

CURSO SUPERIOR DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

WESLEN JHONY

LINGUAGEM DE PROGRAMAÇÃO

VISUALIZAÇÃO DE DADOS EM PYTHON

PRESIDENTE SARNEY

2025

Sumário

1 INTRODUÇÃO	3
2 DESENVOLVIMENTO	3
2.1 ESTRUTURA DO BANCO DE DADOS	3
2.2 INSERÇÃO DOS DADOS INICIAIS	4
2.3 MANIPULAÇÃO DOS DADOS COM PANDAS	4
2.4 GERAÇÃO DE GRÁFICOS	5
3 RELATÓRIO	7
3.1 EVOLUÇÃO TEMPORAL DAS VENDAS	7
3.2 DISTRIBUIÇÃO DE VENDAS POR CATEGORIA	7
3.3 RANKING DE PRODUTOS MAIS VENDIDOS	8
3.4 TICKET MÉDIO MENSAL	9
4 CONCLUSÃO	10
3 REFERÊNCIAS	11

1 INTRODUÇÃO

O presente relatório tem como objetivo demonstrar a criação de um sistema de banco de dados simples para armazenamento de informações de vendas, utilizando o **SQLite** como mecanismo de persistência e o **Python** como linguagem de programação. A partir desses dados, foram aplicadas técnicas de análise exploratória e geração de gráficos com a biblioteca **Matplotlib** e o **Pandas** para manipulação e sumarização das informações.

2 DESENVOLVIMENTO

Para a realização da atividade, foi utilizado o ambiente Visual Studio Code, que permite a execução de códigos em Python e diversas outras linguagens.

O sistema desenvolvido contempla quatro etapas principais, descritas a seguir.

2.1 ESTRUTURA DO BANCO DE DADOS

Foi criada uma tabela denominada vendas1, contendo os seguintes campos:

```
def criar_tabelas(self):  
  
    #executa o sql que cria a tabela caso ela não exista  
    self.cursor.execute('''  
        CREATE TABLE IF NOT EXISTS vendas1 (  
            id_venda INTEGER PRIMARY KEY AUTOINCREMENT,  
            data_venda DATE,  
            produto TEXT,  
            categoria TEXT,  
            valor_venda REAL  
        )  
    ''')  
  
    #efetua o commit que salva as alterações efetuadas no sql anterior  
    self.conn.commit()
```

Fonte: Elaboração própria com print de tela do aplicativo Visual Studio Code

2.2 INSERÇÃO DOS DADOS INICIAIS

Foram inseridos registros fictícios de vendas realizadas ao longo de 2023, contemplando diferentes produtos e categorias (Eletrônicos, Roupas e Livros).

```
# função que adiciona os dados principais para manipulação e geração de insights
def adicionar_dados_iniciais(self):

    #adiciona os dados com um INSERT
    self.cursor.execute('''
        INSERT OR IGNORE INTO vendas1 (data_venda, produto, categoria, valor_venda) VALUES
        ('2023-01-01', 'Produto A', 'Eletrônicos', 1500.00),
        ('2023-01-05', 'Produto B', 'Roupas', 350.00),
        ('2023-02-10', 'Produto C', 'Eletrônicos', 1200.00),
        ('2023-03-15', 'Produto D', 'Livros', 200.00),
        ('2023-03-20', 'Produto E', 'Eletrônicos', 800.00),
        ('2023-04-02', 'Produto F', 'Roupas', 400.00),
        ('2023-05-05', 'Produto G', 'Livros', 150.00),
        ('2023-06-10', 'Produto H', 'Eletrônicos', 1000.00),
        ('2023-07-20', 'Produto I', 'Roupas', 600.00),
        ('2023-08-25', 'Produto J', 'Eletrônicos', 700.00),
        ('2023-09-30', 'Produto K', 'Livros', 300.00),
        ('2023-10-05', 'Produto L', 'Roupas', 450.00),
        ('2023-11-15', 'Produto M', 'Eletrônicos', 900.00),
        ('2023-12-20', 'Produto N', 'Livros', 250.00);
    ''')

    #efetua o commit que salva as alterações efetuadas no sql anterior
    self.conn.commit()
```

