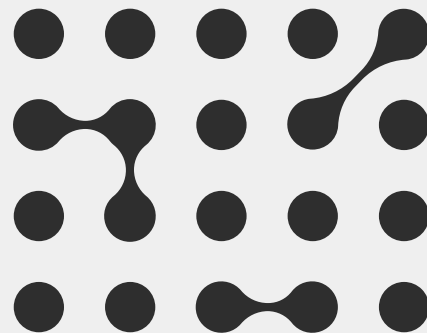


# Python Nordeste 2024

## Tutorial

Visualização e Processamento  
de Dados Geoespaciais:  
Uma Introdução Prática

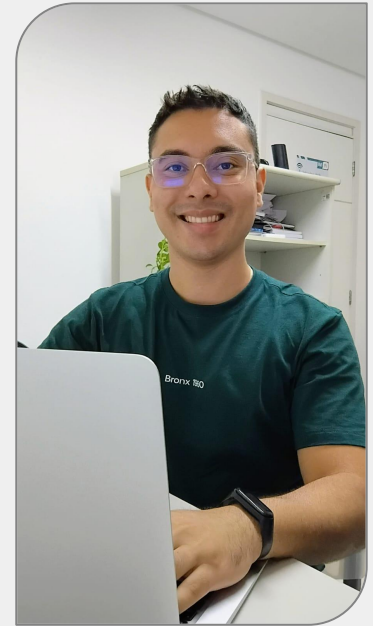
■ Pedro Florencio - Data Scientist @Ipplan



# Quem sou eu?

- **Graduando de Engenharia Elétrica** na Universidade Federal do Ceará (UFC);
- **Assessor Técnico** na área de **Ciência de Dados** no Instituto de Pesquisa e Planejamento de Fortaleza (Ipplan Fortaleza);
- **Pesquisador Voluntário** de graduação no Biomedical Data Analytics Research Group (BioData - UFC);

*Fortalezense, cicloativista, pseudomúsico, cinéfilo e colecionador de funks.*



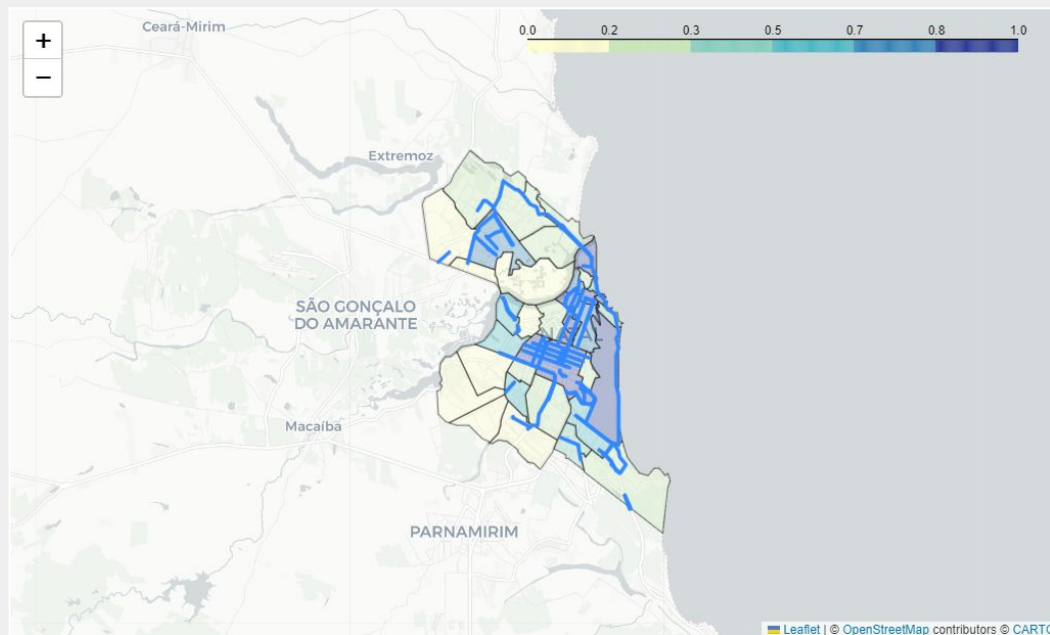
# Conteúdo Programático



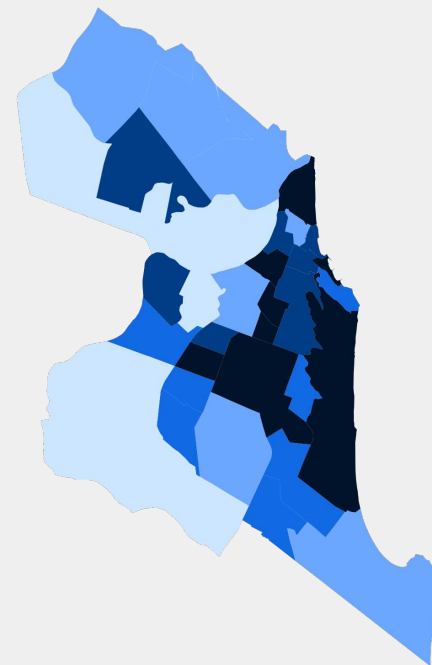
Ao final deste tutorial, você será capaz de:

1. Compreender o modelo e estrutura de dados geoespaciais;
2. Realizar processamento de dados com **Shapely**, **Pandas** e **GeoPandas**;
3. Produzir visualizações de mapas dos tipos:
  - a. Estática
    - i. **Matplotlib**
  - b. Interativa
    - i. **Plotly**
    - ii. **Folium** (Leaflet)
    - iii. **Datawrapper** (Plataforma *no-code*)
4. Entender o processo de uma análise de dados direcionada ao suporte de decisões de políticas públicas do seu município.

# Resultados esperados



**Figura 01:** Distribuição da Infraestrutura Ciclovária de Natal



**Figura 02:** *People Near Bike Lanes*

# Estrutura de Dados Geoespaciais



**Figura 03:** Tabela (objeto DataFrame)

	id	bairro
0	9	Rocas
1	11	Praia do Meio
2	34	Neópolis
3	35	Capim Macio
4	26	Nova Descoberta

"Everything is related to everything else, but near things are more related than distant things, as a consequence of accessibility."

