Operações espaciais no QGIS

Disciplina: Programação aplicada à engenharia cartográfica

Maurício C. M. de Paulo - D.Sc.

23 de fevereiro de 2026

Objetivos

Ilustrar operações de geoprocessamento em dados matriciais e vetoriais.

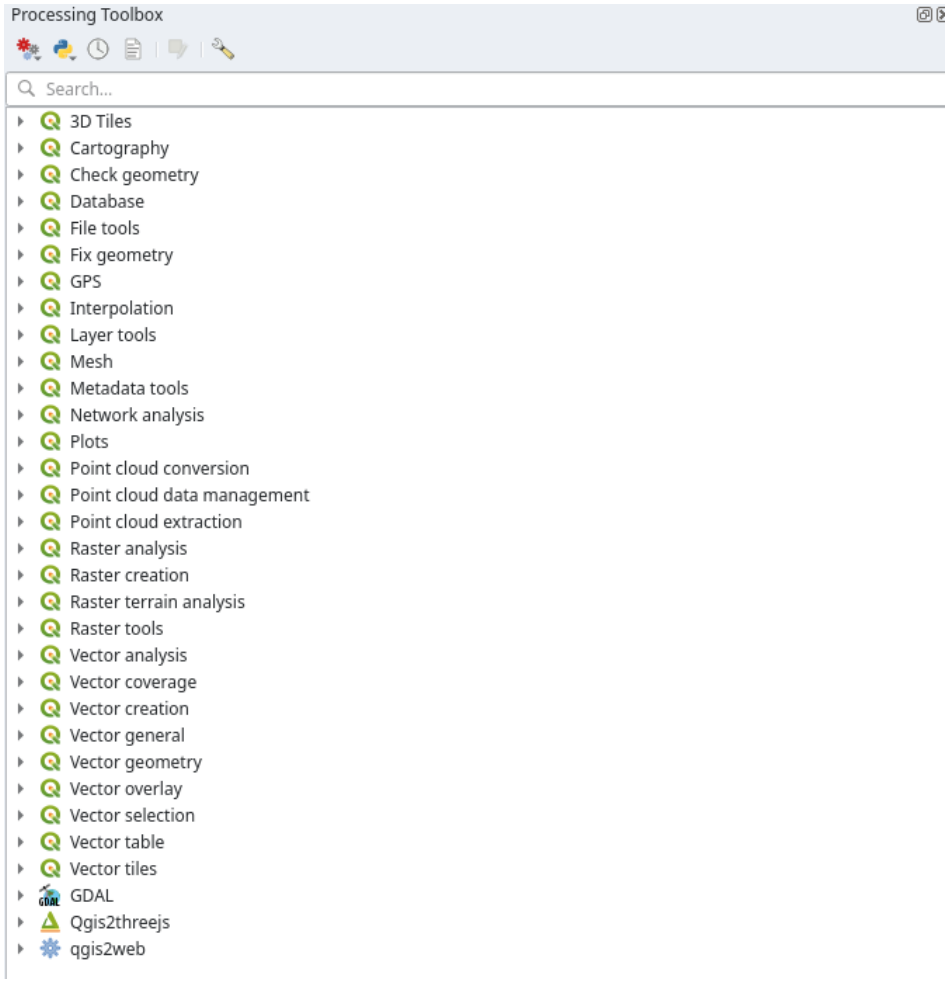
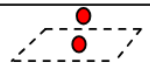


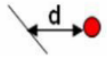
Figura: Exemplo: Ferramentas de geoprocessamento no QGIS (processing toolbox).