Fonte: Elaboração própria com print de tela do aplicativo Visual Studio Code

2.3 MANIPULAÇÃO DOS DADOS COM PANDAS

Os dados foram extraídos do banco e carregados em um DataFrame Pandas, o que possibilitou a realização de operações de **agrupamento, soma e média**.

```
#classe que abstrai e implementa a logica de geração e análise de graficos para insights
You, antenitem | 1 author (You)
class Dados:

    def __init__(self,db:Database):
        # carrega todos os dados em um DataFrame
        self.df = pd.read_sql(sql="SELECT * FROM vendas1",con=db.conn)

    # mostra o cabeçalho dos dados contidos no DataFrame
```

2.4 GERAÇÃO DE GRÁFICOS

Com base nos dados processados, foram elaborados gráficos para analisar:

1. Evolução temporal
2. Distribuição por categoria
3. Ranking dos produtos mais vendidos
4. Ticket médio mensal

```
#função que gera um grafico em linha para analise de evolução temporal
def evolucao_temporal(self):
    plt.figure(figsize=(10,5))

    #define os dados que serão plotados no grafico de barras,
    # sendo eles a data_venda e valor_venda respectivamente
    plt.plot(self.df['data_venda'], self.df['valor_venda'], marker='o')

    # define a label do eixo X
    plt.xlabel("Data")
    # define a label do eixo Y
    plt.ylabel("Valor de Vendas")
    # define o titulo ou proposito do grafico
    plt.title("Evolução Temporal das Vendas")
```

```
#função que gera um grafico em barra para analise de distribuição de vendas por categoria
def distribuicao_por_categoria(self):

    # efetua o groupby da categoria e soma o valor de vendas por categoria
    soma = self.df.groupby("categoria")["valor_venda"].sum().reset_index()
    #define os dados que serão plotados no grafico de barras
    plt.bar(soma["categoria"], soma["valor_venda"])
    # define a label do eixo X