**Waldo R. Tobler**

# Estrutura de Dados Geoespaciais



**Figura 04:** Tabela Geográfica

	id	bairro	geometry
0	9	Rocas	POLYGON ((256848.065 9361403.003, 256791.200 9...
1	11	Praia do Meio	POLYGON ((257073.088 9361374.537, 257074.693 9...
2	34	Neópolis	POLYGON ((256774.934 9350774.711, 256732.016 9...
3	35	Capim Macio	POLYGON ((256758.629 9353564.405, 256880.038 9...
4	26	Nova Descoberta	POLYGON ((257075.549 9355457.924, 257105.533 9...

"Everything is related to everything else, but near things are more related than distant things, as a consequence of accessibility."

**Waldo R. Tobler**

# Estrutura de Dados Geoespaciais



## Geometry types

[Point](#) (\*args)

A geometry type that represents a single coordinate with x,y and possibly z values.



Point

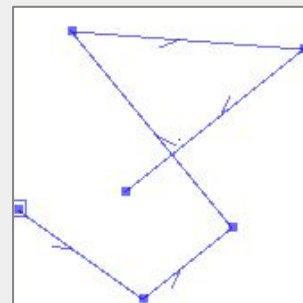
Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

# Estrutura de Dados Geoespaciais



## Geometry types

<code>Point</code> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<code>LineString</code> ([coordinates])	A geometry type composed of one or more line segments.



**LineString**

Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>



# Estrutura de Dados Geoespaciais



## Geometry types

<code>Point</code> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<code>LineString</code> ([coordinates])	A geometry type composed of one or more line segments.

Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

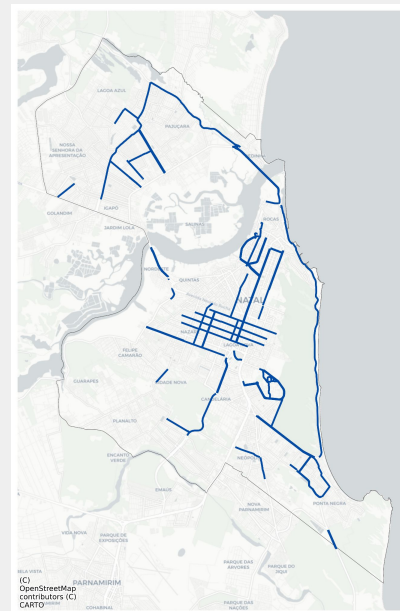


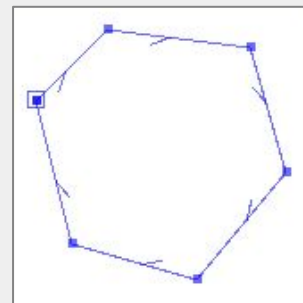
Figura 03: Malha Ciclovária de Natal

# Estrutura de Dados Geoespaciais



## Geometry types

<code>Point</code> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<code>LineString</code> ((coordinates))	A geometry type composed of one or more line segments.
<code>LinearRing</code> ((coordinates))	A geometry type composed of one or more line segments that forms a closed loop.



LinearRing

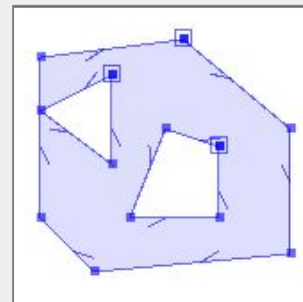
Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

# Estrutura de Dados Geoespaciais



## Geometry types

<a href="#">Point</a> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<a href="#">LineString</a> ([coordinates])	A geometry type composed of one or more line segments.
<a href="#">LinearRing</a> ([coordinates])	A geometry type composed of one or more line segments that forms a closed loop.
<a href="#">Polygon</a> ([shell, holes])	A geometry type representing an area that is enclosed by a linear ring.



Polygon

Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

# Estrutura de Dados Geoespaciais



## Geometry types

<a href="#">Point</a> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<a href="#">LineString</a> ((coordinates))	A geometry type composed of one or more line segments.
<a href="#">LinearRing</a> ((coordinates))	A geometry type composed of one or more line segments that forms a closed loop.
<a href="#">Polygon</a> ((shell, holes))	A geometry type representing an area that is enclosed by a linear ring.

Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

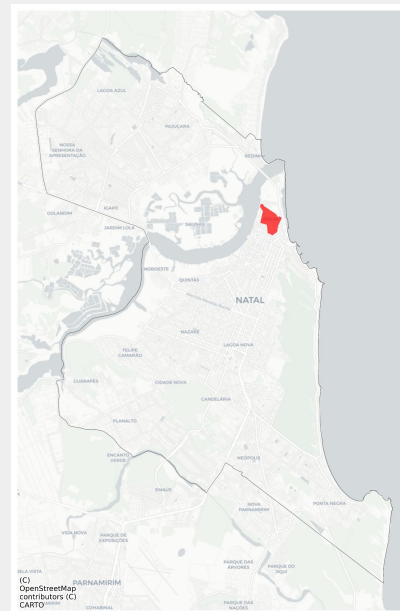


Figura 04: Bairro Rocas

# Estrutura de Dados Geoespaciais



## Geometry types

<a href="#">Point</a> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<a href="#">LineString</a> ([coordinates])	A geometry type composed of one or more line segments.
<a href="#">LinearRing</a> ([coordinates])	A geometry type composed of one or more line segments that forms a closed loop.
<a href="#">Polygon</a> ([shell, holes])	A geometry type representing an area that is enclosed by a linear ring.
<a href="#">MultiPoint</a> ([points])	A collection of one or more Points.