Relações espaciais: Pontos

Relação entre pontos

PONTO / PONTO	
Disjunto	
Perto de	
Coincidente	
Acima/Abaixo	
Em frente a	

Relação entre ponto e linha

Ponto/Linha	
Disjunto	
Toca/Adjacente	
Perto de	
Sobre	
Acima/Abaixo	

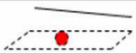
Relação entre ponto e polígono

PONTO / POLÍGONO	
Disjunto	
Adjacente/Toca	
Perto de	
Dentro de	
Acima/Abaixo	
Em frente a	

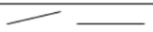
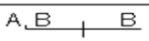
Fonte: ET-EDGV 3.0 (link)

Relações espaciais: Linhas

Relação entre ponto e linha

Ponto/Linha	
Disjunto	
Toca/Adjacente	
Perto de	
Sobre	
Acima/Abaixo	

Relação entre linhas

LINHA / LINHA	
Disjunto	
Toca	
Cruza	
Coincidente	
Acima/Abaixo	
Adjacente	
Perto de	
Entre	
Paralelo a	
Sobre	

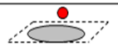
Relação entre linha e polígono

LINHA / POLÍGONO	
Disjunto	
Adjacente	
Perto de	
Dentro	
Acima/Abaixo	
Cruza	
Atravessa	
Em frente a	
Toca	

Fonte: ET-EDGV 3.0 (link)

Relações espaciais: Polígonos

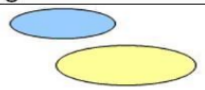
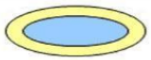
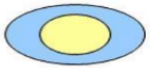
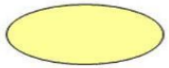
Relação entre ponto e polígono

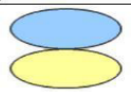
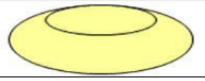
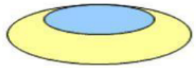
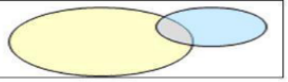
PONTO / POLÍGONO	
Disjunto	
Adjacente/Toca	
Perto de	
Dentro de	
Acima/Abaixo	
Em frente a	

Relação entre linha e polígono

LINHA / POLÍGONO	
Disjunto	
Adjacente	
Perto de	
Dentro	
Acima/Abaixo	
Cruza	
Atravessa	
Em frente a	
Toca	

Relação entre polígonos:

Polígono/Polígno	
Disjunto	
Contém	
Dentro	
Igual	

Polígono/Polígno	
Encontram	
Cobre	
Coberto por	
Sobreposição	

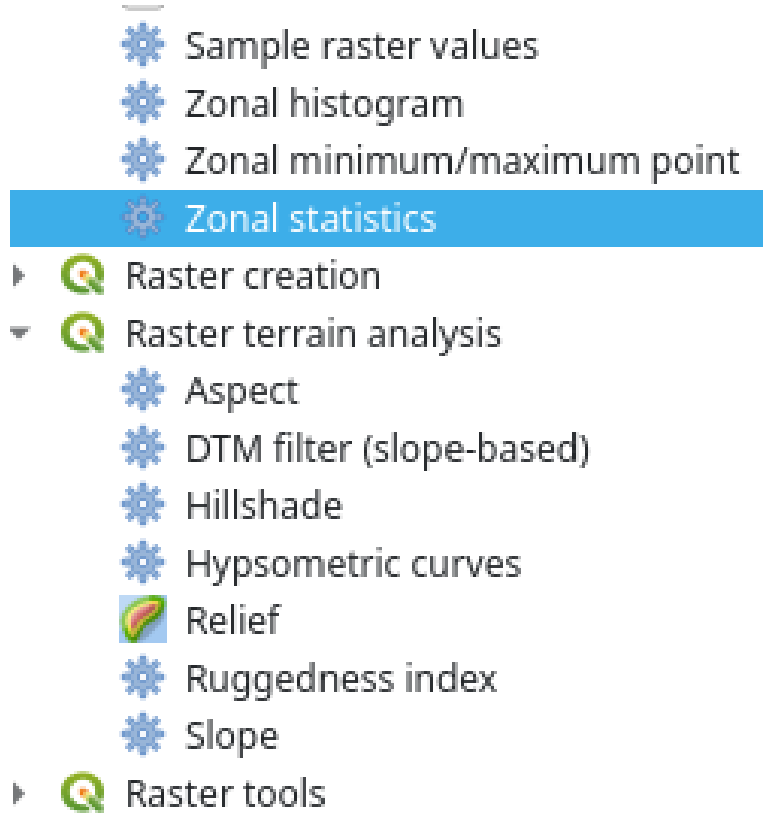
Fonte: ET-EDGV 3.0 (link)

Simple feature access

- O padrão determina os tipos de geometrias e as operações entre elas.
- As operações são definidas principalmente para SQL (bancos relacionais).
- Acessem em https://www.ogc.org/standards/sfs/
- OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 2: SQL option

Operações entre dados matriciais e vetoriais

Existem várias operações entre dados matriciais e vetoriais.



