MERCADINHO PYTHON

João Geraldo da Silva Neto, FC20230222

### É um sistema básico de supermercado construído em Python ele gerenciar o estoque de produtos, adicionar itens ao carrinho de compras, calcular a compra total, finalizar o processo de compras e entre outros.

### **Classes:**

1. **Produto**:
   1. Representa um produto com atributos como nome, preco e quantidade.
   2. Métodos:
      1. atualizar\_quantidade: Atualiza a quantidade do produto após a venda.
      2. \_\_str\_\_: Método especial para imprimir a representação do produto.
2. **Cliente**:
   1. Representa um cliente com nome, documento (CPF) e um carrinho de compras.
   2. Métodos:
      1. adicionar\_produto\_ao\_carrinho: Adiciona um produto ao carrinho do cliente.
      2. visualizar\_carrinho: Retorna o carrinho do cliente.
      3. \_\_str\_\_: Método especial para imprimir a representação do cliente.
3. **Carrinho**:
   1. Representa o carrinho de compras, que pode conter vários produtos.
   2. Métodos:
      1. adicionar\_produto: Adiciona um produto ao carrinho, se houver estoque suficiente.
      2. calcular\_total: Calcula o total da compra no carrinho.
      3. \_\_str\_\_: Método especial para imprimir os itens do carrinho e o total da compra.
4. **Supermercado**:
   1. Representa o supermercado que mantém o controle de seus produtos e clientes.
   2. Métodos:
      1. adicionar\_produto: Adiciona um produto ao estoque.
      2. buscar\_produto: Pesquisa um produto pelo nome.
      3. registrar\_cliente: Registra um cliente no sistema.
      4. realizar\_venda: Finaliza a compra de um cliente, exibindo o total da compra.
      5. visualizar\_produtos: Exibe todos os produtos cadastrados no supermercado.
5. **Terminal**:
   1. Classe com métodos estáticos para exibir menus e coletar entradas do usuário. Serve como interface para interação com o sistema.

### **Fluxo do Programa (Função main):**

1. **Registrar Cliente**: O sistema permite registrar novos clientes, pedindo nome e CPF.
2. **Adicionar Produto**: O sistema permite adicionar novos produtos ao estoque, incluindo nome, preço e quantidade.
3. **Visualizar Produtos**: Exibe todos os produtos disponíveis no estoque.
4. **Adicionar Produto ao Carrinho**: Permite que o cliente adicione um produto ao seu carrinho, se houver estoque disponível.
5. **Finalizar Compra**: O cliente pode finalizar a compra, o sistema calcula o total e limpa o carrinho após o pagamento.

### **Pontos de Melhoria ou Expansão:**

1. **Validação de Entrada**:
   1. As entradas de dados, como preço, quantidade e CPF, não são validadas. Seria interessante garantir que o usuário forneça dados válidos (ex.: o CPF precisa ter 11 dígitos, o preço não pode ser negativo, etc.).
2. **Múltiplos Clientes**:
   1. O sistema já suporta múltiplos clientes, mas não há verificação para garantir que um cliente com o mesmo nome não seja registrado mais de uma vez.
3. **Recuperação de Carrinho após Fechamento**:
   1. Após a finalização da compra, o carrinho é limpo (cliente.carrinho = Carrinho()), mas não há registro de compras passadas do cliente. Um sistema que registra o histórico de compras poderia ser interessante.
4. **Interface de Usuário (UI)**:
   1. O sistema está inteiramente baseado em linha de comando. Uma melhoria seria criar uma interface gráfica ou uma aplicação web para uma melhor interação com o usuário.
5. **Estoque de Produtos**:
   1. O estoque é atualizado quando um produto é adicionado ao carrinho. No entanto, seria interessante verificar o estoque antes de realizar a venda, para garantir que o produto ainda está disponível no momento do pagamento.
6. **Mensagens de Erro**:
   1. Quando o cliente não é encontrado ou um produto está fora de estoque, o programa apenas imprime uma mensagem simples. Pode ser interessante melhorar essas mensagens para torná-las mais informativas.

### **Exemplo de Execução:**

Suponha que você execute o sistema e realize as seguintes operações:

1. **Registrar Cliente**:
   1. Cliente João Silva é registrado.
2. **Adicionar Produto**:
   1. Produto "Arroz" com preço R$ 10.00 e 50 unidades é adicionado.
3. **Adicionar Produto ao Carrinho**:
   1. João adiciona 3 unidades de "Arroz" ao seu carrinho.
4. **Finalizar Compra**:
   1. João finaliza a compra, e o sistema calcula o total (R$ 30.00).