

Aluna: Nataly Vitória Silva De Sena RA:12722217647
Aluno: Ariel Oliveira de Souza e Sousa RA: 12722214421

Relatorio Sistemas Distribuídos

Requisitos de Software:

Linguagem: **Python**

Bibliotecas: O código utiliza a biblioteca padrão do Python, além do módulo **sqlite3** para interagir com o banco de dados **SQLite**.

Justificativa da Escolha da Tecnologia:

Python: Linguagem de programação versátil, com uma sintaxe clara e ampla comunidade de suporte. A escolha do SQLite provavelmente se deve à sua simplicidade e portabilidade, adequado para uma aplicação local de gerenciamento de vendas.

Instruções para Instalação e Execução da Aplicação:

Utilizando o IDE VSCODE, abra o arquivo via CMD ou indo dentro do proprio VSCODE, na opção "File", depois "Open Folder" e selecionar a pasta. Se estiver com o VSCODE traduzido para o português, é só ir na opção "Arquivo", depois "Abrir Pasta" e selecionar a pasta.

Para executar o código é bem simples. No lado superior direito, vai ter um símbolo de play "▶", é só clicar que aparece o terminal para executar o código

Apresentação e Detalhamento da Arquitetura, Estratégia e Algoritmos:

Arquitetura:

O código adota uma arquitetura modular baseada em classes e funções, seguindo o paradigma de programação orientada a objetos. As principais classes são `Cliente` e as funções relacionadas ao gerenciamento de vendas.

1. Classe Cliente:

- A classe `Cliente` representa um cliente e possui um método para exibir suas informações.

2. Funções de Banco de Dados:

- Funções como `criar_banco_dados` e `criar_tabela_clientes` são responsáveis pela criação e inicialização do banco de dados SQLite e suas tabelas.

3. Funções de Operações de Vendas:

- Funções como `adicionar_produto`, `receber_pedido` e relatórios como `relatorio_produtos_mais_vendidos` realizam operações no banco de dados relacionadas a produtos e pedidos.

4. Menus e Funções de Cliente:

- Os menus, como `menu_gerenciar_vendas` e `menu_clientes`, fornecem uma interface interativa para o usuário, permitindo operações de gerenciamento de vendas e clientes.

Estratégia:

O código utiliza um banco de dados SQLite para armazenar informações persistentes sobre produtos, pedidos e clientes. A estratégia é modular, com funções distintas para diferentes operações, facilitando a manutenção e expansão do sistema.

1. Operações Atômicas:

- Cada função realiza uma operação atômica, como adicionar um produto, receber um pedido ou gerar relatórios.

2. Menus Interativos:

- Menus interativos são implementados para permitir que o usuário escolha entre diferentes operações, facilitando a interação com o sistema.

Algoritmos:

1. Consultas SQL:

- O código utiliza consultas SQL para interagir com o banco de dados SQLite. Por exemplo, a função `relatorio_produtos_mais_vendidos` utiliza uma consulta SQL para obter os produtos mais vendidos.

2. Atualização de Estoque:

- A função `receber_pedido` realiza a atualização do estoque após receber um pedido, utilizando uma transação que consiste em inserir um novo pedido na tabela `pedidos` e atualizar o estoque na tabela `produtos`.

3. Relatórios Estatísticos:

- As funções que geram relatórios, como `relatorio_consumo_medio_cliente` e `relatorio_produtos_mais_vendidos`, utilizam consultas SQL agregadas para calcular estatísticas como a média de consumo por cliente e os produtos mais vendidos.

O código apresenta uma arquitetura organizada, estratégia modular e utiliza consultas SQL eficientes para interagir com o banco de dados. As operações são realizadas de forma atômica, proporcionando um sistema robusto para o gerenciamento de vendas. O uso de menus interativos facilita a interação do usuário com o sistema.