**MultiPoint**

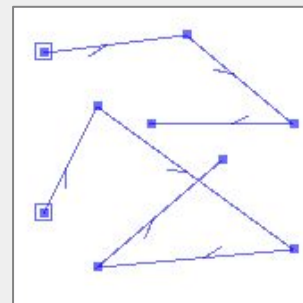
Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

# Estrutura de Dados Geoespaciais



## Geometry types

<a href="#">Point</a> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<a href="#">LineString</a> ((coordinates))	A geometry type composed of one or more line segments.
<a href="#">LinearRing</a> ((coordinates))	A geometry type composed of one or more line segments that forms a closed loop.
<a href="#">Polygon</a> ((shell, holes))	A geometry type representing an area that is enclosed by a linear ring.
<a href="#">MultiPoint</a> ((points))	A collection of one or more Points.
<a href="#">MultiLineString</a> ((lines))	A collection of one or more LineStrings.



**MultiLineString**

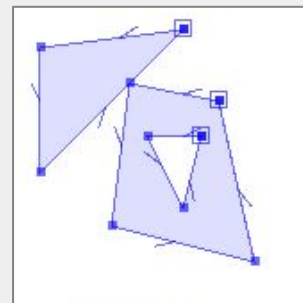
Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

# Estrutura de Dados Geoespaciais



## Geometry types

<a href="#">Point</a> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<a href="#">LineString</a> ((coordinates))	A geometry type composed of one or more line segments.
<a href="#">LinearRing</a> ((coordinates))	A geometry type composed of one or more line segments that forms a closed loop.
<a href="#">Polygon</a> ((shell, holes))	A geometry type representing an area that is enclosed by a linear ring.
<a href="#">MultiPoint</a> ([points])	A collection of one or more Points.
<a href="#">MultiLineString</a> ([lines])	A collection of one or more LineStrings.
<a href="#">MultiPolygon</a> ([polygons])	A collection of one or more Polygons.



**MultiPolygon**

Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

# Estrutura de Dados Geoespaciais



## Geometry types

<a href="#">Point</a> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<a href="#">LineString</a> ([coordinates])	A geometry type composed of one or more line segments.
<a href="#">LinearRing</a> ([coordinates])	A geometry type composed of one or more line segments that forms a closed loop.
<a href="#">Polygon</a> ([shell, holes])	A geometry type representing an area that is enclosed by a linear ring.
<a href="#">MultiPoint</a> ([points])	A collection of one or more Points.
<a href="#">MultiLineString</a> ([lines])	A collection of one or more LineStrings.
<a href="#">MultiPolygon</a> ([polygons])	A collection of one or more Polygons.

Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

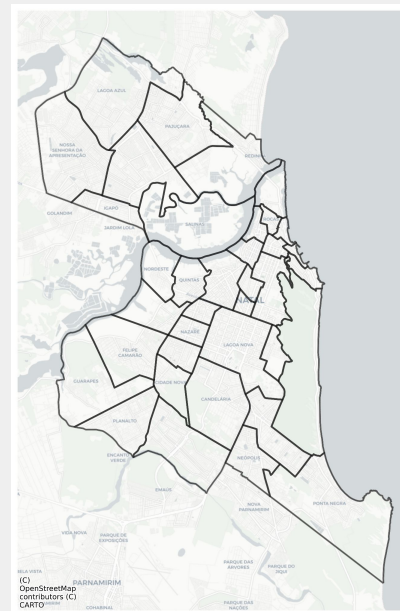


Figura 05: Bairros de Natal

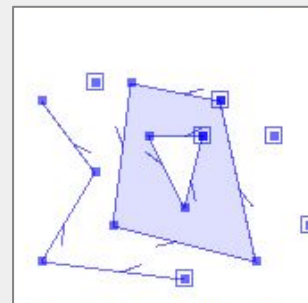


# Estrutura de Dados Geoespaciais



## Geometry types

<a href="#">Point</a> (*args)	A geometry type that represents a single coordinate with x,y and possibly z values.
<a href="#">LineString</a> ((coordinates))	A geometry type composed of one or more line segments.
<a href="#">LinearRing</a> ((coordinates))	A geometry type composed of one or more line segments that forms a closed loop.
<a href="#">Polygon</a> ((shell, holes))	A geometry type representing an area that is enclosed by a linear ring.
<a href="#">MultiPoint</a> ((points))	A collection of one or more Points.
<a href="#">MultiLineString</a> ((lines))	A collection of one or more LineStrings.
<a href="#">MultiPolygon</a> ((polygons))	A collection of one or more Polygons.
<a href="#">GeometryCollection</a> ((geoms))	A collection of one or more geometries that may contain more than one type of geometry.



GeometryCollection

Fonte: <https://shapely.readthedocs.io/en/stable/geometry.html>

# Estrutura de Dados Geoespaciais



**Tabela 01:** Tipos de arquivos de dados geoespaciais

Formato	Descrição
<b>Shapefile (.shp)</b>	Conjunto de arquivos para armazenar dados geoespaciais, amplamente utilizado. Inclui .shp (geometria), .shx (índice) e .dbf (atributos).
<b>GeoJSON</b>	Formato baseado em JSON para representar dados vetoriais, como pontos, linhas e polígonos.
<b>KML</b>	Formato XML desenvolvido para o Google Earth, usado para representar dados como placemarks, polígonos e camadas de imagens.
<b>GPKG (GeoPackage)</b>	Formato baseado em SQLite para armazenamento de dados vetoriais e raster, alternativa ao Shapefile.

# Estrutura de Dados Geoespaciais

## Sistemas de Referência de Coordenadas (SRC)



Um sistema de georreferenciamento de coordenadas, também conhecido como Sistema de Referência de Coordenadas (CRS - Coordinate Reference System), é um sistema que utiliza **coordenadas** (geralmente latitude e longitude) para **identificar** e **localizar** pontos na superfície da Terra. Esses sistemas podem ser baseados em **modelos diferentes da Terra** (esferoides ou elipsóides) e podem ser projetados para diversas finalidades, como mapeamento global, regional ou local.