Execução de módulos do processing em Python

```
from qgis import processing
result = processing.run(
"native:buffer",
{
  'INPUT': layer,
  'OUTPUT': 'memory:'
},
context,
feedback
)
QgsProject.instance().addMapLayer(result['OUTPUT'])
```

```
import processing
processing.run(
"native:extractbyextent",
{
  'INPUT':'C:/Users/MDT-NOT/Desktop/Programação Aplicada/Aulas/poli
  'EXTENT':'-54.832642739,-54.570154142,-29.489490136,-29.270209539
  'CLIP':True,
  'OUTPUT':'TEMPORARY_OUTPUT'
```

Consultas Espaciais no Processing: Seleção

content...

Consultas Espaciais no Processing: Extração

content...

Consultas Espaciais no PyQGIS

```
# polygon_geometry contains a complex polygon, with many vertices
polygon_geometry = QgsGeometry.fromWkt('Polygon((...))')


# now we are ready to quickly test intersection against many other
for feature in my_layer.getFeatures():
  feature_geometry = feature.geometry()
  # test whether the feature's geometry intersects our original con
  if polygon_geometry.intersects(feature_geometry):
    print('feature intersects the polygon!')
```

Consultas Espaciais no PyQGIS

Geometry Engine:

```
# polygon_geometry contains a complex polygon, with many vertices
polygon_geometry = QgsGeometry.fromWkt('Polygon((...))')

# create a QgsGeometryEngine representation of the polygon
polygon_geometry_engine = QgsGeometry.createGeometryEngine(polygon_

# since we'll be performing many intersects tests, we can speed up
# by first "preparing" the geometry engine
polygon_geometry_engine.prepareGeometry()
# now we are ready to quickly test intersection against many other
for feature in my_layer.getFeatures():
feature_geometry = feature.geometry()
# test whether the feature's geometry intersects our original compl
if polygon_geometry_engine.intersects(feature_geometry.constGet()):
print('feature intersects the polygon!')
```

Índices Espaciais (Spatial Index)

teste

Índices Espaciais: Criação no PyQGIS

```
processing.run(
"native:createspatialindex",
{'INPUT':inputLyr},
feedback=feedback,
context=context,
is_child_algorithm=is_child_algorithm
)
```

Índices Espaciais: Criação no PyQGIS

```
spatialIdx = QgsSpatialIndex()
idDict = {}
for feature in layer.getFeatures():
idDict[feature.id()] = feature
spatialIdx.addFeature(feature)
```

Índices Espaciais: Uso de índice espacial no PyQGIS

```
for featA in layerA.getFeatures():
geomA = featA.geometry()
bbox = geomA.boundingBox()
for featB in layerB.getFeatures(bbox): # usa o índice espacial para
geomB = featB.geometry()
if geomB.intersects(geomA):
print(f"Feicoes {featA.id()} e {featB.id()} se intersectam")

layer_B_id_dict = {}
layerB_spatial_idx = QgsSpatialIndex()
for featB in layerB.getFeatures():
layer_B_id_dict[featB.id()] = featB
layerB_spatial_idx.addFeature(featB)

for featA in layerA.getFeatures():
geomA = featA.geometry()
bbox = geomA.boundingBox()
for id in layerB_spatial_idx.intersects(bbox):
featB = layer_B_id_dict[id]
```

Índices Espaciais: Uso de índice espacial no PyQGIS

1. Buffer
2. Intersecção
3. União
4. Diferença Simétrica
5. Diferença

Exercícios

- Processamento de dados vetoriais (Link)
- Processamento de dados matriciais (Link)