

Objetivos Gerais

- **Avaliar** o conhecimento e a proficiência do candidato em **Python / FastApi**.
 - **Testar** a capacidade de criar APIs RESTful funcionais e seguras.
 - **Analisar** o entendimento de conceitos de autenticação e autorização sem depender de bibliotecas externas.
 - **Observar** as boas práticas de codificação, organização do código e tratamento de erros.
-

Teste para Python / FastApi

Descrição do Projeto

Desenvolver uma API RESTful simples para gerenciamento de tarefas (To-Do List), utilizando Python com FastApi

Requisitos do Projeto

1. **Operações CRUD:**
 - o **Create:** Permitir a criação de novas tarefas com campos como título e descrição.
 - o **Read:** Permitir a listagem de todas as tarefas e a obtenção de uma tarefa específica por ID.
 - o **Update:** Permitir a atualização dos detalhes de uma tarefa existente.
 - o **Delete:** Permitir a exclusão de uma tarefa.
2. **Autenticação Simples:**
 - o Implementar uma autenticação básica utilizando tokens simples gerados manualmente.
 - o O usuário deve enviar o token em um header HTTP para acessar as rotas protegidas.
 - o Não é necessário implementar sistema de registro ou login completo; um token estático pode ser usado para simplificar.
3. **Estrutura do Projeto:**
 - o Organizar o código de forma clara e compreensível, separando as rotas da lógica de negócio.
4. **Persistência de Dados:**
 - o Utilizar banco de dados PostgreSQL
 - o Criar modelagem de dados
 - o Implementar interações básicas de criação, leitura, atualização e exclusão dos dados no banco de dados
5. **Tratamento e Validação de Dados:**
 - o Validar dados de entrada para garantir que todas as requisições contenham as informações necessárias e no formato correto.
 - o Fornecer mensagens de erro claras e códigos de status HTTP adequados em caso de validações falhadas.
6. **Estrutura e Boas Práticas:**

- o Organizar o projeto seguindo uma estrutura modularizada, separando modelos, rotas e serviços.
- o Escrever código limpo e bem documentado, seguindo as convenções de estilo do Python.

7. **Documentação:**

- o Fornecer um arquivo README.md detalhado com instruções de instalação, execução e exemplos de uso da API.
- o Incluir documentação das rotas e exemplos de requisições e respostas.

Critérios de Avaliação

- **Implementação correta** das funcionalidades de CRUD e das rotas.
 - **Persistência eficaz** dos dados no banco de dados PostgreSQL.
 - **Validação e tratamento de dados** robustos.
 - **Estruturação e organização** do código seguindo boas práticas.
 - **Documentação completa** e clara.
-