

03

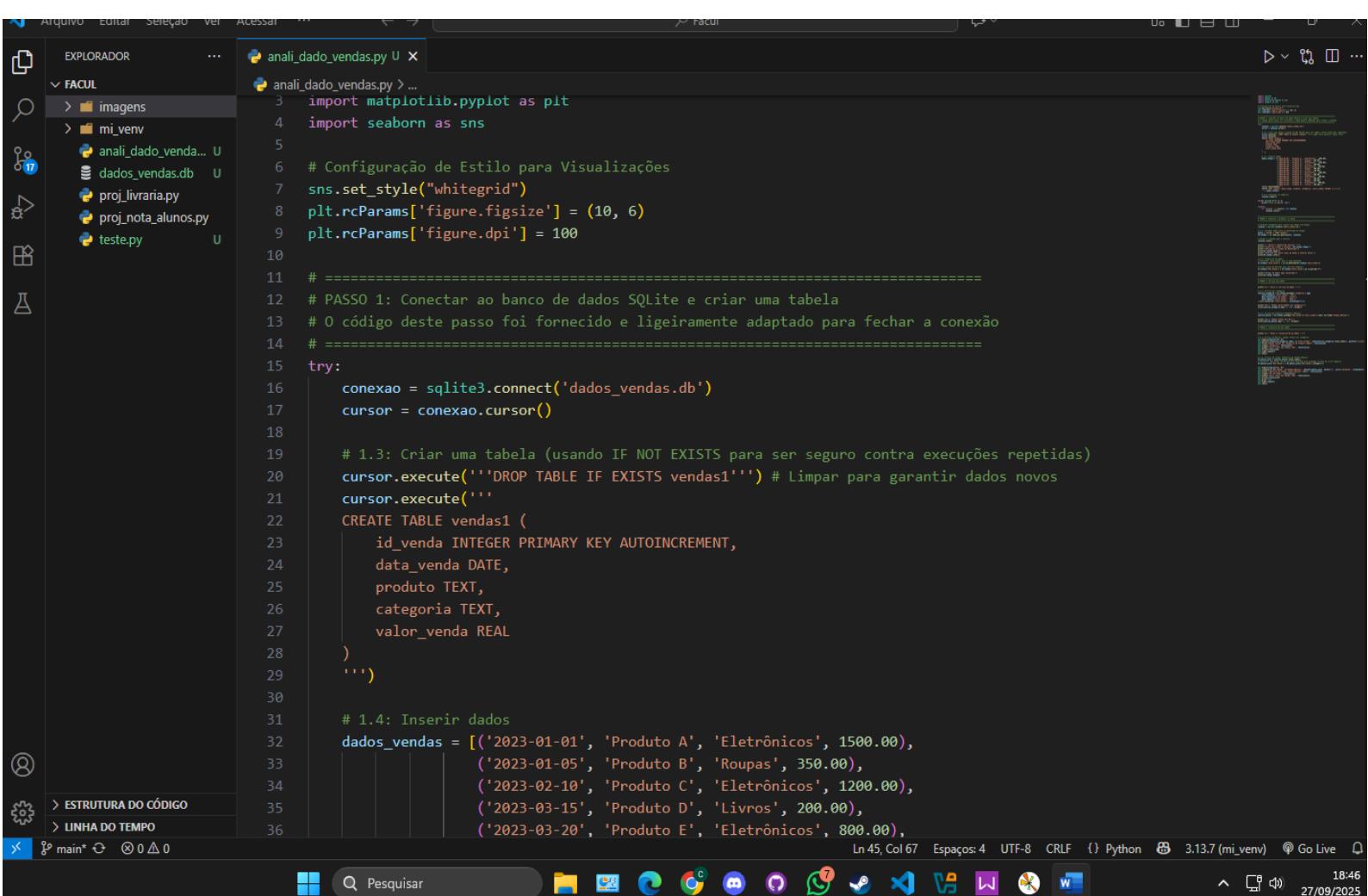
TRABALHO DE LINGUAGEM DE PROGRAMAÇÃO EM PYTHON

Objetivo Geral

O objetivo deste projeto é analisar os dados de vendas do último ano para identificar padrões, tendências e insights valiosos que possam ser utilizados para melhorar o desempenho e a estratégia de vendas da empresa.