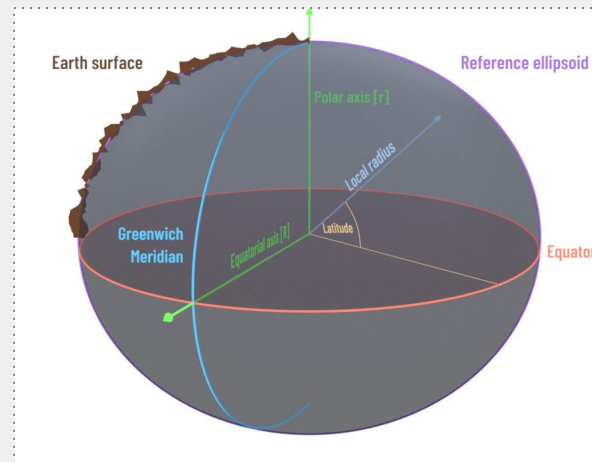
Leitura recomendada: Chang, Kang-Tsung (2006). **Introduction to Geographic Information Systems**. 3rd Edition. McGraw Hill.

# Estrutura de Dados Geoespaciais

## Tipos de Sistemas de Coordenadas



**Geodésico:** Utiliza a superfície do **elipsoide** para representar a Terra. As coordenadas são expressas em latitude, longitude e, às vezes, altitude. Um exemplo comum é o WGS 84, que é utilizado pelo GPS.



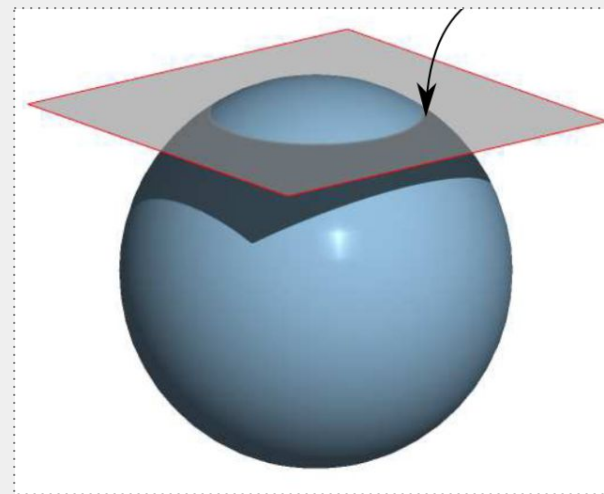
Leitura recomendada: Chang, Kang-Tsung (2006). **Introduction to Geographic Information Systems**. 3rd Edition. McGraw Hill.

# Estrutura de Dados Geoespaciais

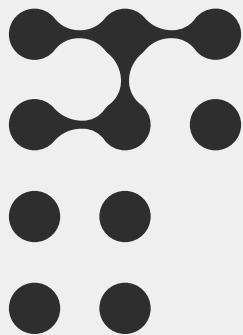
## Tipos de Sistemas de Coordenadas



**Projetado:** Transforma coordenadas esféricas (latitude e longitude) em um **sistema cartesiano** (x, y), permitindo que a superfície curva da Terra seja representada em uma **superfície plana**, como em mapas. Exemplos incluem UTM (Universal Transverse Mercator) e projeções como a de Mercator, Lambert, entre outras.



Leitura recomendada: Chang, Kang-Tsung (2006). **Introduction to Geographic Information Systems**. 3rd Edition. McGraw Hill.



**Hands-On:** Onde investir  
em infraestrutura cicloviária  
na minha cidade?

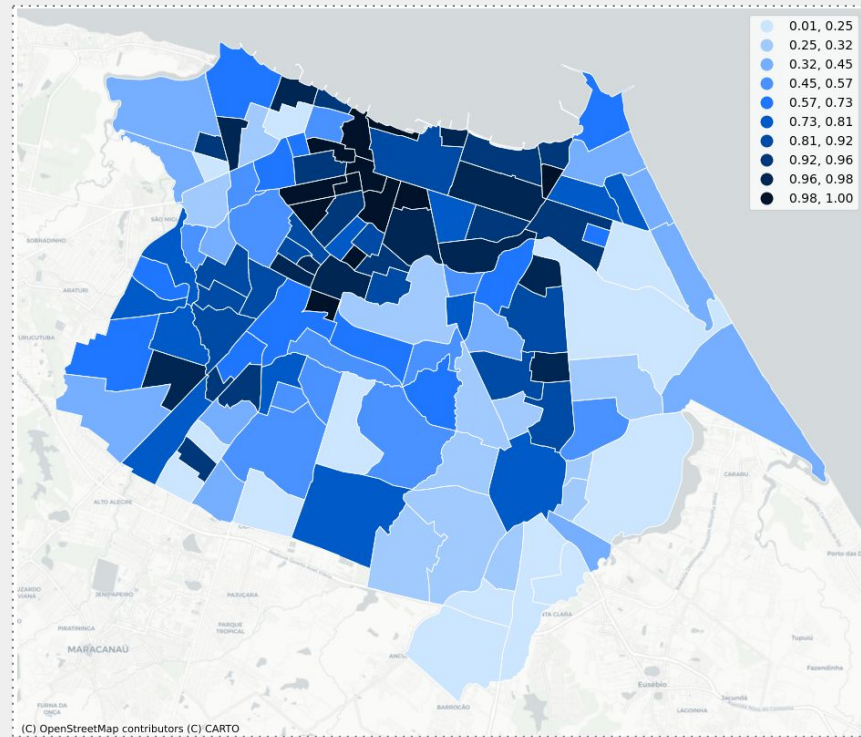
**Estudo de Caso**

# People Near Bike Lanes (PNB)

## *Acesso à Infraestrutura Ciclovitária*

> Desenvolvido pelo **Institute for Transportation and Development Policy (ITDP)**, foi criado para medir a acessibilidade das infraestruturas de ciclovias e a proximidade da população a essas vias.

> O **PNB** é definido matematicamente como a proporção da população que vive a pelo menos **300 metros de uma infraestrutura ciclovitária**, sendo um valor de varia de 0 a 1, isto é, de 0 a 100% da população.



