



Claro! Vou explicar o código de forma detalhada, abordando cada parte e sua função dentro do programa.

Este código implementa um **sistema de gerenciamento de loja** com controle de estoque, vendas, promoções e mais. O estoque é controlado em **gramas**, e o sistema permite cadastrar produtos, realizar vendas, aplicar promoções e calcular o total de vendas.


Estrutura do Código

1. Dicionário `produtos`:

- O dicionário `produtos` armazena os produtos da loja. Cada produto é uma chave (ex: "Arroz", "Feijão"), e o valor associado a cada produto é outro dicionário que contém:
 - `preço`: o preço do produto.
 - `estoque`: a quantidade em estoque do produto, medida em **gramas**.

Exemplo:

python


 Copiar código

```
produtos = {  
    "Arroz": {"preço": 20.00, "estoque": 50000}, # 50kg = 50.000g  
    "Feijão": {"preço": 10.00, "estoque": 30000}, # 30kg = 30.000g  
    "Macarrão": {"preço": 5.00, "estoque": 40000}, # 40kg = 40.000g  
}
```

2. Variável `total_vendas`:

- Essa variável mantém o **total acumulado de vendas realizadas** em reais.

python

 Copiar código

```
total_vendas = 0.0
```

Funções

1. Função `cadastrar_produto()`

- Esta função permite ao usuário **cadastrar um novo produto** no sistema, inserindo nome, preço e quantidade de estoque.
- A quantidade de estoque é informada em **quilos**, mas é convertida para **gramas** (multiplicando por 1000) antes de ser armazenada.

Fluxo:

- O nome do produto é inserido.
- O preço do produto é inserido.
- O estoque do produto em quilos é informado e convertido para gramas.
- O novo produto é adicionado ao dicionário `produtos`.

Código:

python

 Copiar código

```
def cadastrar_produto():
    nome = input("Digite o nome do produto: ")
    preco = float(input(f"Digite o preço de {nome}: R$"))
    estoque_kg = float(input(f"Digite o estoque de {nome} em quilos: "))
    estoque_g = estoque_kg * 1000 # Convertendo para gramas
    produtos[nome] = {"preço": preco, "estoque": estoque_g}
    print(f"Produto {nome} cadastrado com sucesso!\n")
```

2. Função `exibir_produtos()`

- Exibe todos os produtos disponíveis no estoque, mostrando o **nome**, o **preço** e a **quantidade de estoque em quilos**.
- O estoque é mostrado em **quilos**, que é calculado a partir dos gramas armazenados (dividindo por 1000).

Código:

python

 Copiar código

```
def exibir_produtos():
    print("\nProdutos disponíveis:")
    for produto, info in produtos.items():
        estoque_kg = info["estoque"] / 1000 # Convertendo de gramas para quilos
        print(f"{produto} - Preço: R${info['preço']:.2f}, Estoque: {estoque_kg:.2f} kg")
    print()
```

3. Função `realizar_venda()`


- Permite ao usuário **realizar uma venda** de um produto.
- O usuário escolhe o produto e informa a **quantidade em quilos** que deseja comprar.
- O estoque é verificado para garantir que há quantidade suficiente.
- O estoque do produto é **diminuído** conforme a venda, e o **valor total da venda** é somado ao `total_vendas`.

Fluxo:

- O nome do produto a ser comprado é inserido.
- A quantidade desejada é informada em **quilos**, convertendo para **gramas**.
- O programa verifica se o estoque é suficiente para atender à venda.
- Se o estoque for suficiente, o valor total da venda é calculado e o estoque do produto é reduzido.

Código:

python

 Copiar código

```
def realizar_venda():
    global total_vendas
    produto_vendido = input("Digite o nome do produto que deseja comprar: ")

    if produto_vendido in produtos:
        quantidade_kg = float(input(f"Digite a quantidade de {produto_vendido} que deseja "))
        quantidade_g = quantidade_kg * 1000 # Convertendo para gramas


        if produtos[produto_vendido]["estoque"] >= quantidade_g:
            valor_venda = quantidade_kg * produtos[produto_vendido]["preço"]
            produtos[produto_vendido]["estoque"] -= quantidade_g
            total_vendas += valor_venda
            print(f"Venda realizada: {quantidade_kg:.2f} kg de {produto_vendido} - Total: R$ {valor_venda:.2f}")
        else:
            print("Quantidade em estoque insuficiente.\n")
    else:
        print("Produto não encontrado.\n")
```

4. Função `exibir_vendas()`

- Exibe o **total de vendas realizadas** até o momento, somando todos os valores das vendas feitas.

Código:

python

 Copiar código

```
def exibir_vendas():  
    print(f"\nTotal de vendas realizadas: R${total_vendas:.2f}\n")
```

5. Função `sortear_promocao()`


- Aplica uma **promoção aleatória** a um produto sorteado, com um **desconto entre 10% e 50%**.
- O preço do produto sorteado é **modificado** com o desconto sorteado.

Fluxo:

- Um produto é sorteado aleatoriamente do dicionário `produtos`.
- Um valor de desconto é sorteado (entre 10% e 50%).
- O preço do produto sorteado é ajustado com o desconto.

Código:

python

 Copiar código

```
def sortear_promocao():  
    produto_sorteado = random.choice(list(produtos.keys()))  
    desconto = random.randint(10, 50) # Desconto entre 10% e 50%  
    produtos[produto_sorteado]["preço"] *= (1 - desconto / 100)  
    print(f"\nPromoção! O produto {produto_sorteado} está com {desconto}% de desconto!\n")
```

Função `menu()` :


- Exibe o **menu principal** do sistema, onde o usuário pode escolher a operação que deseja realizar.
- O menu tem opções para cadastrar produto, exibir produtos, realizar vendas, exibir o total de vendas, aplicar uma promoção e sair do sistema.

Fluxo:

- O menu é exibido repetidamente até o usuário escolher a opção de sair.
- Dependendo da escolha, o programa chama a função correspondente.

Código:

python

 Copiar código

```
def menu():
    while True:
        print("=== Sistema de Gerenciamento de Loja ===")
        print("1. Cadastrar produto")
        print("2. Exibir produtos")
        print("3. Realizar venda")
        print("4. Exibir total de vendas")
        print("5. Sortear promoção")
        print("6. Sair")

        opcao = input("Escolha uma opção: ")

        if opcao == "1":
            cadastrar_produto()
        elif opcao == "2":
            exibir_produtos()
        elif opcao == "3":
            realizar_venda()
        elif opcao == "4":
            exibir_vendas()
        elif opcao == "5":
            sortear_promocao()
        elif opcao == "6":
            print("Saindo do sistema...")
            break
        else:
            print("Opção inválida! Tente novamente.\n")
```