Ferramentas e Bibliotecas Utilizadas

- **Python:** Linguagem de programação principal.
- **SQLite:** Banco de dados relacional para armazenar os dados de vendas.
- **Pandas:** Biblioteca de manipulação e análise de dados em Python. Essencial para transformar e processar o conjunto de dados.
- **Matplotlib:** Biblioteca de visualização de dados para criação de gráficos estáticos.
- **Seaborn:** Biblioteca de visualização baseada no Matplotlib, oferecendo uma interface de alto nível para criar gráficos estatísticos atraentes e informativo.



The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with icons for file operations like Open, Save, Find, and Run. Below the sidebar, there are sections for 'EXPLORADOR' (explorer), 'ESTRUTURA DO CÓDIGO' (code structure), and 'LINHA DO TEMPO' (time line). The main area displays a Python script named 'anali_dado_vendas.py'. The script starts with imports for matplotlib.pyplot and seaborn. It then sets a white grid style for plots and defines a figure size of (10, 6) and a dpi of 100. A try block follows, connecting to a SQLite database named 'dados_vendas.db' and creating a cursor. Inside the try block, it drops a table named 'vendas1' if it exists and creates a new one with columns id_venda (integer primary key autoincrement), data_venda (date), produto (text), categoria (text), and valor_venda (real). The script then inserts five rows of data into the table. The bottom status bar shows the line number (Ln 45), column number (Col 67), and encoding (UTF-8 CRLF). It also indicates the Python language and version (3.13.7 (mi_venv)). Other status indicators include 'Espaços: 4', 'Python', 'Go Live', and a date/time stamp '27/09/2023 18:46'.

