

## Documentação do Código: Análise e Visualização de Vendas por Marca

Este documento tem como objetivo explicar as funcionalidades e estrutura do código fornecido, que realiza a análise e visualização de vendas por marca a partir de um conjunto de dados. O código está estruturado em uma classe chamada `GraficoVendas`, que encapsula as operações relacionadas à geração do gráfico de barras horizontais.

### Importação de Bibliotecas

O código inicia com as importações necessárias das bibliotecas pandas, matplotlib e seaborn para manipulação e visualização de dados.

```
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```
```

### Classe `GraficoVendas`

#### Método `\_\_init\_\_(self, caminho\_arquivo: str)`

O construtor da classe `GraficoVendas` recebe o caminho do arquivo CSV como parâmetro e carrega os dados para o atributo `dados`.

#### Método `calcular\_vendas\_por\_marca(self) -> pd.DataFrame`

Este método calcula o número total de vendas por marca. Retorna um DataFrame contendo as marcas e seus respectivos volumes de vendas, ordenados por volume.

#### Método `plotar\_grafico(self) -> None`

O método `plotar\_grafico` realiza as seguintes operações:

1. Calcula o número de vendas por marca usando o método `calcular\_vendas\_por\_marca`.
2. Define uma paleta de cores Viridis para o gráfico.
3. Inicializa um gráfico de barras horizontais usando a biblioteca seaborn.
4. Adiciona os valores das vendas em frente a cada barra.
5. Adiciona rótulos e título ao gráfico.
6. Ajusta o layout e salva o gráfico como uma imagem PNG chamada 'grafico\_vendas.png'.
7. Exibe o gráfico.

## **Função `main()`**

A função `main()` serve como ponto de entrada para o código. Nela, é definido o caminho do arquivo CSV e criada uma instância da classe `GraficoVendas`. Em seguida, é chamado o método `plotar\_grafico` para gerar e exibir o gráfico de vendas por marca.

## **Execução do Código**

Para executar o código, basta chamar a função `main()` dentro do bloco condicional `if \_\_name\_\_ == "\_\_main\_\_":`.

```
```python
if __name__ == "__main__":
    main()
```
```

Este bloco garante que o código dentro da função `main()` seja executado apenas se o script for executado diretamente e não importado como um módulo.