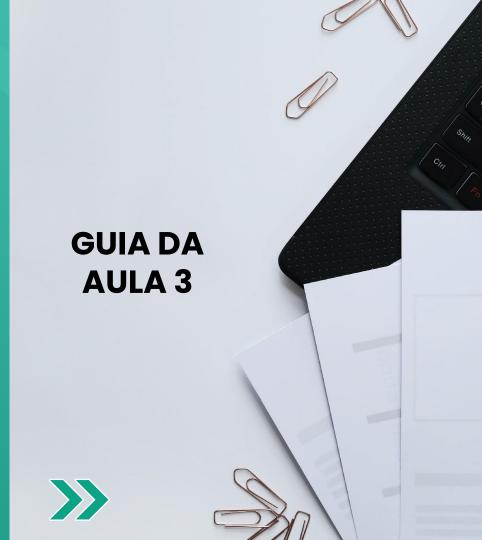


Profissão: Analista de dados





BIG DATA I – PROCESSAMENTO







Conheça o Data Wrangling

- Exploração
- Limpeza
- Junção
- Escrita



Acompanhe aqui os temas que serão tratados na videoaula







Exploração

Vamos utilizar a API Python do Spark, o pacote PySpark, para limpar os dados da aula 2.

Nota: Atente-se sempre à natureza distribuída das operações.

```
In [ ]: data.count()
In [ ]: data.columns
In [ ]: len(data.columns)
```





Limpeza

O método select seleciona colunas do DataFrame. Já o método withColumnRenamed renomeia colunas.









O método filter seleciona linhas do DataFrame baseado no conteúdo de uma coluna

```
In []: data_description = data.filter(data['year'] == 'Units')
In []: data_description.show(n=10)
```





Junção

```
In []: (data.count(), len(data.columns))
In []: (data_description.count(), len(data_description.columns))
```

O método join faz a junção de dois DataFrames. Já o método broadcast marca" um DataFrame como "pequeno" e força o Spark a trafegá-lo pela rede.

```
In []: from pyspark.sql.functions import broadcast
```





```
In []: data = data.join(
    other=broadcast(data_description)
    , on=['year'],
    how='left_anti'
)
In []: data.show(n=10)
```

O método dropna remove todas as linhas que apresentarem ao menos um valor nulo.





O método withColumn ajuda a criar novas colunas.





O método colleta é uma ação que coleta os resultados dos nós e retorna para o Python.





Escrita

O método write.csv persiste o DataFrame em formato csv. Já o método repartition controla o número de partições da escrita.

```
In []:
    data.\
        repartition('century').
        \ write.\
        csv(
            path="uk-macroeconomic-data-clean"
            , sep=",",
            header=True,
            mode="overwrite"
            "
            )
```

