Guia de Implementação: Sistema de Gerenciamento de Estoque

Iago Flávio Hertz Rafael Cauã Wendel

Table of contents

1 Introdução

Este guia orienta a implementação do Sistema de Gerenciamento de Estoque, abordando desenvolvimento, documentação e controle de versão.

2 Estrutura do Projeto

1. Classe `Produto`
2. Classe `Estoque` (utilizando lista encadeada)
3. Classe `LinkedList`
4. Funções de gerenciamento (adicionar, remover, atualizar, listar)

3 Implementação

Todas as funcionalidades abaixo serão implementadas em `main.py` visando alcançar o número mínimo de linhas pedidas pela professora.

3.1 Funcionalidades

3.1.1 Classe Produto (hertz)

3.1.2 Classe Estoque –> apenas inicializar (hertz)

3.1.3 Classe LinkedList –> apenas inicializar (hertz)

3.1.4 Funções Principais

1. `adicionar_produto()` (cauã)
2. `remover_produto()` (cauã)
3. `atualizar_produto()` (hertz)
4. `listar_produtos()` (iago)
5. `buscar_produto()` (iago)
6. `buscar_por_categoria(categoria)` (hertz) –> Busca mais complexa e específica.

7. `ordenar_por_quantidade()` (iago) –> Ordena a lista de produtos por quantidade (do menor para o maior).
8. `get_action(string)` (cauã)
9. `main()`(cauã)

3.1.5 Ciclo while True

Implementar while True na função main(), com opções a serem escolhidas abordando cada uma das funções implementadas, e uma opção para quebrar o ciclo, como o número "0" por exemplo.

Lembrar-se de incluir:

```
if __name__ == '__main__':
    main()
```

3.2 Responsabilidade de implementação do projeto pelos membros

A responsabilidade de cada um ainda será definida através de uma chamada via discord.

4 Boas Práticas

4.1 Comentários no Código

Usar comentários para explicar a lógica complexa e docstrings para funções e classes.

Exemplo:

```
def buscar_produto(self, id):
    """
    Busca um produto no estoque pelo ID.

    Args:
        id (int): O ID do produto a ser buscado.

    Returns:
        Produto: O produto encontrado ou None se não existir.
    """
    # Implementação da busca
```

4.2 Tratamento de Erros

Usar blocos try/except para lidar com possíveis erros:

```
try:
    produto = estoque.buscar_produto(id)
    if produto:
        print(f"Produto encontrado: {produto.nome}")
    else:
        print("Produto não encontrado.")
except Exception:
    print(f"Erro ao buscar produto.")
```

5 Controle de Versão com GitHub

5.1 Configuração Inicial

Clone o repositório localmente: `git clone (link_do_repositório) --> O link ainda será enviado no grupo do whatsapp do projeto pois o repositório ainda não foi criado.`

5.2 Fluxo de Trabalho

1. Criar uma branch para cada nova feature:

   ```
   git checkout -b feature/adicionar-produto
   ```

2. Fazer commits frequentes com mensagens descritivas:

   ```
   git commit -m "Implementa função de adicionar produto"
   ```

3. Fazer um push para o GitHub:

   ```
   git push origin feature/adicionar-produto
   ```

4. Abrir um Pull Request para revisão do código

5.3 Documentação de Erros

1. Usar as Issues do GitHub para rastrear erros
2. Ao encontrar um bug:
    - Criar uma nova Issue descrevendo o problema
    - Adicionar labels relevantes (ex: “bug”, “high-priority”)
    - Atribuir a um membro da equipe, ou você mesmo caso seja o responsável
3. Ao resolver um bug:
    - Referenciar o número da Issue no commit:

      ```
      git commit -m "Corrige erro na atualização de quantidade (#42)"
      ```

    - Fechar a Issue através do Pull Request ou manualmente

6 Apresentação

6.1 Estrutura da apresentação

A apresentação será feita de acordo com a seguinte ordem:

1. Introdução
2. Objetivos do Projeto
3. Estruturas Utilizadas
4. Demonstração das Classes Produto e Estoque
5. Funções Principais
6. Demonstração do Código Rodando na IDE
7. Mostrar os desafios enfrentados
8. Soluções adotadas pelo projeto
9. Conclusão e Perguntas

6.2 Falas dos membros

Atenção: As seguintes falas servem de guia para o que será falado. Fique a vontade para incrementar mais coisas de acordo com o seu entendimento do projeto.

6.2.1 Iago Flávio

(A ser definido)

6.2.2 Hertz Rafael

(A ser definido)

6.2.3 Cauã Wendel

(A ser definido)

6.3 Ordem das falas

A ordem das falas ainda será definida de acordo com a implementação do projeto por parte de cada um.

7 Conclusão

Seguindo estas diretrizes, será desenvolvido um Sistema de Gerenciamento de Estoque com funçõe básicas bem documentado. Lembrar-se de comunicar regularmente, fazer commits frequentes e manter a documentação atualizada.