**Figura 06:** Proximidade a Infraestrutura Ciclovitária nos Bairros de Fortaleza



# Passo-a-Passo



## Aquisição de Dados

Leitura dos dados e  
formação do objeto  
*GeoDataFrame*

01

03

## E.D.A.

Análise Exploratória dos  
Dados com *Pandas*,  
*Matplotlib* e *Seaborn*

## Pré-Processamento

Limpeza e preparação  
dos dados usando  
*Shapely* e *GeoPandas*

02

04

## Visualização

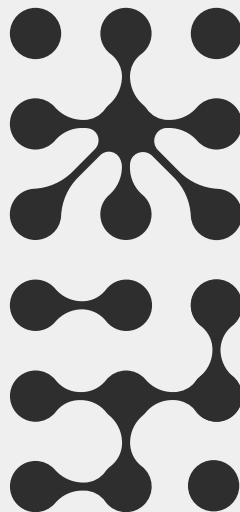
Composição dos mapas  
como produto



# 01

## Aquisição de Dados

Leitura dos dados e formação dos  
objetos *GeoDataFrame* ■



# Bibliotecas



```
1 import folium
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import folium.plugins
6 import geopandas as gpd
7 import contextily as cx
8 import plotly.express as px
9 from google.colab import drive
10 import matplotlib.pyplot as plt
11 from shapely.geometry import Point
12 import matplotlib.colors as mcolors
13
14 drive.mount('/gdrive', force_remount=True)
15
16 %matplotlib inline
```



**Folium**

Biblioteca que realiza o mapeamento do pacote *Leaflet.JS* para o Python.

# Bibliotecas



```
1 import folium
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import folium.plugins
6 import geopandas as gpd
7 import contextily as cx
8 import plotly.express as px
9 from google.colab import drive
10 import matplotlib.pyplot as plt
11 from shapely.geometry import Point
12 import matplotlib.colors as mcolors
13
14 drive.mount('/gdrive', force_remount=True)
15
16 %matplotlib inline
```



Biblioteca responsável  
por funções  
matemáticas, geração de  
números aleatórios,  
álgebra linear,  
Transformada de Fourier,  
entre outros.

# Bibliotecas



```
1 import folium
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import folium.plugins
6 import geopandas as gpd
7 import contextily as cx
8 import plotly.express as px
9 from google.colab import drive
10 import matplotlib.pyplot as plt
11 from shapely.geometry import Point
12 import matplotlib.colors as mcolors
13
14 drive.mount('/gdrive', force_remount=True)
15
16 %matplotlib inline
```



Pandas

Biblioteca que realiza a manipulação e permite a análise de dados estruturados.

# Bibliotecas



```
1 import folium
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import folium.plugins
6 import geopandas as gpd
7 import contextily as cx
8 import plotly.express as px
9 from google.colab import drive
10 import matplotlib.pyplot as plt
11 from shapely.geometry import Point
12 import matplotlib.colors as mcolors
13
14 drive.mount('/gdrive', force_remount=True)
15
16 %matplotlib inline
```



**Seaborn**

Biblioteca de visualização  
de dados baseada na  
Matplotlib.

# Bibliotecas



```
1 import folium
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import folium.plugins
6 import geopandas as gpd
7 import contextily as cx
8 import plotly.express as px
9 from google.colab import drive
10 import matplotlib.pyplot as plt
11 from shapely.geometry import Point
12 import matplotlib.colors as mcolors
13
14 drive.mount('/gdrive', force_remount=True)
15
16 %matplotlib inline
```



**Geopandas**

Biblioteca de  
manipulação de dados  
geoespaciais, que se  
baseia nos métodos do  
**Pandas** e faz uso de  
métodos do **Shapely** e  
**Matplotlib**.

# Bibliotecas



```
1 import folium
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import folium.plugins
6 import geopandas as gpd
7 import contextily as cx
8 import plotly.express as px
9 from google.colab import drive
10 import matplotlib.pyplot as plt
11 from shapely.geometry import Point
12 import matplotlib.colors as mcolors
13
14 drive.mount('/gdrive', force_remount=True)
15
16 %matplotlib inline
```

## Contextily

Pacote Python que permite a inserção de *tilemaps* (mapas de textura) como uma camada base nas visualizações.

# Bibliotecas



```
1 import folium
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import folium.plugins
6 import geopandas as gpd
7 import contextily as cx
8 import plotly.express as px
9 from google.colab import drive
10 import matplotlib.pyplot as plt
11 from shapely.geometry import Point
12 import matplotlib.colors as mcolors
13
14 drive.mount('/gdrive', force_remount=True)
15
16 %matplotlib inline
```



Biblioteca de plotagem interativa que permite a visualização de gráficos e mapas. O módulo `express` permite criar figuras inteiras de maneira prática.



# Bibliotecas



```
1 import folium
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import folium.plugins
6 import geopandas as gpd
7 import contextily as cx
8 import plotly.express as px
9 from google.colab import drive
10 import matplotlib.pyplot as plt
11 from shapely.geometry import Point
12 import matplotlib.colors as mcolors
13
14 drive.mount('/gdrive', force_remount=True)
15
16 %matplotlib inline
```



**Matplotlib**

Biblioteca de plotagem  
estática, interativa e  
visualizações animadas  
no Python. Além disso ela  
permite a visualização de  
**mapas**.

# Bibliotecas



```
1 import folium
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
5 import folium.plugins
6 import geopandas as gpd
7 import contextily as cx
8 import plotly.express as px
9 from google.colab import drive
10 import matplotlib.pyplot as plt
11 from shapely.geometry import Point
12 import matplotlib.colors as mcolors
13
14 drive.mount('/gdrive', force_remount=True)
15
16 %matplotlib inline
```

## Shapely

Pacote para análise teórica e manipulação de modelos geoespaciais.