    plt.xlabel("Categoria")
    # define o titulo ou proposito do grafico
    plt.ylabel("Soma das Vendas")
    # define o titulo ou proposito do grafico
    plt.title("Vendas por Categoria")
```

```
#função que gera um grafico em barra para analise de produtos mais vendidos
def ranking_de_produtos(self):
```

```
    # efetua o groupby do produto e soma o valor de vendas por produto
    ranking = self.df.groupby("produto")["valor_venda"].sum().reset_index()

    #define os dados que serão plotados no grafico de barras
    plt.bar(ranking["produto"], ranking["valor_venda"])
    # define a label do eixo X
    plt.xlabel("Produto")
    # define o titulo ou proposito do grafico
    plt.ylabel("Soma das Vendas")
    # define o titulo ou proposito do grafico
    plt.title("Vendas por Produto")
```

```
#função que gera um grafico em barra para analise do ticket medio mensal dos produtos
def ticket_medio_mensal(self):
```

```
    # converte as datas contidas em data_venda para datetime, garantindo que todas estão no mesmo formato
    self.df["data_venda"] = pd.to_datetime(self.df["data_venda"])

    # efetua o groupby da data_venda com periodo mensal e efetuando a media do mês
    ticket_mes = (
        self.df.groupby(self.df["data_venda"].dt.to_period("M"))["valor_venda"].mean().reset_index()
    )

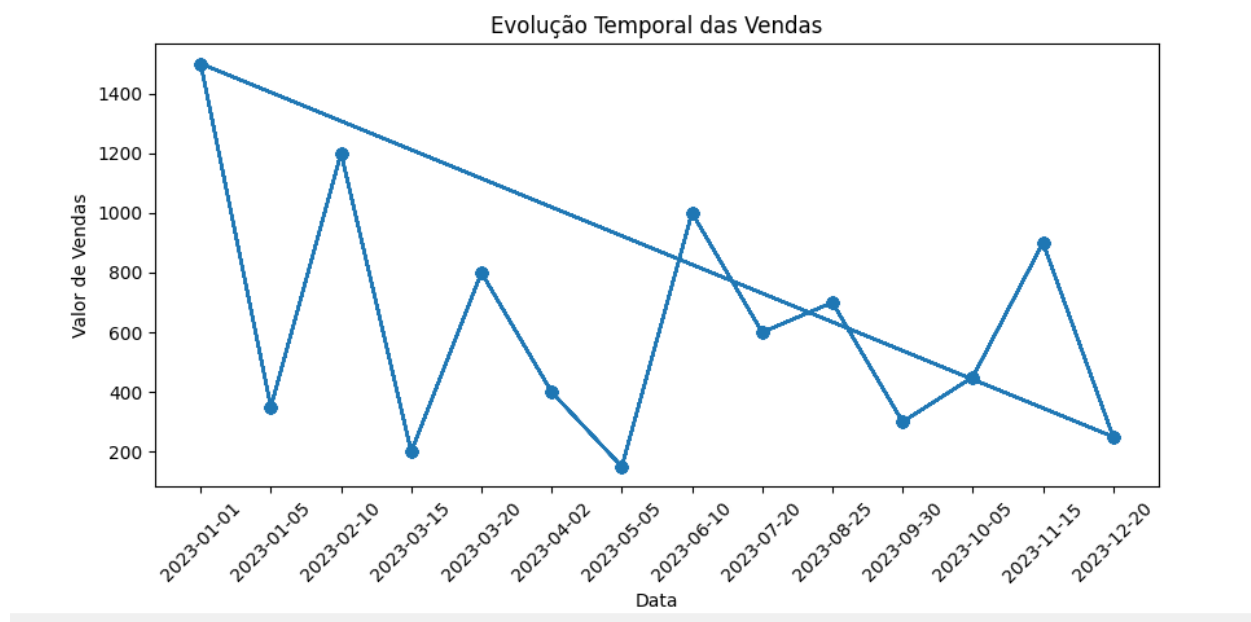
    #converte novamente a data_venda para string
    ticket_mes["data_venda"] = ticket_mes["data_venda"].astype(str)

    #define os dados que serão plotados no grafico de barras
    plt.bar(ticket_mes["data_venda"], ticket_mes["valor_venda"])
    # define a label do eixo X
    plt.xlabel("Mês")
    # define o titulo ou proposito do grafico
    plt.ylabel("Ticket médio")
    # define o titulo ou proposito do grafico
    plt.title("Ticket médio mensal")
```

