

Engenharia de Dados – Parte 01

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

Engenharia de Dados

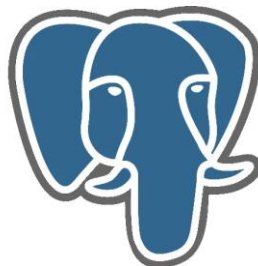
- Principais Atividades
 - Extração;
 - Tratamento;
 - Limpeza;
 - Manipulação de Dados;

Configurando o Ambiente

Ambiente de Desenvolvimento

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

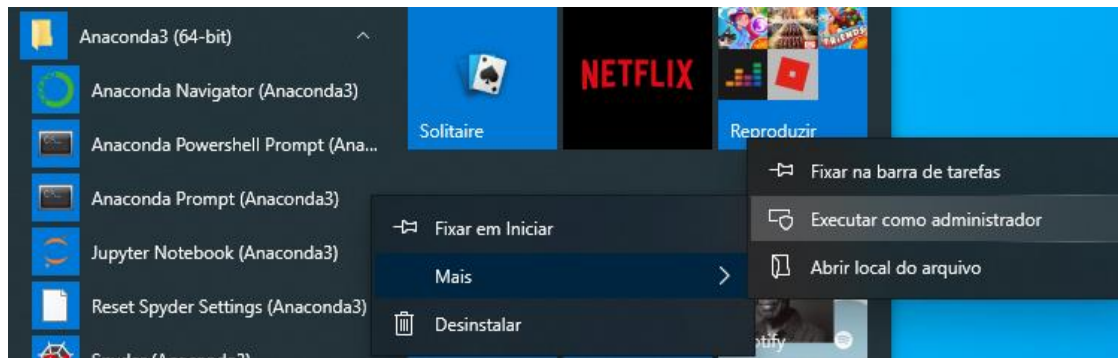
- **Anaconda:** <https://www.anaconda.com/products/individual>
 - Usaremos *Jupyter Notebook* como ambiente de desenvolvimento;
 - Utilizaremos as seguintes bibliotecas para desenvolvimento:
 - *NumPy, Pandas, Scrapy, PyPDF4, Spacy, Pdfminer, db-sqlite3*
- **PgAdmin 4:** <https://www.pgadmin.org/download/>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

```
Administrador: Anaconda Prompt (Anaconda3)

(base) C:\Windows\system32>pip install db-sqlite3
Collecting db-sqlite3
  Downloading db-sqlite3-0.0.1.tar.gz (1.4 kB)
Collecting db
  Downloading db-0.1.1.tar.gz (3.4 kB)
Collecting antiorm
  Downloading antiorm-1.2.1.tar.gz (171 kB)
    | 171 kB 595 kB/s
Building wheels for collected packages: db-sqlite3, db, antiorm
  Building wheel for db-sqlite3 (setup.py) ... done
  Created wheel for db-sqlite3: filename=db_sqlite3-0.0.1-py3-none-any.whl size=1800 sha256=d9ef896cb917413fc9f16ce33ec1
2f573c7f579785496b2a7301ba3b4c879e28
  Stored in directory: c:\users\cristiano\appdata\local\pip\cache\wheels\02\38\d5\2f54461050571bf5330fee2a37ab1c9b5e7540
b0572f1acdab
  Building wheel for db (setup.py) ... done
  Created wheel for db: filename=db-0.1.1-py3-none-any.whl size=3899 sha256=e25b33234a770e9a3d5972ed602583dfc927eb1c9d4a
7a8c03e28889706699b8
  Stored in directory: c:\users\cristiano\appdata\local\pip\cache\wheels\8e\97\82\741d2b360507411ec233d0280d7371faa94b03
bde834e4a9be
  Building wheel for antiorm (setup.py) ... done
  Created wheel for antiorm: filename=antiorm-1.2.1-py3-none-any.whl size=31670 sha256=5d8979123f6ca29505e08731cac99fa60
af336f5bb39a91e74f4376e30fd77b6
  Stored in directory: c:\users\cristiano\appdata\local\pip\cache\wheels\c5\43\70\9e729370cfff40c49d3e3d05377d54b3ecd71f
64e62341ea80
Successfully built db-sqlite3 db antiorm
Installing collected packages: antiorm, db, db-sqlite3
Successfully installed antiorm-1.2.1 db-0.1.1 db-sqlite3-0.0.1

(base) C:\Windows\system32>
```

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;

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

```
In [2]: a = 2
```

```
In [3]: a
```

```
Out[3]: 2
```

```
In [4]: b = a*a
```

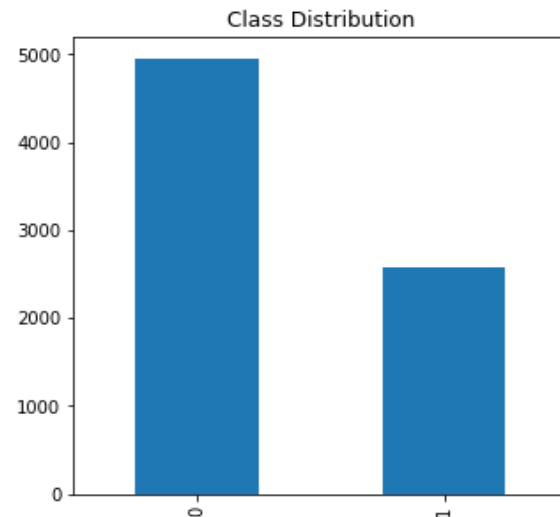
```
In [5]: b
```

```
Out[5]: 4
```

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**

- ***Panel *datas**** (“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 |

```
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 |

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

Visualmente mais intuitivo!

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 |

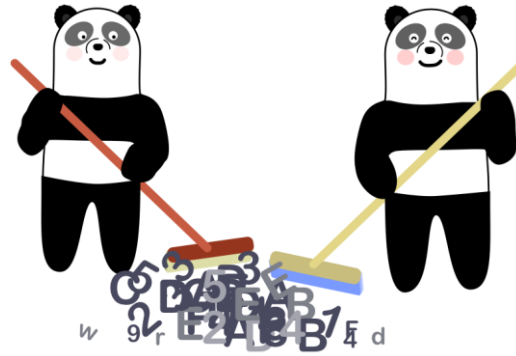
```
In [9]: import numpy as np
```

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

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

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

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



Manipulação e Limpeza de Dados