Desafio Data Wrangling e Pipeline

Bootcamp Business Intelligence



Extração e Exploração

```
##1. Leitura e Exploração Inicial
from google.colab import drive
drive.mount('/content/drive')

##1. Leia o arquivo CSV e visualize as 5 primeiras linhas do dataset.
import pandas as pd
dataframe = pd.read_csv("/content/vendas.csv")
print(dataframe.head())
```

```
##2. Verifique o número de linhas e colunas.
contLinha = len(dataframe.axes[0])
contColuna = len(dataframe.axes[1])
print("Número de colunas:"+str(contColuna))
print("Número de linhas:"+str(contLinha))
```

Número de colunas:7



Exploração

```
##3. Descubra o tipo de dados (dtype) de cada coluna.
 tipo colunas = dataframe.dtypes
 print(tipo colunas)
                    int64
id venda
data venda
                   object
cliente
                   object
produto
                   object
quantidade
                 object
preco unitario
                 float64
categoria
                   object
dtype: object
```

```
##4. Conte os valores ausentes existentes em cada coluna.
valores ausentes = dataframe.isnull().sum()
# Exibir o resultado
print("Valores ausentes por coluna:\n")
print(valores ausentes)
Valores ausentes por coluna:
id venda
data venda
cliente
                  335
produto
quantidade
preco unitario
                   52
categoria
dtype: int64
```



Filtragem e ordenação

#1. Filtre apenas as linhas onde o preço unitário é maior que 100.

```
preco_maior_que_100 = dataframe['preco_unitario'] > 100
print(dataframe[preco_maior_que_100])
```

∓ ₹		id_venda	data_venda	cliente	produto	quantidade	١
	0	1	2025/06/01	Fernanda Lima	iPhone 13	5	
	2	3	04-04-2025	Maria Silva	Notebook Dell	2	
	4	5	15-09-2025	Fernanda Lima	Mouse Gamer	3	
	7	8	2025/01/22	Pedro Costa	Mochila	1	
	9	10	2025-02-03	Pedro Costa	Notebook Dell	2	
	1987	1988	10-09-2025	Pedro Costa	iPhone 13	5	
	1991	1992	2025-04-05	Pedro Costa	Impressora HP	4	
	1994	1995	2025/04/19	Maria Silva	Monitor LG	2	
	1995	1996	2025-07-20	Pedro Costa	iPhone 13	3	
	1999	2000	2025/12/27	Ana Pereira	iPhone 13	três	

```
#2. Ordene o dataset pelo valor do preço em ordem decrescente.
# Ordenar pelo preço unitário em ordem decrescente
df ordenado = dataframe.sort values(by='preco unitario', ascending=False)
# Exibir as 5 primeiras linhas para verificação
print(df ordenado.head())
     id venda data venda
                                cliente
                                           produto quantidade \
           26 2025-06-07
                                    NaN iPhone 13
            1 2025/06/01 Fernanda Lima iPhone 13
          1983 01-04-2025
                            Ana Pereira iPhone 13
           25 2025-07-23
                            Pedro Costa iPhone 13
                            Maria Silva iPhone 13
           20 26-09-2025
     preco unitario categoria
             5500.0 Fletrônicos
             5500.0 Eletrônicos
1982
             5500.0 Eletrônicos
             5500.0 Eletrônicos
             5500.0 Eletrônicos
```



Transformação

```
# Padronizar a coluna de data (YYYY-MM-DD).
dataframe['data venda'] = (
    dataframe['data venda']
    .astype(str)
    .str.strip()
    .str.replace(r'[/\.]', '-', regex=True)
datas = pd.to_datetime(dataframe['data_venda'], errors='coerce', format='%Y-\%m-\%d')
mask_nat = datas.isna()
datas[mask nat] = pd.to datetime(dataframe.loc[mask nat, 'data venda'], errors='coerce', dayfirst=True)
dataframe['data venda'] = datas.dt.strftime('%Y-%m-%d')
print(dataframe['data venda'].head())
     2025-06-01
     2025-05-02
2 2025-04-04
     2025-06-21
     2025-09-15
Name: data venda, dtype: object
```



