

## Join espacial

Em um *join* espacial, dois objetos de geometria ( *geometry* ) são combinados com base na relação espacial existente entre eles.

O parâmetro *op* , que utilizamos em nossa última aula, determina que tipo de operação será aplicada para fazer a junção dos dados. Este parâmetro pode assumir os seguintes valores: “intersects”, “within” ou “contains”.

Para entendermos melhor o funcionamento desta funcionalidade do GeoPandas considere o código abaixo e o resultado que ele produz:

In [1]:

```
import geopandas as gpd
from shapely.geometry import Point, Polygon

# GeoDataFrame representando um polígono
p = {'Ident': 'Polygon A',
     'geometry': Polygon([(0, 0), (1, 0), (1, 1), (0, 1)])}

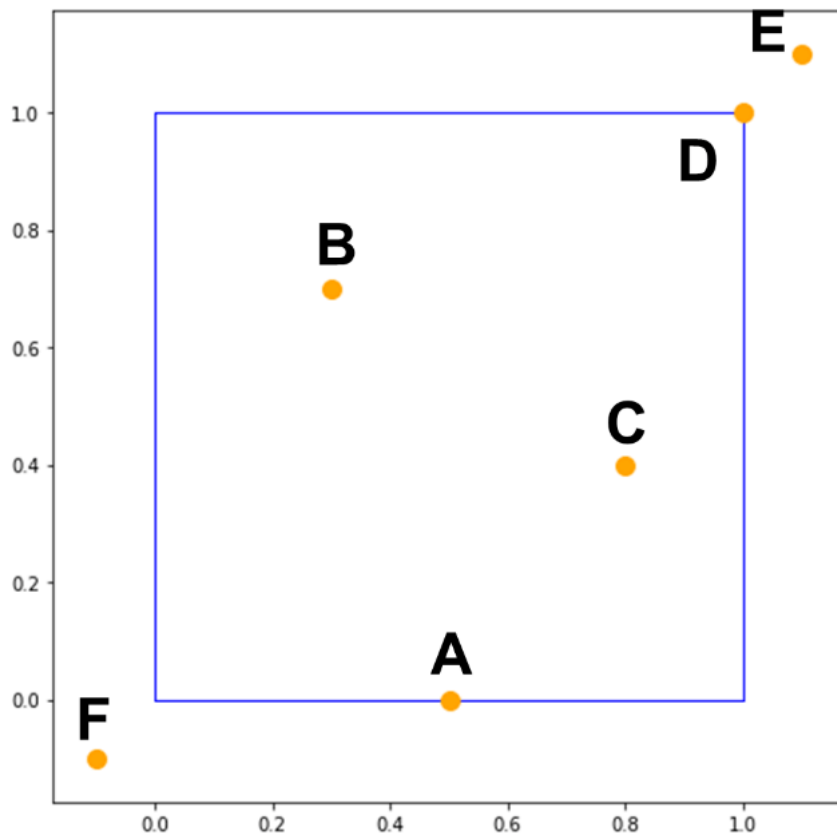
polygon = gpd.GeoDataFrame([p])

# GeoDataFrame representando pontos
pt = {'Ident': ['A', 'B', 'C', 'E', 'F', 'G'],
     'geometry': [Point(0.5, 0.0),
                  Point(0.3, 0.7),
                  Point(0.8, 0.4),
                  Point(1.0, 1.0),
                  Point(1.1, 1.1),
                  Point(-0.1, -0.1)]}

points = gpd.GeoDataFrame(pt)

# Plotagem dos dois GeoDataFrames
base = polygon.plot(color='white', edgecolor='blue', figsize=(15, 8))
points.plot(ax=base, color='orange', markersize=100)
```

Out [1]:



Utilizando os dois *GeoDataFrames* criados no código acima, assinale as opções corretas.

Observação: Utilize um notebook para executar os códigos deste exercício.

Dica: Utilize este exercício como ferramenta para entender melhor a mecânica por trás da função `sjoin()` do GeoPandas.

Selecione 3 alternativas

**A**

In [1]:

```
sjoin = gpd.sjoin(polygon, points, how='inner', op='contains')
sjoin
```

Out [1]:

	index_left	Ident_x	Ident_y	geometry
index_right				
1	0.0	Polygon A	B	POINT (0.3 0.7)
2	0.0	Polygon A	C	POINT (0.8 0.4)
0	NaN	NaN	A	POINT (0.5 0)
3	NaN	NaN	E	POINT (1 1)
4	NaN	NaN	F	POINT (1.1 1.1)
5	NaN	NaN	G	POINT (-0.1 -0.1)

**B**

In [1]:

```
sjoin = gpd.sjoin(points, polygon, how='inner', op='within')
sjoin
```

Out [1]:

	Ident_left	geometry	index_right	Ident_right
1	B	POINT (0.3 0.7)	0	Polygon A
2	C	POINT (0.8 0.4)	0	Polygon A

C

```
In [1]:
sjoin = gpd.sjoin(polygon, points, how='left', op='intersects')
sjoin
```

Out [1]:

	Ident_left	geometry	index_right	Ident_right
0	Polygon A	POLYGON ((0 0, 1 0, 1 1, 0 1, 0 0))	1	B
0	Polygon A	POLYGON ((0 0, 1 0, 1 1, 0 1, 0 0))	0	A
0	Polygon A	POLYGON ((0 0, 1 0, 1 1, 0 1, 0 0))	2	C
0	Polygon A	POLYGON ((0 0, 1 0, 1 1, 0 1, 0 0))	3	E

D

```
In [1]:
sjoin = gpd.sjoin(polygon, points, how='right', op='intersects')
sjoin
```

Out [1]:

	index_left	Ident_x	Ident_y	geometry
index_right				
1	0.0	Polygon A	B	POINT (0.3 0.7)
0	0.0	Polygon A	A	POINT (0.5 0)
2	0.0	Polygon A	C	POINT (0.8 0.4)
3	0.0	Polygon A	E	POINT (1 1)
4	NaN	NaN	F	POINT (1.1 1.1)
5	NaN	NaN	G	POINT (-0.1 -0.1)