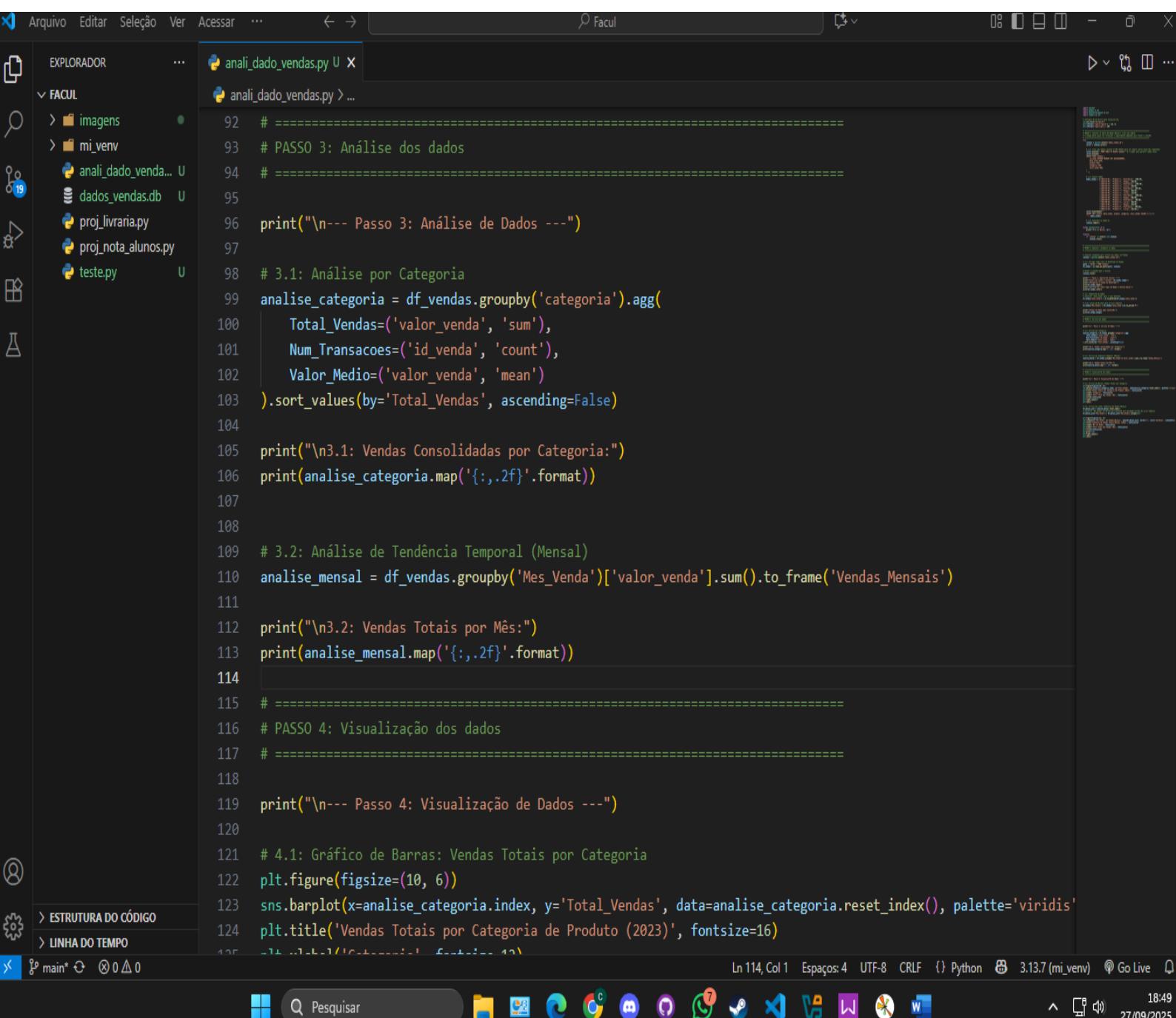
```
anali_dado_vendas.py
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3
4 # Configuração de Estilo para Visualizações
5 sns.set_style("whitegrid")
6 plt.rcParams['figure.figsize'] = (10, 6)
7 plt.rcParams['figure.dpi'] = 100
8
9 # =====
10 # PASSO 1: Conectar ao banco de dados SQLite e criar uma tabela
11 # O código deste passo foi fornecido e ligeiramente adaptado para fechar a conexão
12 # =====
13 try:
14     conexao = sqlite3.connect('dados_vendas.db')
15     cursor = conexao.cursor()
16
17     # 1.3: Criar uma tabela (usando IF NOT EXISTS para ser seguro contra execuções repetidas)
18     cursor.execute('''DROP TABLE IF EXISTS vendas1''') # Limpar para garantir dados novos
19     cursor.execute('''
20         CREATE TABLE vendas1 (
21             id_venda INTEGER PRIMARY KEY AUTOINCREMENT,
22             data_venda DATE,
23             produto TEXT,
24             categoria TEXT,
25             valor_venda REAL
26         )
27     ''')
28
29     # 1.4: Inserir dados
30     dados_vendas = [
31         ('2023-01-01', 'Produto A', 'Eletrônicos', 1500.00),
32         ('2023-01-05', 'Produto B', 'Roupas', 350.00),
33         ('2023-02-10', 'Produto C', 'Eletrônicos', 1200.00),
34         ('2023-03-15', 'Produto D', 'Livros', 200.00),
35         ('2023-03-20', 'Produto E', 'Eletrônicos', 800.00),
36     ]
37
38     cursor.executemany("INSERT INTO vendas1 (data_venda, produto, categoria, valor_venda) VALUES (?, ?, ?, ?)", dados_vendas)
39
40     conexao.commit()
41
42     print("Tabela criada com sucesso!")
43
44 except sqlite3.Error as e:
45     print(f'Ocorreu um erro ao conectar ao banco de dados: {e}')
46 finally:
47     cursor.close()
48     conexao.close()
```