3 RELATÓRIO

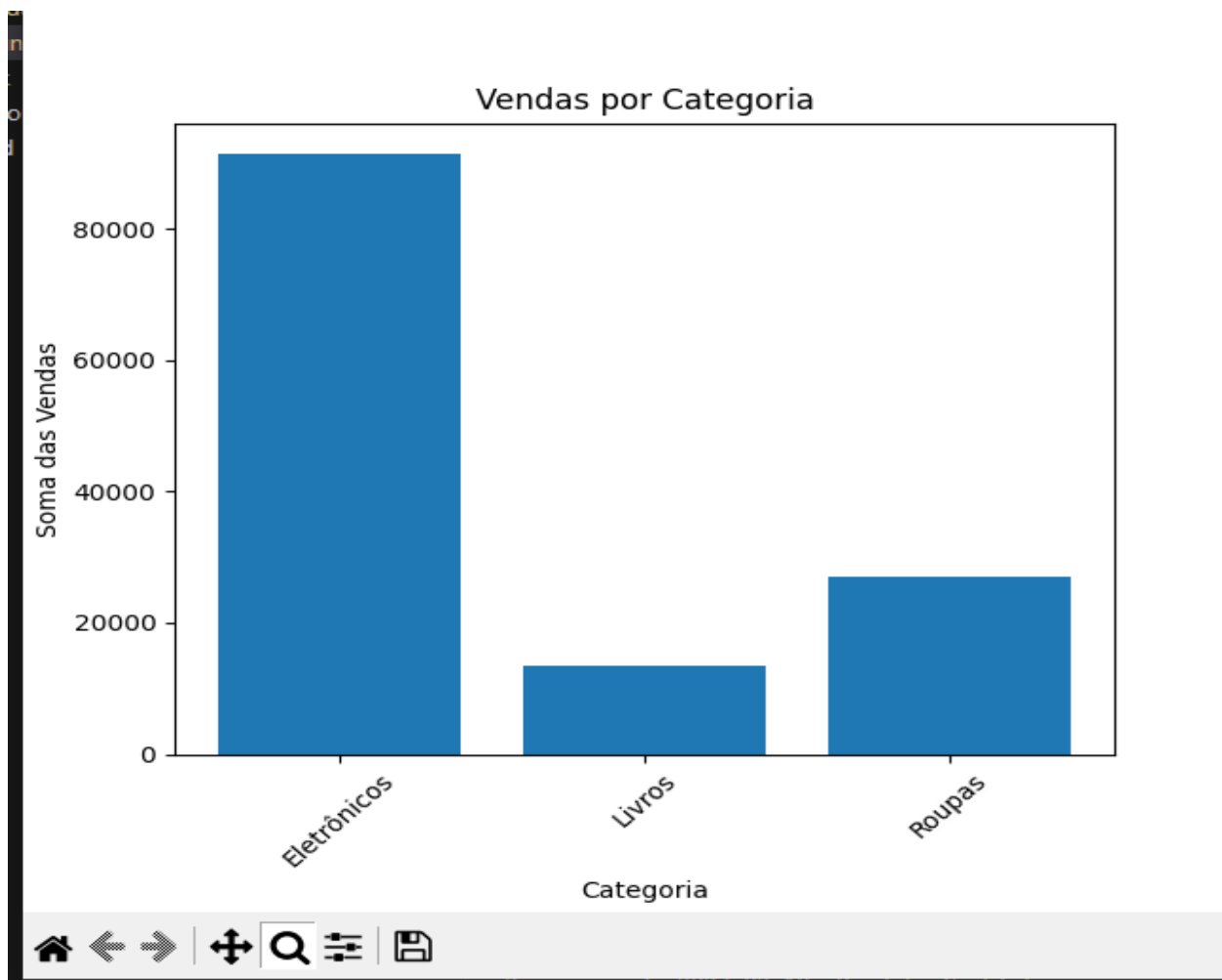
3.1 EVOLUÇÃO TEMPORAL DAS VENDAS

O gráfico de linha permitiu observar a variação das vendas ao longo do ano de 2023, destacando picos nos meses de janeiro, março e novembro.