# Acesso aos Datasets

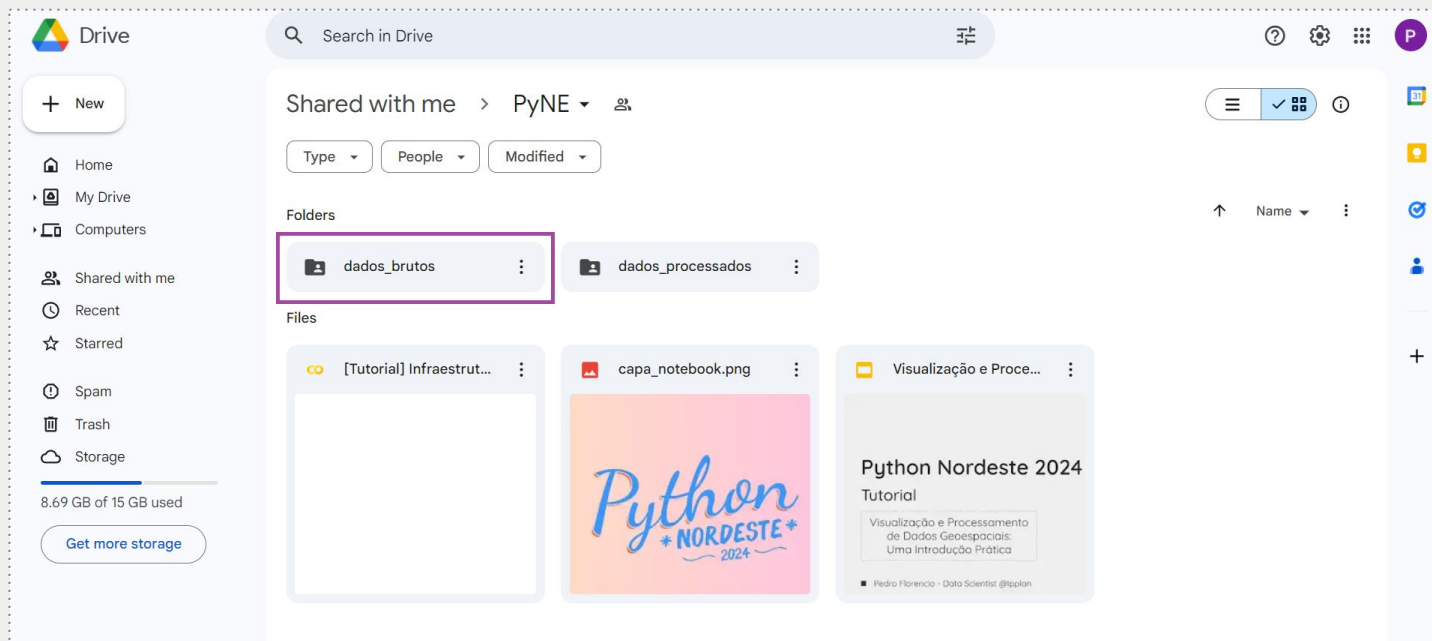



Figura 07: Local de armazenamento dos datasets no Google Drive

# Acesso aos Datasets



Name	↑
	bairros_natal



**bairros\_natal**

*Shapefile* (.shp) dos  
polígonos dos bairros do  
município.

**Figura 08:** Conjuntos de dados necessários para o cálculo.

# Acesso aos Datasets



Name	↑
	bairros_natal
	limites_municipio



## **limites\_municipio**

*Limites municipais de Natal  
no formato .shp*

**Figura 08:** Conjuntos de dados necessários para o cálculo.

# Acesso aos Datasets



Name	↑
	bairros_natal
	limites_municipio
	agregados_preliminares_por_setores_censitarios_RN.csv 

## agregados\_preliminares

*Dados sobre os setores  
censitários do Rio Grande  
do Norte*

**Figura 08:** Conjuntos de dados necessários para o cálculo.

# Acesso aos Datasets



Name	↑
	bairros_natal
	limites_municipio
	agregados_preliminares_por_setores_censitarios_RN.csv 
	Dicionario_CNEFE_Censo_2022.xls 
	infraestrutura_ciclovitaria_natal.geojson 

## infraestrutura\_ciclovitaria

*Dados sobre a  
infraestrutura ciclovária de  
Natal*

**Figura 08:** Conjuntos de dados necessários para o cálculo.

# Acesso aos Datasets



Name	↑
	bairros_natal
	limites_municipio
	agregados_preliminares_por_setores_censitarios_RN.csv 
	Dicionario_CNEFE_Censo_2022.xls 
	infraestrutura_ciclovitaria_natal.geojson 
	RN_malha_preliminar_2022.gpkg 

**RN\_malha\_preliminar**  
*Geolocalização dos setores  
censitários do Rio Grande  
do Norte*

**Figura 08:** Conjuntos de dados necessários para o cálculo.



# Acesso aos Datasets



## Métodos



```
1 pandas.read_csv(...)  
2  
3 geopandas.read_file(...)
```

Name	↑
 bairros_natal	
 limites_municipio	
 agregados_preliminares_por_setores_censitarios_RN.csv	
 Dicionario_CNEFE_Censo_2022.xls	
 infraestrutura_ciclovitaria_natal.geojson	
 RN_malha_preliminar_2022.gpkg	

