

Desafio de Desenvolvimento Full-Stack – To-Do List Python

Objetivo

Desenvolver uma aplicação simples de gerenciamento de tarefas (To-Do List), utilizando Python como linguagem principal, integrando Back-end (API), Front-end, banco de dados, e funcionalidades de segurança. O objetivo é avaliar a capacidade técnica em Python, assim como habilidades com tecnologias complementares.

Requisitos

- Back-end (Python):

- Utilize Flask ou FastAPI para criar uma API que permita:
- Adicionar uma nova tarefa.
- Atualizar o status de uma tarefa (pendente/completa).
- Remover uma tarefa.
- Listar todas as tarefas.
- Use Redis para cache de tarefas ou sessões, além de um banco de dados relacional ou não de sua escolha (MySQL, PostgreSQL, MongoDB, etc.) para armazenar as tarefas de maneira persistente.
- Implemente validação de dados para garantir que as informações das tarefas estejam corretas e evitar erros (ex.: uma tarefa não pode ter título vazio).

- Front-end:

- Crie uma interface simples em HTML e CSS que se comunique com a API via JavaScript (AJAX).
- A interface deve permitir:
- Adicionar novas tarefas.
- Atualizar o status de uma tarefa.

- Remover uma tarefa.
- Listar todas as tarefas.

- Segurança:

- Implementar medidas básicas de segurança, como a sanitização das entradas para evitar injeção de código e uso de HTTPS.
- Utilizar autenticação básica (usuário e senha) com JWT (JSON Web Token) para garantir que apenas usuários autenticados possam adicionar ou remover tarefas.

- Deploy e Cloud Computing:

- Implantar a aplicação em uma plataforma de nuvem (AWS, Google Cloud, ou Azure).
- A aplicação deve ser acessível publicamente via HTTPS.
- Utilize algum serviço de armazenamento em nuvem para gerenciar arquivos estáticos (se houver).

- Banco de Dados:

- Gerenciar os dados usando Redis (para dados em cache) e outro banco de dados para persistência.

- Git & GitHub:

- Utilizar Git para versionamento do código. Todo o desenvolvimento deve ser feito em um repositório público no GitHub. Envie o link do repositório ao final do desafio.

Avaliação

O projeto será avaliado com base nos seguintes critérios:

- Qualidade do código Python (organização, clareza, boas práticas de PEP8).
- Implementação da API e das funcionalidades de gerenciamento de tarefas.
- Integração adequada do Redis e do banco de dados relacional.

- Uso de práticas de segurança no desenvolvimento da aplicação.
- Capacidade de deploy em uma plataforma de nuvem.
- Uso correto do Git e GitHub para controle de versão.
- Implementação da interface de Front-end (mesmo que simples, deve ser funcional e interativa).

Tempo estimado

3 a 5 dias para conclusão.