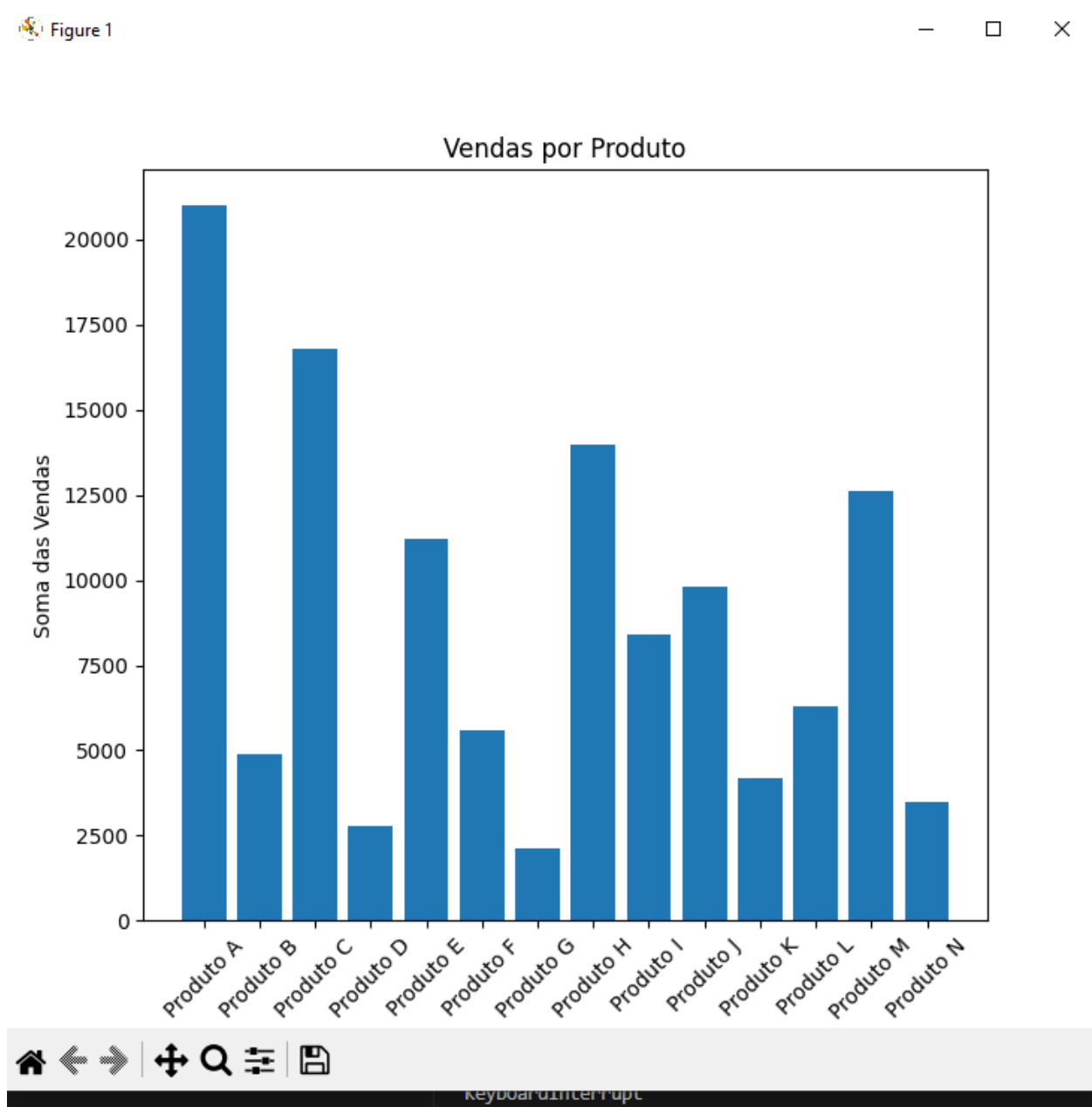
3.2 DISTRIBUIÇÃO DE VENDAS POR CATEGORIA

A categoria **Eletrônicos** concentrou o maior volume de vendas no período analisado, seguida por **Roupas** e **Livros**.