Transformação

```
# Substituir valores nulos por "Não informado".
    newDataframe = dataframe.fillna('Não Informado')
    print(newDataframe.head())
∓*
                 data venda
                                    cliente
                                                   produto quantidade
       id venda
                 2025-06-01
                             Fernanda Lima
                                                 iPhone 13
    0
                                João Souza
                 2025-05-02
                                                   Caderno
                 2025-04-04
                                Maria Silva
                                            Notebook Dell
                             Fernanda Lima
                 2025-06-21
                                                   Caderno
                 2025-09-15
                             Fernanda Lima
                                               Mouse Gamer
       preco unitario
                          categoria
                       Eletrônicos
    0
               5500.0
                          Papelaria
                 25.0
```

```
# Corrigir quantidade para ser sempre um número inteiro.
     dataframe['quantidade'].unique()
     dataframe['quantidade'] = dataframe['quantidade'].replace('três', '3')
    dataframe['quantidade'] = pd.to numeric(dataframe['quantidade'])
     dataframe.dtypes
∓
                         0
                      int64
       id venda
       data_venda
                     object
         cliente
                     object
        produto
                     object
       quantidade
                      int64
     preco_unitario
                     float64
       categoria
                     object
```

dtype: object



3500.0

25.0

200.0

Eletrônicos

Eletrônicos

Papelaria

Transformação

```
# Remover ou corrigir preços negativos.
if 'preco_unitario' in dataframe.columns:
    dataframe = dataframe[dataframe['preco_unitario'] >= 0]
```

```
# Criar uma nova coluna valor_total = quantidade * preco_unitario.

if 'quantidade' in dataframe.columns and 'preco_unitario' in dataframe.columns:

dataframe['valor_total'] = dataframe['quantidade'] * dataframe['preco_unitario']
```





Criar um banco SQLite (arquivo vendas.db).

Desafio Extra

```
# Criar uma segunda tabela tb_clientes contendo apenas clientes únicos.
df_clientes = dataframe[["cliente"]].drop_duplicates().reset_index(drop=True)
df_clientes.to_sql("tb_clientes", conexao, if_exists="replace", index=False)
print("Tabela 'tb_clientes' criada com clientes únicos no banco de dados 'vendas.db'.\n\n")

# Relacionar tb_vendas com tb_clientes via chave estrangeira.
    dataframe = dataframe.merge(df_clientes, on="cliente", how="left")
    print("Relacionamento lógico entre tb_vendas e tb_clientes criado (coluna: cliente). \n\n")
```



