes são utilizadas com o propósito de especificar as camadas que serão empregadas para filtrar a identificação de linhas que apresentem problemas de construção na rede de drenagem. As camadas filtradas são, então, empregadas para a identificação de problemas geométricos na construção da rede.

O parâmetro **"GEOGRAPHIC BOUNDARY"** se refere à delimitação geográfica para a execução do processamento em uma região determinada. A "moldura" representa a área delimitada em questão, isso evita que sejam incluídas informações irrelevantes ou imprecisas de outras áreas que estão fora da delimitação.

O parâmetro **"FLAGS"** é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações de validação. Neste caso, o valor **"TEMPORARY_OUTPUT"** especifica que uma camada temporária será gerada para armazenar informações sobre os problemas de construção na rede de drenagem.

Identify Unmerged Lines With Same Attribute Set

Properties Comments

Description Identify Unmerged Lines With Same Attribute Set infra_via_deslocamento_1

Input layer
123 infra_via_deslocamento_1

Process only selected features
123 Não

Fields to ignore [opcional]
123 o_txt;visivel;carta_simbolizacao;simbolizar_carta_mini;simb_rot;rotular_carta_mini;espacamento;tamanho_txt;estilo_fonte;cor;cor_buffer;tamanho_buffe

Ignore virtual fields
123 Sim

Ignore primary key fields
123 Sim

Point Filter Layers [opcional]
123 0 selecionar entradas

Line Filter Layers [opcional]
A usar saída de algoritmo "Lista múltipla de camadas" do algoritmo "infra_ferrovia_1"

Identify Unmerged Lines With Same Attribute Set Flags
123 [Digite o nome se este for um resultado final]

Dependencies
0 dependências selecionadas

OK Cancel Help

O parâmetro **"SELECTED"** determina se a validação deve ser realizada apenas nas geometrias selecionadas pelo usuário. Quando marcado como **"true"**, a validação é realizada apenas nas geometrias selecionadas. Caso contrário, todas as geometrias são validadas.

No parâmetro **"FIELDS TO IGNORE"** é definido uma lista negra de atributos (**ATTRIBUTE_BLACKLIST**) que não serão considerados na comparação das feições, a fim de evitar falsos positivos.

O parâmetro **"IGNORE_PK_FIELDS"** é um booleano que indica se os campos da chave primária das camadas de entrada devem ser ignorados na comparação. (apesar de estar marcado como **"true"**, o campo de chave primária **"id"** já está sendo ignorado na lista negra de atributos).

O parâmetro **"IGNORE_VIRTUAL_FIELDS"** é um booleano que indica se os campos virtuais (calculados dinamicamente) devem ser ignorados na comparação.

O parâmetro **"FILTER LAYER"** as camadas de filtro presentes são usadas para especificar camadas que serão usadas para identificar linhas que não foram unidas corretamente.

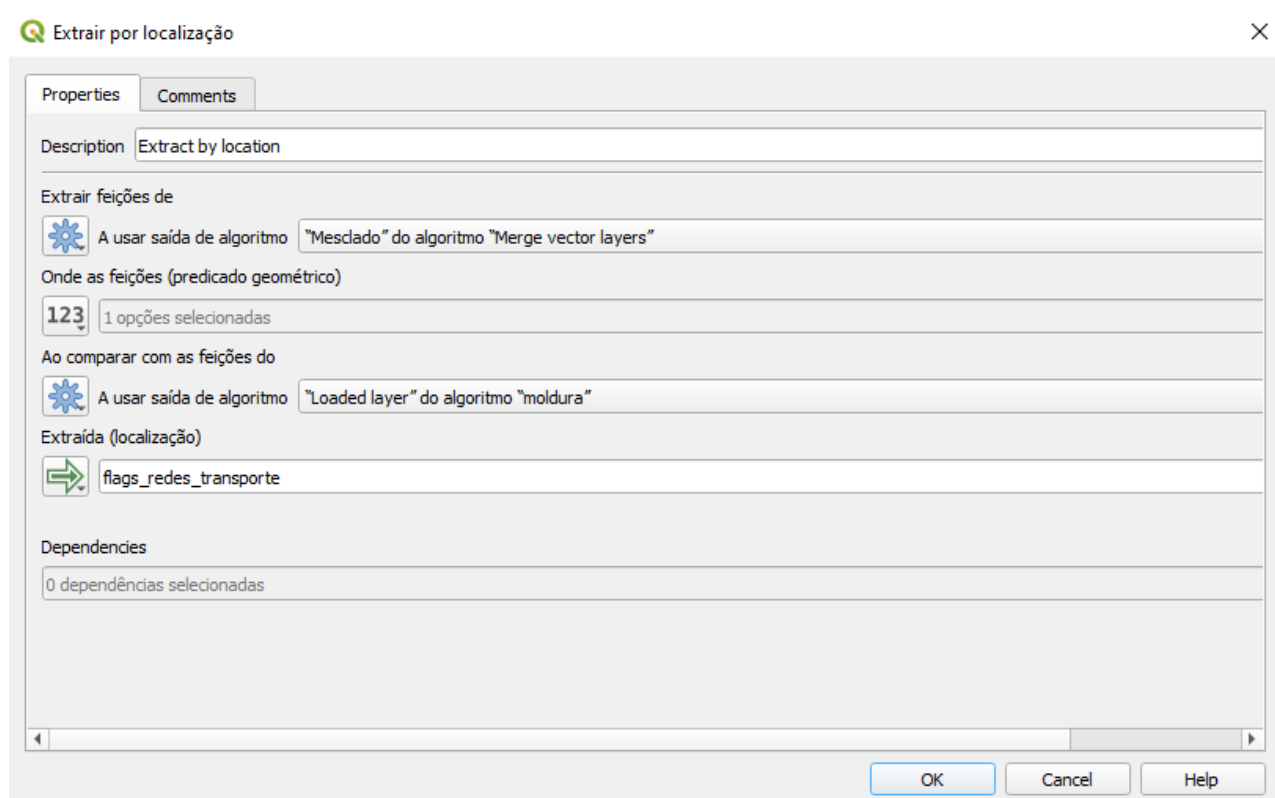
O parâmetro **"FLAGS"** é utilizado para definir o nome da camada que será criada pela rotina para armazenar as informações que necessitam de correção. Neste caso, o valor

"*TEMPORARY_OUTPUT*" especifica que uma camada temporária será gerada para armazenar informações sobre as linhas que não foram unidas corretamente.

Camadas de entrada:

Tipo de geometria	Camadas de entrada
Linha	infra_via_deslocamento_1 infra_ferrovia_1

Por fim é utilizado o algoritmo de extrair pela localização para filtrar os elementos que estão dentro da moldura.



Extrair feições de: Utiliza os dados de saída do procedimento anterior.

Predicado geométrico: Uma lista que especifica o tipo de relação espacial entre as geometrias que devem ser consideradas na extração. Neste caso, o valor [0] significa que apenas as geometrias que se intersectam serão extraídas.

F) nome da camada de *flags*: flags_rede_transportes

g) Resultado do processo e exemplo de erros:

