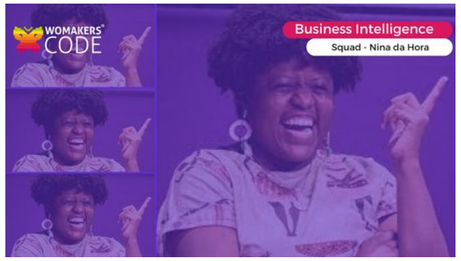
Desafio Data Wrangling e Pipeline

EQUIPE: NINA DA HORA



Equipe Nina da Hora

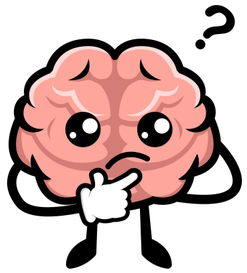
Chamada para ajudar uma startup.

CSV com inconsistências

- Datas diferentes
- Valores nulos
- Quantidades em texto
- Preços negativos

Colocar ordem no caos

- Limpar
- Transformar
- Carregar em SQLite



Primeiros passos: entendendo os dados

Explorando os dados

pd.read_csv("vendas.csv")

df_venda.head()

	id_venda	data_venda	cliente	produto	quantidade	preco_unitario	categoria
0	1	2025/06/01	Fernanda Lima	iPhone 13	5	5500.0	Eletrônicos
1	2	2025-05-02	João Souza	Caderno	1	25.0	Papelaria
2	3	04-04-2025	Maria Silva	Notebook Dell	2	3500.0	Eletrônicos
3	4	21-06-2025	Fernanda Lima	Caderno	1	25.0	Papelaria
4	5	15-09-2025	Fernanda Lima	Mouse Gamer	3	200.0	Eletrônicos

Dimensão dos dados

df_vendas.shape

	Descrição	Quantidade
0	Número de linhas	2000
1	Número de colunas	7

Tipos de dados

df_vendas.dtypes

	Coluna	Tipo de dado
0	id_venda	int64
1	data_venda	object
2	cliente	object
3	produto	object
4	quantidade	object
5	preco_unitario	float64
6	categoria	object

Valores ausentes

df_venda.isnull().sum()

	Coluna	Valores ausentes
0	id_venda	0
1	data_venda	0
2	cliente	335
3	produto	0
4	quantidade	0
5	preco_unitario	0
6	categoria	52

Seleção e Filtragem de Dados

df_csv.sort_values(by='preco_unitario', ascending=False)

preco_unitario
5500.0
5500.0
5500.0
5500.0
5500.0

df_csv[df_csv['preco_unitario'] > 100]

preco_unitario
5500.0
3500.0
200.0
150.0
3500.0

Objetivo do Desafio

Extração

CSV carregado e explorado

Extração de Dados

```
# ==============================
# EXTRAÇÃO DE DADOS
# ==============================

print(Fore.GREEN + ' Etapa 1: Extraindo dados...')
df_vendas = pd.read_csv("dataset/vendas.csv")
# Visualização inicial
print(df_vendas.head())
```

- Carregar o dataset vendas.csv.
- Explorar os dados: visualizar as primeiras linhas.

Extração

E

CSV carregado e explorado

T

Transformação

Nesta etapa, realizamos ajustes como padronizar datas, substituir valores nulos, corrigir quantidades e preços, e criamos uma nova coluna de valor total.

Bibliotecas utilizadas no projeto

panda

Leitura e manipulação de dados

datetime

Tratamento de datas

sqlite3

Criação e carga no banco de dados

colorama

Realce visual no terminal

Padronização

```python
#  Padronizar a coluna de data (YYYY-MM-DD)
print(Fore.GREEN + '\n====== Padronizar a coluna de data (YYYY-MM-DD) ======')
def normalizando_data(date_str):
    date_str = str(date_str).replace("/", "-")
    formatos = ["%Y-%m-%d", "%d-%m-%Y", "%m-%d-%Y"]
    for formato in formatos:
        try:
            return datetime.strptime(date_str, formato)
        except ValueError:
            pass
    return pd.NaT

df_vendas['data_venda'] = df_vendas['data_venda'].apply(normalizando_data)
df_vendas['data_venda'] = df_vendas['data_venda'].dt.strftime('%Y-%m-%d')
print(df_vendas.head())
```

- Padronização de datas para YYYY-MM-DD

Limpeza

```python
# Valores nulos e correção de quantidade
df_vendas = df_vendas.fillna("Não informado")
df_vendas['quantidade'] = df_vendas['quantidade'].replace('três', '3').astype(int)

# Preços negativos
df_vendas = df_vendas[df_vendas['preco_unitario'] >= 0]

# Criar coluna valor_total
df_vendas["valor_total"] = df_vendas["quantidade"] * df_vendas["preco_unitario"]
```

- Valores nulos substituídos por "Não informado".
- Quantidades convertidas para números inteiros.
- Remover preços negativos.
- Calcular valor_total.

Extração

E

CSV carregado e explorado

Transformação

T

Nesta etapa, realizamos ajustes como padronizar datas, substituir valores nulos, corrigir quantidades e preços, e criar uma nova coluna de valor total.

Carga

L

Após realizar todas as transformações, salvamos o dataset final limpo e padronizado.

Banco de Dados

```
conexao = sqlite3.connect("vendas.db")
cursor = conexao.cursor()

cursor.execute("PRAGMA foreign_keys = ON;")

# Criar tabela de clientes únicos
tb_clientes = df_vendas[['cliente']].drop_duplicates().reset_index(drop=True)
tb_clientes['cliente_id'] = tb_clientes.index + 1
df = df_vendas.merge(tb_clientes, on='cliente', how='left')
```

- Criar base de dados “vendas.db”.
- Ativar suporte a chaves estrangeiras.
- Criar a variável tb_clientes (DataFrame com clientes únicos).