Desafio Extra



```
# Adicional gerar um relatório em arquivo texto com um resumo do banco de dados.
print("\n\nGerando relatório do banco de dados...")
print("Relatório gerado com sucesso: relatorio_banco_de_dados.txt\n")
arquivo_saida="relatorio_banco_de_dados.txt"
tabelas = pd.read sql query(
    "SELECT name FROM sqlite master WHERE type='table';", conexao
with open(arquivo saida, "w", encoding="utf-8") as f:
    f.write(f" RELATÓRIO DO BANCO: vendas.db\n")
   f.write("="*60 + "\n\n")
   f.write("Tabelas encontradas:\n")
    f.write(str(tabelas) + "\n\n")
    # Para cada tabela, exibir resumo e 15 primeiros registros
    for tabela in tabelas['name']:
       f.write(f" Tabela: {tabela}\n")
       f.write("-"*60 + "\n")
       # Esquema (colunas e tipos)
        schema = pd.read sql query(f"PRAGMA table info({tabela});", conexao)
       f.write(" Estrutura:\n")
       f.write(schema.to string(index=False) + "\n\n")
```

```
# Primeiras linhas
df_preview = pd.read_sql_query(f"SELECT * FROM {tabela} LIMIT 30;", conexao)
f.write(" Primeiros registros:\n")
f.write(df_preview.to_string(index=False) + "\n\n")

# Contagem de registros
count = pd.read_sql_query(f"SELECT COUNT(*) AS total FROM {tabela};", conexao)
f.write(f" Total de registros: {count['total'][0]}\n")
f.write("="*60 + "\n\n")
conexao.close()
```



```
RELATORIO DO BANCO: vendas.db
Tabelas encontradas:
         nane
    tb vendas
1 tb clientes
Tabela: tb vendas
Estrutura:
                     type notnull dflt value pk
cid
              name
          id venda INTEGER
                                          None
        data venda
                                          None
           cliente
                      TEXT
                                          None
           produto
                      TEXT
                                          None
        quantidade INTEGER
                                          None
  5 preco unitario
                      REAL
                                          None
         categoria
                      TEXT
                                          None
       valor total
                      REAL
                                          None
Primeiros registros:
id venda data venda
                                                                                 categoria valor total
                          cliente
                                          produto quantidade preco unitario
       1 2025-06-01 Fernanda Lima
                                        iPhone 13
                                                                       5588.8
                                                                               Eletrônicos
                                                                                                27500.0
       2 2025-05-02
                       João Souza
                                          Caderno
                                                                         25.8
                                                                                  Papelaria
                                                                                                   25.0
       3 2025-04-04
                      Maria Silva
                                    Notebook Dell
                                                                      3588.8
                                                                               Eletrônicos
                                                                                                 7888.8
       4 2025-06-21 Fernanda Lima
                                          Caderno
                                                                         25.8
                                                                                  Papelaria
                                                                                                   25.8
       5 2025-09-15 Fernanda Lina
                                                                       200.0
                                                                               Eletrônicos
                                                                                                  600.0
                                      Mouse Gamer
       6 2025-01-22 Fernanda Lima
                                          Caderno
                                                                         25.8
                                                                                 Papelaria
                                                                                                  125.0
       7 2025-05-22 Fernanda Lima
                                      Caneta Azul
                                                                         2.5
                                                                                  Papelaria
                                                                                                    7.5
                                          March 27 -
                                                                        200 0
                                                                                                  *** *
```

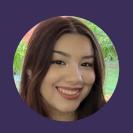


Pipeline.py

```
from etl.extraction import extract_data
   from etl.transform import transform_data
   from etl.load import load data
v def run pipeline():
       # Caminhos
       input path = 'dados/vendas raw.csv'
       output path = 'dados/'
       # Etapas ETL
       df = extract_data(input_path)
       df = transform data(df)
       load data(df, output path)
   if name == " main ":
       run pipeline()
```



Organização do Squad









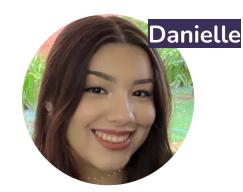








Gestora de Atividades:



- Sorteio sobre quem ficaria responsável por cada atividade;
- Uso do Trello para gestão;
- Acompanhamento através do Discord.





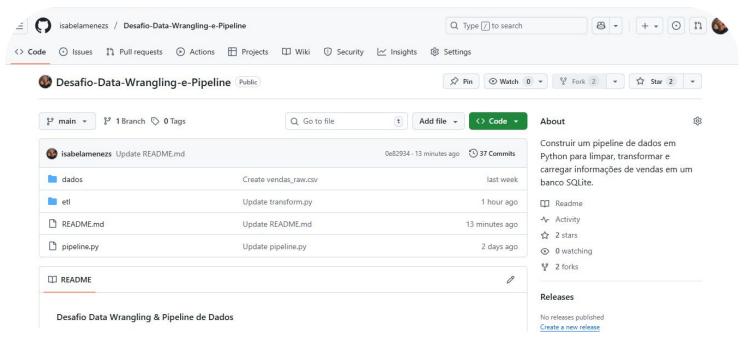


Kanban





Repositório do Github



https://github.com/isabelamenezs/Desafio-Data-Wrangling-e-Pipeline



Facilidades e Dificuldades



