

Questionário de Análise e Reflexão de Código Projeto: Pet Shop "Amigo Fiel"

Abaixo estão as respostas às perguntas de reflexão obrigatórias sobre o desenvolvimento deste projeto.

1. Análise da Lógica de Venda


- a) Qual é o nome da função/método que você criou para processar a venda? A função principal que processa a venda no Flask se chama `registrar_venda`.
- b) Descreva a mensagem de erro exata (palavra por palavra) que o seu sistema exibe ao usuário se o estoque for insuficiente. A mensagem exata é: "Estoque insuficiente. Restam apenas [N] unidades do produto [Nome do Produto]." (Onde [N] e [Nome do Produto] são preenchidos dinamicamente).
- c) Como você garantiu que o estoque só é atualizado depois que a venda é 100% validada? O processo segue uma ordem estrita dentro da função `registrar_venda`:
 1. Primeiro, é feita uma leitura (`SELECT`) no banco para buscar o produto e verificar seu estoque atual.
 2. O código então executa uma verificação (`if`) para comparar o estoque atual com a quantidade solicitada.
 3. Se a quantidade solicitada for maior que o estoque, a função exibe a mensagem de erro e é interrompida (`return redirect(...)`) antes que qualquer escrita no banco ocorra.
 4. Somente se a verificação for aprovada, o sistema prossegue para as operações de escrita: `INSERT` na tabela `Vendas` e `UPDATE` na tabela `Produtos`. Ambas as operações são concluídas com um único `db.commit()`, garantindo que ou tudo funciona, ou nada é alterado.

2. Decisão de Implementação (Ordenação)

- a) Qual foi o algoritmo de ordenação que você pessoalmente implementou? Implementei o algoritmo `Insertion Sort` (Ordenação por Inserção) na função `insertion_sort(lista_de_produtos)`.
- b) Por que você escolheu esse algoritmo específico? A escolha foi baseada em um equilíbrio entre simplicidade de implementação e eficiência para o caso de uso. Dado o prazo de desenvolvimento, o Insertion Sort é um algoritmo que domino e é rápido de implementar corretamente. Além disso, para uma lista de produtos de um pet shop (que

raramente passará de algumas centenas de itens), o Insertion Sort é perfeitamente eficiente e não causa qualquer gargalo de performance perceptível.

3. Ponto Crítico da Simulação

- a) **Descreva o momento em que você encontrou a maior dificuldade lógica.** O bug mais complicado ocorreu na tela de vendas, ao tentar exibir o alerta de estoque baixo (ícone ) na lista de produtos.
- b) **O que você esperava que o código fizesse e o que ele realmente estava fazendo?** Eu esperava que o ícone de alerta aparecesse ao lado dos produtos com estoque baixo. No entanto, o ícone aparecia para *todos* os produtos, ou para *nenhum*.
- c) **Qual foi o seu processo de pensamento para depurar e encontrar a solução?**
Meu processo foi:
 1. Verifiquei o HTML (`vendas.html`) para garantir que o `{% if produto.quantidade < 5 %}` estava correto. Estava.
 2. Isso me levou a crer que a variável `produto.quantidade` estava errada.
 3. Usei `print(produtos_ordenados)` no `app.py`, logo antes de enviar os dados para o template.
 4. Descobri que, na minha consulta SQL, eu estava trazendo o nome e o ID, mas tinha esquecido de incluir a `quantidade` na consulta (`SELECT`).
 5. A solução foi ajustar a consulta SQL na rota `registrar_venda` para `SELECT id_produto, nome, preco, quantidade FROM Produtos` e reconstruir a lista de dicionários com todos os dados necessários para o template.

4. Detalhe da Validação

- a) **Em qual arquivo e linha (aproximadamente) está essa validação?** A validação do **Preço > 0** está no arquivo `app.py`, dentro das rotas `add_produto` e `update_produto` (aproximadamente, linhas 110 e 135).
- b) **A validação acontece no frontend ou no backend? Por que você escolheu fazer dessa forma?** A validação acontece no **Backend**. Embora uma validação de frontend (ex: `min="0.01"` no HTML) seja útil para a experiência do usuário, a validação no backend (no `app.py`) é **essencial para a segurança e integridade dos dados**. Um usuário poderia facilmente contornar uma validação em HTML, mas ele não pode contornar a verificação no servidor antes que os dados sejam salvos no banco de dados.

5. Autoavaliação e Próximos Passos

- **Se você tivesse mais três horas, qual seria a primeira funcionalidade que você adicionaria?** A melhoria mais importante seria um **Relatório de Vendas no Dashboard**. Atualmente, o sistema registra vendas, mas não oferece ao gerente nenhuma análise. Eu adicionaria um gráfico ou uma tabela no Dashboard mostrando:
 1. O total vendido no dia (R\$).

2. Os 5 produtos mais vendidos no mês.
3. Um alerta de "Produtos mais vendidos com estoque baixo". Essa seria a melhoria mais crucial, pois transforma o sistema de um simples "coletor de dados" em uma "ferramenta de decisão" para o gerente saber o que comprar e o que está dando lucro.