Banco de Dados

```
cursor.execute("""
CREATE TABLE IF NOT EXISTS tb_clientes (
    cliente_id INTEGER PRIMARY KEY,
    cliente TEXT
);
""")
cursor.execute("""
CREATE TABLE IF NOT EXISTS tb_vendas (
    id_venda INTEGER PRIMARY KEY AUTOINCREMENT,
    cliente_id INTEGER,
    data_venda TEXT,
    produto TEXT,
    quantidade INTEGER,
    preco_unitario REAL,
    valor_total REAL,
    categoria TEXT,
    FOREIGN KEY (cliente_id) REFERENCES tb_clientes(cliente_id)
);
""")
tb_clientes.to_sql("tb_clientes", conexao, if_exists="replace", index=False)
df.to_sql("tb_vendas", conexao, if_exists="replace", index=False)
```

- Criar Tabla tb_vendas para salvar os dados tratados.
- Criar tabela tb_clientes no SQLite a partir do DataFrame com IDs únicos.

.

Consulta

```
query_total_por_categoria = """
SELECT categoria, SUM(valor_total) AS total_vendas
FROM tb_vendas
GROUP BY categoria
ORDER BY total_vendas DESC;
"""
df_total_por_categoria = pd.read_sql_query(query_total_por_categoria, conexao)
print(df_total_por_categoria)

# Fechar conexão
conexao.close()
print(Fore.GREEN + "\nConexão fechada com sucesso!")
```

- Consultar total de vendas por categoria.
- Fechar a conexão.

Resumo Visual

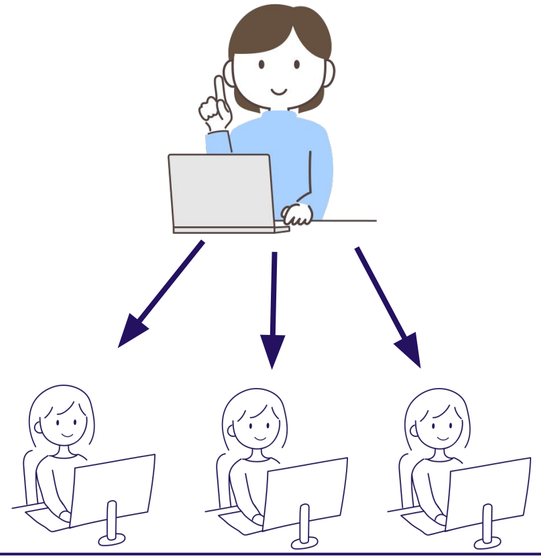
Ana Pereira
Maria Silva
Pedro Costa
Não informado
Maria Silva

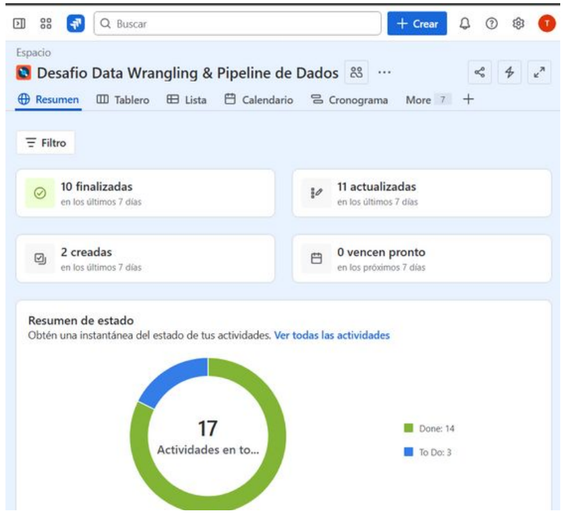
cliente_id	cliente
1	Fernanda Lima
2	João Souza
3	Maria Silva
4	Pedro Costa
5	Ana Pereira

id_venda	data_venda	cliente	produto	quantidade	preco_unitario	categoria	valor_total	cliente_id
1	2025-06-01	Fernanda Lima	iPhone 13	5	5500.0	Eletrônicos	27500.0	1
2	2025-05-02	João Souza	Caderno	1	25.0	Papelaria	25.0	2
3	2025-04-04	Maria Silva	Notebook Dell	2	3500.0	Eletrônicos	7000.0	3
4	2025-06-21	Fernanda Lima	Caderno	1	25.0	Papelaria	25.0	1
5	2025-09-15	Fernanda Lima	Mouse Gamer	3	200.0	Eletrônicos	600.0	1

Como nos organizamos como squad

Organização da equipe





Distribuição

A líder atribuiu tarefas de acordo com os pontos fortes de cada integrante.

Acompanhamento

Utilizamos o Jira para registrar as tarefas e acompanhar o progresso em tempo real.

Comunicação

Mantivemos uma comunicação constante pelo Discord para tirar dúvidas e nos coordenar de forma eficiente.

Facilidades e Dificuldades

Facilidades

- Leitura e exploração dos dados mais tranquila (pandas).
- Trabalho em grupo e troca de aprendizado.
- Estrutura do código em etapas e uso de funções conhecidas.
- Lógica de limpeza dos dados bem definida.

Dificuldades

- Padronização das datas em diferentes formatos.
- Etapa de carga no SQLite e relacionamento entre tabelas.
- Padronizar nomes de variáveis e unir códigos individuais.
- Conciliar diferentes estilos de escrita e manter organização.

Agradecemos a atenção de todas!

Acesse nosso repositório: escaneie o QR code