**Figura 08:** Conjuntos de dados necessários para o cálculo.

# Acesso aos Datasets

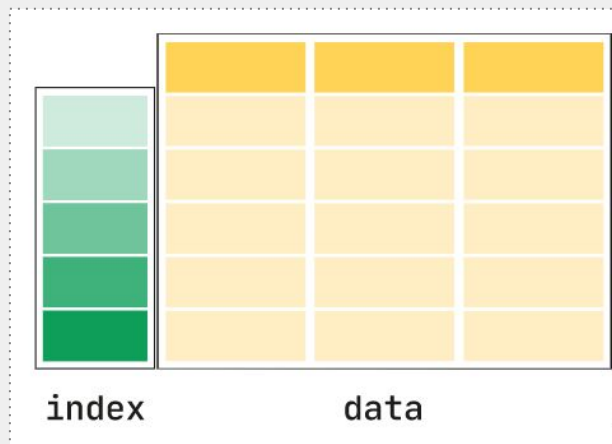


```
1 # caminho para os dados brutos
2 caminho_dados = '/gdrive/MyDrive/PyNE/dados_brutos/'
3
4 # setores censitarios do Rio Grande do Norte
5 df_setores = pd.read_csv(caminho_dados+'agregados_preliminares_por_setores_censitarios_RN.csv', sep=';')
6
7 # malha georreferenciada de setores censitarios
8 gdf_setores = gpd.read_file(caminho_dados+'RN_malha_preliminar_2022.gpkg', crs='EPSG:4326', driver='GPKG')
9
10 # limites dos bairros de Natal
11 gdf_bairros = gpd.read_file(caminho_dados+'bairros_natal/bairros.shp', crs='EPSG:4326')
12
13 # infraestrutura ciclovitaria
14 gdf_infraestrutura = gpd.read_file(caminho_dados+'infraestrutura_ciclovitaria_natal.geojson', crs='EPSG:4326')
```

# Acesso aos Datasets



**Figura 08:** Pandas DataFrame

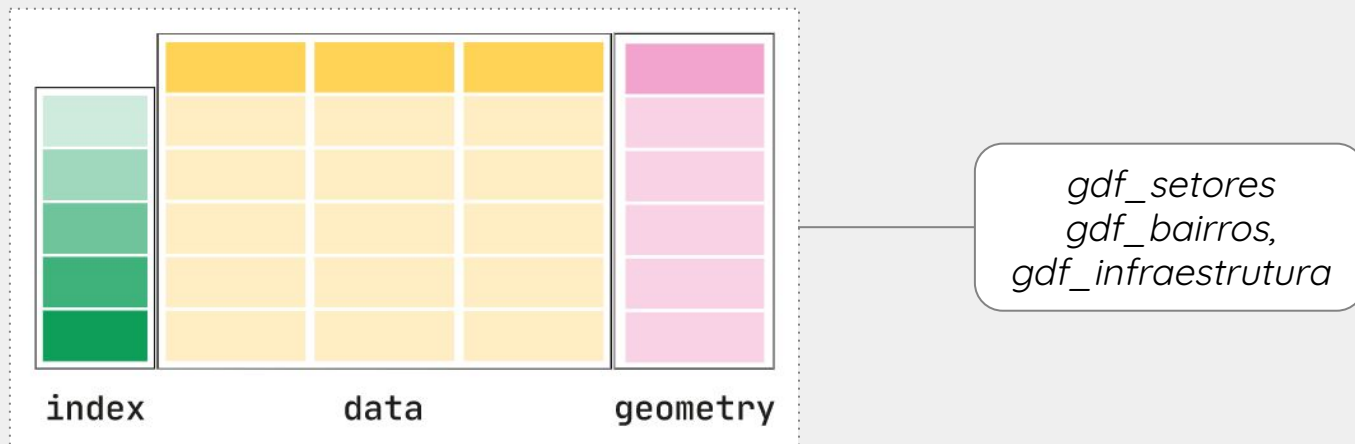


*df\_setores*

# Acesso aos Datasets



**Figura 09:** Geopandas GeoDataFrame



# Acesso aos Datasets



## Entendimento dos Dados

	Name	description	timestamp	begin	end	altitudeMode	tessellate	extrude	visibility	drawOrder
0	Ciclofaixa da Av. Pref. Omar O'Grady	> Administrada pelo Governo do Estado;  > Mão...	None	None	None	None	1	0	-1	None
1	Ciclofaixa da Via Costeira	> Administrada pelo Governo do Estado;  > Mão...	None	None	None	None	1	0	-1	None
2	Ciclovía Anel Viário da UFRN	> Administrada pela UFRN;  > Mão-única.	None	None	None	None	1	0	-1	None
3	Ciclovía do Anel Viário da UFRN	> Administrada pela UFRN;  > Mão-única.	None	None	None	None	1	0	-1	None
4	Ciclovía Interna da UFRN I	> Administrada pela UFRN;  > Mão dupla.	None	None	None	None	1	0	-1	None

## Métodos

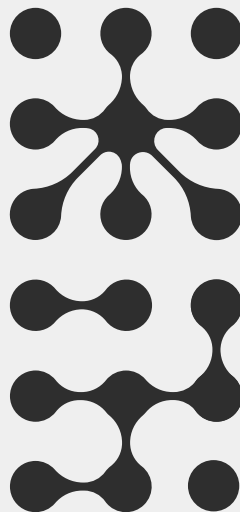


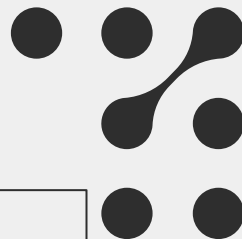
```
1 # para DataFrames
2 name_dataframe.head()
3 name_dataframe.info()
4
5 # para GeoDataFrames
6 name_dataframe.head()
7 name_geodataframe.info()
8 name_geodataframe.crs
```

# 02

## Pré-Processamento

Limpeza e preparação dos dados ■  
usando *Shapely* e *GeoPandas*





# Pré-Processamento de Dados

## Seleção de colunas



```
1 gdf = gdf[['coluna_1', 'coluna_2', ..., 'coluna_n']]
```

## Conversão de tipos



```
1 gdf.nome_coluna = gdf.nome_coluna.astype('nome_tipo').copy()
```

## Substituição de nomes



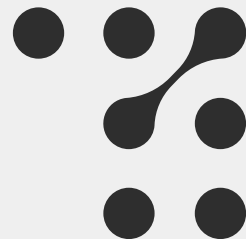
```
1 gdf = gdf.rename(mapper={'nome_atual': 'nome_novo',  
2                        'nome_atual': 'nome_novo'}, axis=1)
```