Passo 1: Conexão e Preparação do Banco de Dados

The screenshot shows a Python development environment with the following details:

- Top Bar:** Arquivo, Editar, Seleção, Ver, Acessar, Facul (search bar), and system icons.
- Left Sidebar:** Icons for file operations (New, Open, Save, Find, Replace, etc.), a search bar, and sections like EXPLORADOR, FACUL, ESTRUTURA DO CÓDIGO, and LINHA DO TEMPO.
- Central Area:** A code editor window titled "anali_dado_vendas.py". The code is a script for analyzing sales data using Pandas and SQLite. It includes comments explaining the steps: "Explorar e preparar os dados", "Carregar dados para um DataFrame do Pandas", "Preparação de Dados", and "Análise dos dados".
- Right Area:** A large panel on the right displays various data visualizations, including a histogram of sales volume, a scatter plot of sales vs time, and other statistical summaries.
- Bottom Bar:** Status bar showing line 45, column 67, and other file information.

Conexão: É estabelecida uma conexão com o banco de dados **SQLite (dados_vendas.db)**. Se o arquivo não existir, ele é criado automaticamente.



The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar contains icons for file operations like Open, Save, and Close, along with sections for 'EXPLORADOR' (File Explorer), 'FACUL' (Faculdade), and 'ESTRUTURA DO CÓDIGO' (Code Structure). The main area displays a Python script named 'anali_dado_vendas.py'. The code performs data analysis on a SQLite database ('dados_vendas.db') and visualizes the results. The status bar at the bottom shows the file name 'main*', line count 'Ln 114, Col 1', character count 'Espaços: 4', encoding 'UTF-8', and file type 'Python'. The bottom right corner shows the date and time '27/09/2023 18:49'.

```
# =====#
# PASSO 3: Análise dos dados
# =====#
92     print("\n--- Passo 3: Análise de Dados ---")
93
94     # 3.1: Análise por Categoria
95     analise_categoria = df_vendas.groupby('categoria').agg(
96         Total_Vendas=('valor_venda', 'sum'),
97         Num_Transacoes=('id_venda', 'count'),
98         Valor_Medio=('valor_venda', 'mean'))
99     analise_categoria = analise_categoria.sort_values(by='Total_Vendas', ascending=False)
100
101    print("\n3.1: Vendas Consolidadas por Categoria:")
102    print(analise_categoria.map('{:.2f}'.format))
103
104
105    # 3.2: Análise de Tendência Temporal (Mensal)
106    analise_mensal = df_vendas.groupby('Mes_Venda')['valor_venda'].sum().to_frame('Vendas_Mensais')
107
108    print("\n3.2: Vendas Totais por Mês:")
109    print(analise_mensal.map('{:.2f}'.format))
110
111    # =====#
112    # PASSO 4: Visualização dos dados
113    # =====#
114
115    print("\n--- Passo 4: Visualização de Dados ---")
116
117    # 4.1: Gráfico de Barras: Vendas Totais por Categoria
118    plt.figure(figsize=(10, 6))
119    sns.barplot(x=analise_categoria.index, y='Total_Vendas', data=analise_categoria.reset_index(), palette='viridis')
120    plt.title('Vendas Totais por Categoria de Produto (2023)', fontsize=16)
121
122
```

