Descrição do Projeto

Título: Sistema de Gestão de Tarefas

Objetivo: Desenvolver um sistema simples de gestão de tarefas onde os usuários podem criar, visualizar, atualizar e excluir tarefas. O projeto deve ser containerizado usando Docker.

Requisitos

1. **Configuração Inicial:**
    - **Crie um repositório Git para o projeto e compartilhe o link.**
    - **Configure um ambiente virtual para o projeto e documente o processo de instalação das dependências no README.**
2. **Funcionalidades Básicas:**
    - **Criação de Tarefas: Permita que o usuário crie novas tarefas com um título e uma descrição.**
    - **Listagem de Tarefas: Exiba todas as tarefas criadas em uma lista.**
    - **Atualização de Tarefas: Permita que o usuário edite o título e a descrição de uma tarefa existente.**
    - **Exclusão de Tarefas: Permita que o usuário exclua uma tarefa.**
3. **Funcionalidades Adicionais:**
    - **Marcar Tarefas como Concluídas: Adicione a opção de marcar uma tarefa como concluída e filtre a lista para mostrar apenas as tarefas não concluídas, se desejado.**
    - **Data de Criação e Conclusão: Registre a data e hora de criação e, se aplicável, a data e hora de conclusão de uma tarefa.**
4. **Interface:**
    - **CLI (Interface de Linha de Comando): Desenvolva uma interface de linha de comando para interagir com o sistema.**
    - **Ou: Web (Usando Flask): Desenvolva uma interface web simples utilizando Flask.**
5. **Persistência de Dados:**
    - **Armazene os dados em um banco de dados PostgreSQL, MariaDB ou em um arquivo JSON.**
6. **Docker:**
    - **Crie um `Dockerfile` para containerizar a aplicação.**
    - **Crie um arquivo `docker-compose.yml` para configurar e iniciar o container da aplicação e do banco de dados (se aplicável).**
    - **Documente o processo de build e execução do container no README.**
7. **Testes:**
    - **Implemente testes unitários para as funcionalidades principais do sistema.**

Avaliação

1. **Código e Documentação:**
    - **Clareza e organização do código.**
    - **Comentários e documentação, incluindo um README detalhado.**
    - **Uso adequado de Git e commits significativos.**
2. **Funcionalidade:**
    - **Implementação correta de todas as funcionalidades exigidas.**
    - **Ausência de bugs e erros.**
3. **Boas Práticas:**
    - **Adesão a boas práticas de programação em Python (PEP8).**
    - **Estruturação do projeto de forma modular.**
4. **Docker:**
    - **Correção e eficiência do `Dockerfile`.**
    - **Uso correto do `docker-compose.yml`.**

    - **Capacidade de iniciar o ambiente de desenvolvimento usando Docker sem problemas.**

Entrega

- **O candidato deve enviar o link do repositório Git contendo o código do projeto, a documentação e os arquivos Docker necessários.**
- **Também deve envia o captura de tela ou vídeo do projeto funcionando.**

Dicas para o Desenvolvedor

- **Utilize bibliotecas populares e bem documentadas quando necessário (por exemplo, Flask para a interface web, click para CLI).**
- **Preste atenção à experiência do usuário, especialmente na interface.**
- **Escreva testes para garantir a qualidade do código e a funcionalidade correta.**
- **Certifique-se de que o ambiente Docker funcione corretamente e facilite a configuração e execução da aplicação.**