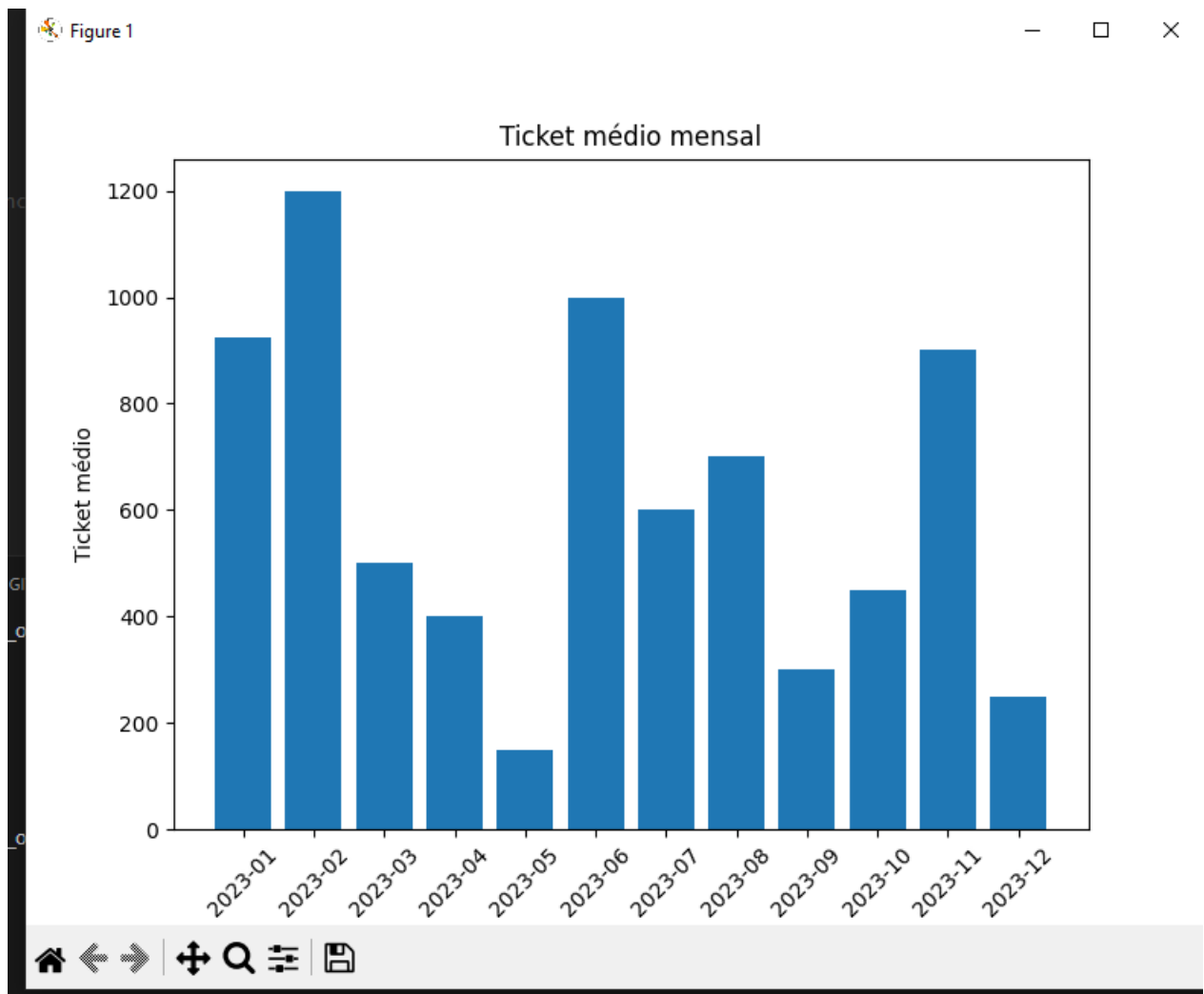
3.3 RANKING DE PRODUTOS MAIS VENDIDOS

Foi possível verificar que os produtos A, C, H e M (todos da categoria Eletrônicos) se destacaram em vendas, confirmando a relevância dessa categoria para a receita total.



3.4 TICKET MÉDIO MENSAL

Ticket médio mensal apresentou oscilações ao longo do ano, com valores mais elevados nos meses de janeiro e junho, o que pode indicar sazonalidade nas vendas.



4 CONCLUSÃO

O relatório demonstrou que é possível, por meio do uso de **Python, SQLite, Pandas e Matplotlib**, estruturar um processo de análise de dados completo e gerar insights relevantes sobre o desempenho de vendas.

A categoria **Eletrônicos** mostrou-se a mais representativa em termos de faturamento, e os meses de janeiro e novembro apresentaram maiores movimentações de vendas, informações que podem subsidiar decisões gerenciais em estratégias de estoque e marketing

3 REFERÊNCIAS

PYTHON SOFTWARE FOUNDATION. **Python 3 Documentation**. Disponível em: <https://docs.python.org/3/>. Acesso em: 10 set. 2025.

UNIVERSIDADE ANHANGUERA. **UNIDADE 3 – Introdução à Linguagem Python**. Material da disciplina. [S. l.: s. n.], 2025. Material em PDF.