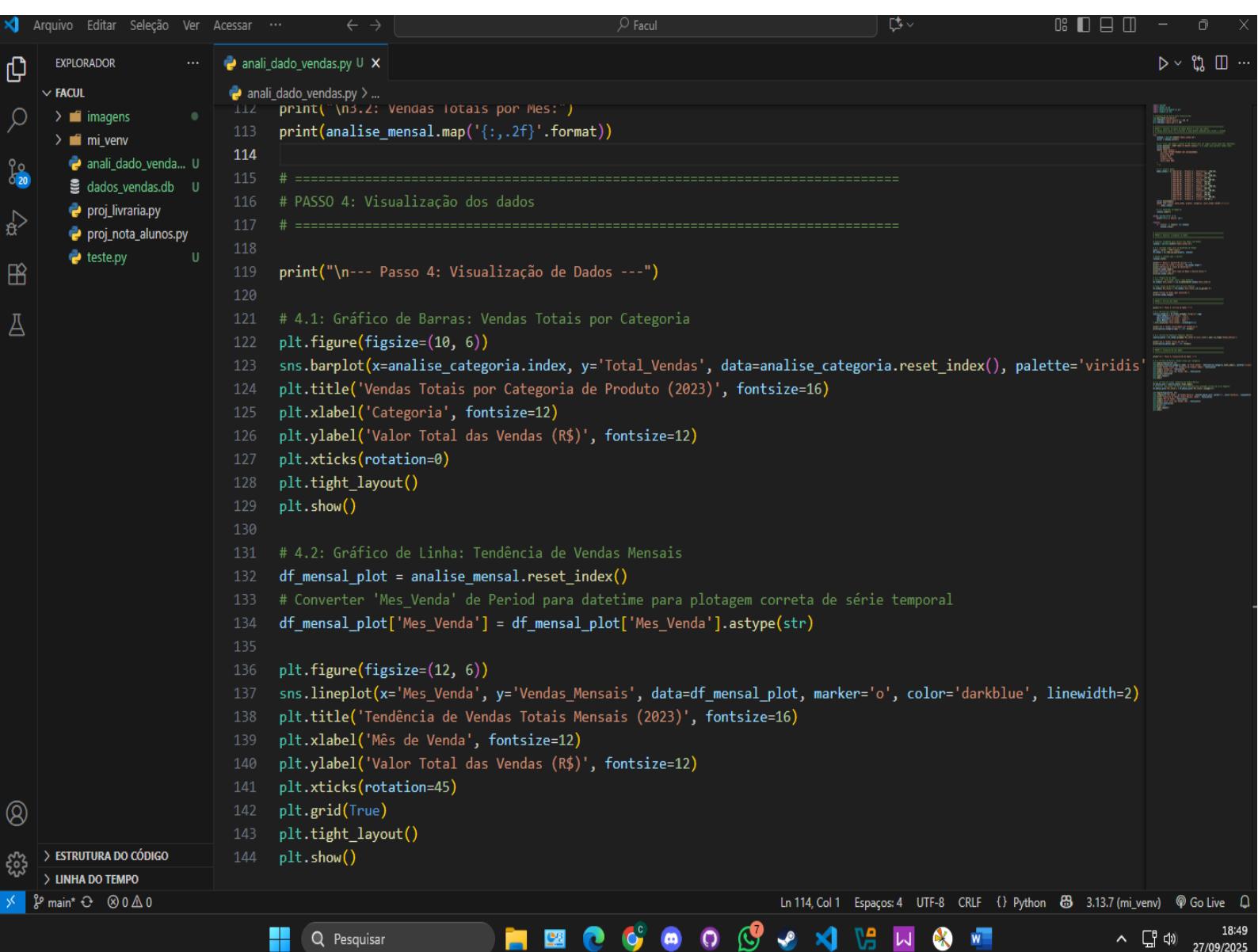
Visualização dos Dados

As visualizações criadas com **Matplotlib e Seaborn** ajudam a apresentar os insights de forma clara e intuitiva.

Vendas Totais por Categoria (Gráfico de Barras): Um gráfico de barras é utilizado para comparar visualmente o desempenho de vendas de cada categoria de produto.

Receita Mensal (Gráfico de Linhas): Um gráfico de linhas mostra a receita ao longo do tempo, destacando tendências e padrões sazonais.

Distribuição dos Valores de Venda (Histograma): Um histograma é usado para visualizar a frequência dos valores de venda, ajudando a entender a distribuição do ticket médio.



