Engenharia de Dados – Parte 01

Wellington Franco
Universidade Federal do Ceará – UFC
Campus da UFC em Crateús
wellington@crateus.ufc.br

Engenharia de Dados

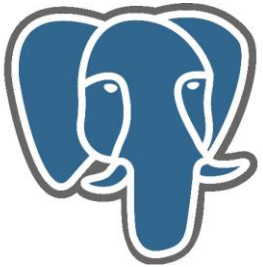
- Principais Atividades
  - Extração;
  - Tratamento;
  - Limpeza;
  - Manipulação de Dados;

Configurando o Ambiente

Ambiente de Desenvolvimento

Para a realização dos experimentos propostos nas seguintes aulas, é necessário ter instalado em sua máquina:

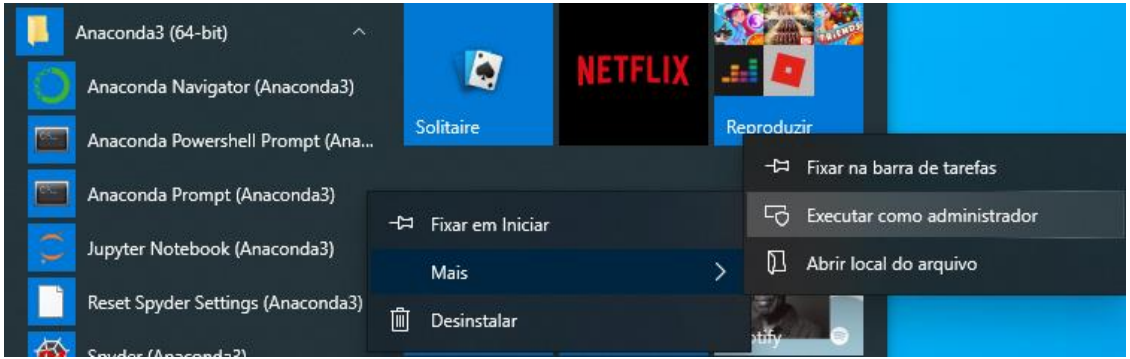
- **Anaconda:** https://www.anaconda.com/products/individual
  - Usaremos *Jupyter Notebook* como ambiente de desenvolvimento;
  - Utilizaremos as seguintes bibliotecas para desenvolvimento:
    - *NumPy, Pandas, Scrapy, PyPDF4, Spacy, Pdfminer, db-sqlite3*
- **PgAdmin 4:** https://www.pgadmin.org/download/
  - Usaremos a IDE do *PgAdmin 4* para fazer experimentos com banco de dados *plsql*.
- Colab



Ambiente de Desenvolvimento: DICAS

Para instalar qualquer **biblioteca** que iremos utilizar no **Jupyter Notebook**:

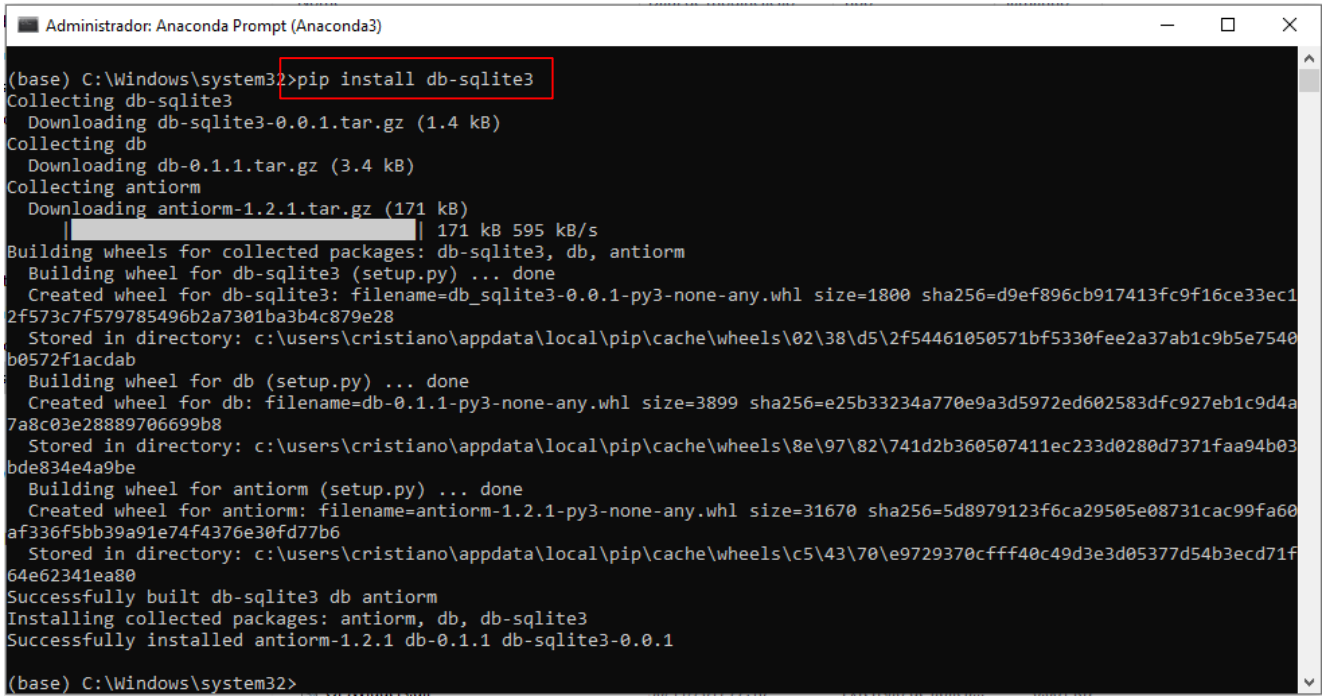
- No Windows:
  - Vá em INICIAR > Anaconda > clique com o botão direito em Anaconda *Prompt* e execute como administrador.