## Filtro de linhas



```
1 df = df[df['coluna'] == numero]
```

Com estes métodos você será capaz de realizar até o item 6.6



# Pré-Processamento de Dados

## Junção de tabelas

left				right				Result					
	key	A	B		key	C	D		key	A	B	C	D
0	K0	A0	B0	0	K0	C0	D0	0	K0	A0	B0	C0	D0
1	K1	A1	B1	1	K1	C1	D1	1	K1	A1	B1	C1	D1
2	K2	A2	B2	2	K2	C2	D2	2	K2	A2	B2	C2	D2
3	K3	A3	B3	3	K3	C3	D3	3	K3	A3	B3	C3	D3



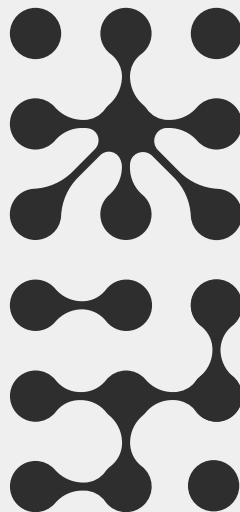
```
1 gdf = pd.merge(df_1, df_2[['key', 'geometry']], on='key', how='left')
```

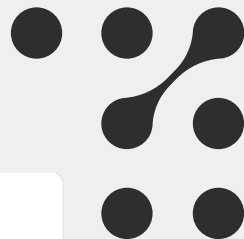


# 03

## E.D.A

Análise Exploratória dos Dados com ■  
*Pandas, Matplotlib e Seaborn*

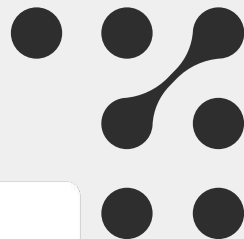




# Limites dos Bairros de Natal

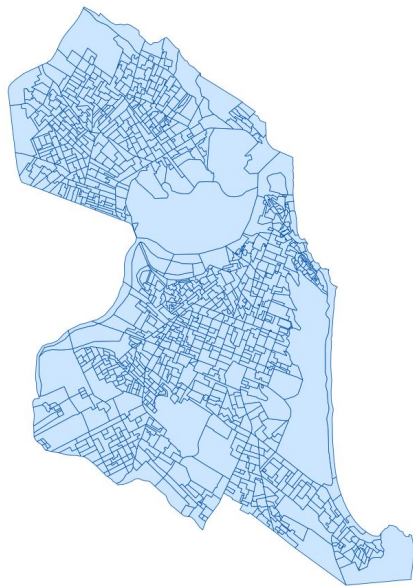


```
1 # parametros do mapa
2 titulo_do_mapa = 'Mapa 01: Limites dos Bairros de Natal'
3 resolucao_dpi = 'figure'
4 tamanho_do_mapa = (10,10)
5
6 # criando figura base
7 fig, ax = plt.subplots(figsize=tamanho_do_mapa)
8
9 # criando camada de limites dos bairros
10 gdf_bairros.boundary.plot(ax=ax, color='black', linewidth=.7, alpha=.5)
11
12 # configurando nome
13 plt.title(titulo_do_mapa, loc='left', fontsize=10)
14
15 # removendo eixos da visualizacao
16 plt.axis('off')
17
18 # adicionando cartografia base
19 cx.add_basemap(ax=ax, crs=gdf_bairros.crs, zoom=13, source='CartoDB positron')
20
21 plt.show()
```

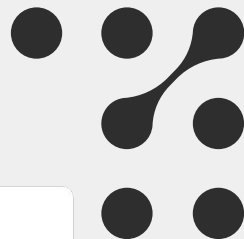


# Setores Censitários de Natal

Mapa 02: Distribuição dos Setores Censitários de Natal



```
1 titulo_do_mapa = 'Mapa 02: Distribuição dos Setores Censitários de Natal'
2 resolucao_dpi = 'figure'
3 tamanho_do_mapa = (12,12)
4
5 # criando figura base
6 fig, ax = plt.subplots(figsize=tamanho_do_mapa)
7
8 # malha de setores censitarios
9 gdf_setores_natal.plot(ax=ax, lw=.6, edgecolor='#004BA6', color='#CCE6FF')
10
11 # configurando nome
12 plt.title(titulo_do_mapa, loc='left', fontsize=10)
13
14 # removendo eixos da visualizacao
15 plt.axis('off')
16
17 plt.show()
```



# Infraestrutura Ciclovitária de Natal



```
1 titulo_do_mapa = 'Mapa 03: Distribuição da Infraestrutura Ciclovitária de Natal'
2 resolucao_dpi = 'figure'
3 tamanho_do_mapa = (12,12)
4
5 # criando figura base
6 fig, ax = plt.subplots(figsize=tamanho_do_mapa)
7
8 # adicionando camada de infraestrutura ciclovitaria
9 gdf_infraestrutura.plot(ax=ax, color='#004BA6')
10
11 # configurando nome
12 plt.title(titulo_do_mapa, loc='left', fontsize=10)
13
14 # removendo eixos da visualizacao
15 plt.axis('off')
16
17 cx.add_basemap(ax=ax, crs=gdf_infraestrutura.crs, zoom=13, source='CartoDB positron')
18
19 plt.show()
```



# Cálculo do PNB

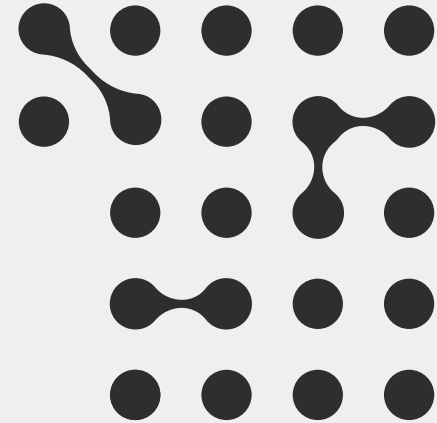


Vamos para o tópico 7.4 do notebook =)

# Obrigado!

Ficou com dúvidas?

*[pedroflorencio@alu.ufc.br](mailto:pedroflorencio@alu.ufc.br)*



 <https://www.linkedin.com/in/pedroflorencioneto/>

- CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**