The screenshot shows a Python code editor interface with the following details:

- Left Sidebar:** Includes icons for File Explorer, Search, and various project-related tools.
- Top Bar:** Shows menu items like Arquivo, Editar, Seleção, Ver, Acessar, and others, along with a search bar labeled "Facul".
- Code Editor Area:** Displays a Python script named "anali_dado_vendas.py". The code uses Matplotlib and Seaborn libraries to create visualizations. It includes sections for printing total sales by category and monthly sales trends.
- Right Panel:** Shows a preview of the generated plots, including a bar chart for total sales by category and a line chart for monthly sales trends.
- Bottom Status Bar:** Provides information about the current file ("main.py"), line and column numbers (Ln 114, Col 1), encoding (UTF-8), and other settings.
- Taskbar:** At the bottom, it shows the Windows taskbar with various pinned icons.

Conclusões e Insights

Nesta etapa, os resultados da análise e das visualizações são interpretados para formular conclusões e recomendações estratégicas.

- **Principais Insights:** Apresentação dos achados mais relevantes, como as categorias mais lucrativas, os meses de maior e menor venda, e o ticket médio das transações.
- **Recomendações:** Sugestões práticas para a empresa, como a alocação de mais recursos para as categorias de alto desempenho, planejamento de promoções em meses de baixa sazonalidade ou a criação de campanhas de marketing direcionadas.

Com essa documentação, o projeto se torna mais organizado e fácil de ser compreendido e replicado. Você consegue visualizar a ordem lógica do processo, desde a coleta dos dados até a tomada de decisões estratégicas.

The screenshot shows a Python development environment with the following details:

- File Explorer:** Shows files in the 'FACUL' directory, including 'anali_dado_vendas.py', 'dados_vendas.db', 'proj_livraria.py', 'proj_nota_alunos.py', and 'teste.py'.
- Terminal:** Displays the command line interface with the script 'anali_dado_vendas.py' running, showing its execution path and some initial analysis steps.
- Output:** Shows the first 5 rows of a DataFrame and information about data types and null values.
- Figure:** A bar chart titled 'Vendas Totais por Categoria de Produto (2023)' comparing sales totals for three categories: Eletrônicos, Roupas, and Livros.
- Status Bar:** Shows file paths ('main.py'), line and column numbers (Ln 114, Col 1), and other system information.

Arquivo Editar Seleção Ver Acessar ... 🔍 Facul

EXPLORADOR ...

FACUL

- imagens
- mi_venv
- anali_dado_vendas.py** U
- dados_vendas.db U
- proj_livraria.py
- proj_nota_alunos.py
- teste.py U

anali_dado_vendas.py > ...

```

1 import sqlite3
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5
6 # Configuração de Estilo para Visualizações

```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

Figure 1

Tendência de Vendas Totais Mensais (2023)

Mês de Venda	Vendas_Mensais
2023-01	1,850.00
2023-02	1,200.00
2023-03	1,000.00
2023-04	400.00
2023-05	150.00
2023-06	1,000.00
2023-07	600.00
2023-08	700.00
2023-09	300.00
2023-10	450.00
2023-11	900.00
2023-12	250.00

--- Passo 3: Análise de Dados ---

3.1: Vendas Consolidadas por Categoria:

Categoria	Total_Vendas	Num_Transacoes	Valor_Medio
Eletrônicos	6,100.00	6.00	1,016.67
Roupas	1,800.00	4.00	450.00
Livros	900.00	4.00	225.00

3.2: Vendas Totais por Mês:

Mes_Venda	Vendas_Mensais
2023-01	1,850.00
2023-02	1,200.00
2023-03	1,000.00
2023-04	400.00
2023-05	150.00
2023-06	1,000.00
2023-07	600.00
2023-08	700.00
2023-09	300.00
2023-10	450.00
2023-11	900.00
2023-12	250.00

--- Passo 4: Visualização de Dados ---

d:\Facul\anali_dado_vendas.py:123: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

Ln 114, Col 1 Espaços: 4 UTF-8 CRLF Python 3.13.7 (mi_venv) Go Live

Pesquisar

18:53 27/09/2025