Ambiente de Desenvolvimento: DICAS

Com o *prompt* aberto, digite o comando sugerido pela documentação da biblioteca desejada.

Ex:



Colab

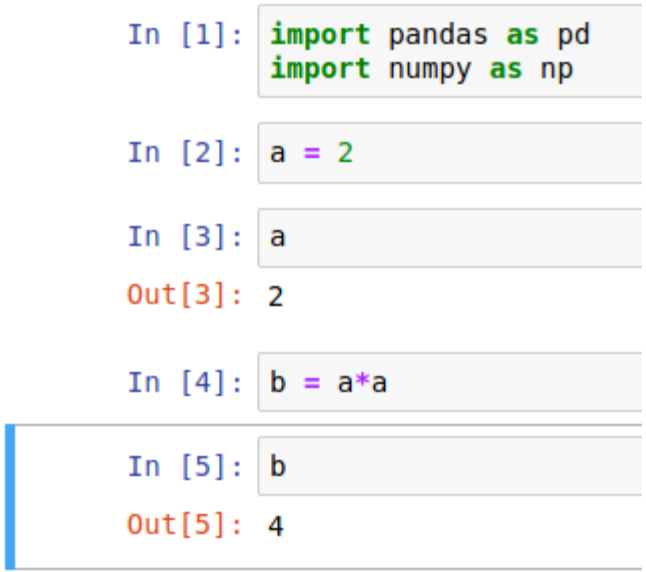
Definições Preliminares

Definindo Ambiente de Programação

Para realização de nossos experimentos, utilizaremos **Jupyter Notebook**.

Motivos:

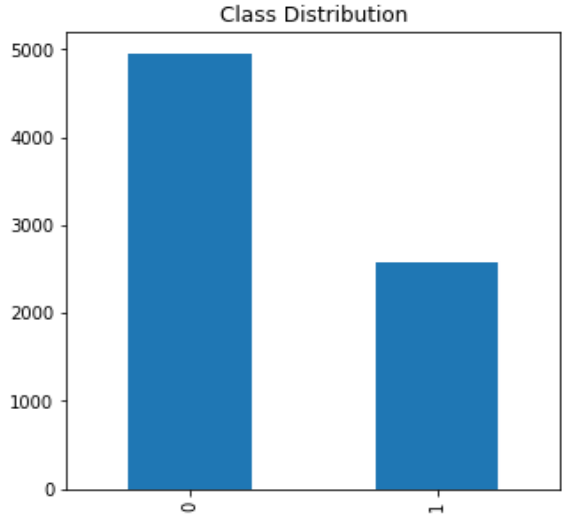
- *Python* 3;
- Organização;
- Mantém o *log* das execuções;
- Melhor controle do fluxo de trabalho;



Definindo Ambiente de Programação

- Permite uso de gráficos em células intermediárias

```
Out[4]: will_change
        0    4954
        1    2582
        Name: will_change, dtype: int64
```



Bibliotecas

- **Pandas**
  - ***Pan****el* ***da****ta****s*** (“dados em painel”);
  - Manipulação e análise de dados em Python;
  - Torna mais fácil manipulação de diferentes formatos de arquivo (*ex:* csv);
  - *Dataframe*.

Diferença entre usar lista e *dataframe*

```
In [1]: import pandas as pd

In [2]: # generating lists
        list_people = [['Gil', 35], ['Gal', 32], ['Zé', 45]]

In [3]: list_people
Out[3]: [['Gil', 35], ['Gal', 32], ['Zé', 45]]

In [4]: # generating dataframe
        list_people_df = pd.DataFrame(list_people)

In [5]: list_people_df
Out[5]:
```

	0	1
0	Gil	35
1	Gal	32
2	Zé	45

Para um grande volume de dados, utilizar lista é mais trabalhoso

Visualmente mais intuitivo!

```
In [6]: list_people_df.columns = ['Nome', 'Idade']

In [7]: list_people_df
Out[7]:
```

	Nome	Idade
0	Gil	35
1	Gal	32
2	Zé	45

Mais fácil de manipular dado

Bibliotecas

- **NumPy**
  - Manipulação algébrica de forma mais fácil:
    - Ordenar vetor, pegar o maior elemento, o menor elemento;
    - Função inversa, transposta, produto interno.
  - Realização de cálculo numérico em operações de *Machine Learning*;
  - “Facilidade” em manipulação de matrizes multidimensionais.

Obtendo valores estatísticos

```
In [7]: list_people_df
```

Out[7]:

	Nome	Idade
0	Gil	35
1	Gal	32
2	Zé	45

```
In [8]: list_people_df[['Idade']]
```

Out[8]:

	Idade
0	35
1	32
2	45

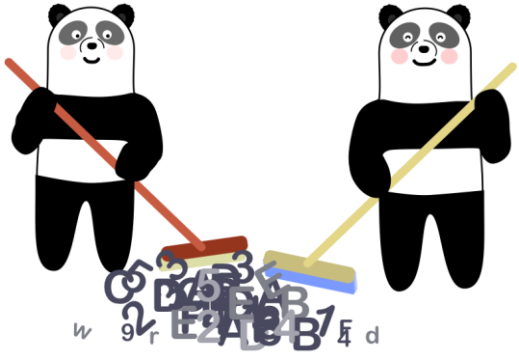
```
In [9]: import numpy as np
```

```
In [10]: mean = np.mean(list_people_df['Idade'])
```

usando a biblioteca *numpy* para obter a média das idades

```
In [11]: mean
```

```
Out[11]: 37.333333333333336
```



Manipulação e Limpeza de Dados