# Desafio - To-Do List

Desenvolver uma aplicação de gerenciamento de tarefas (To-Do List), utilizando Python ou Node.js como linguagem principal no back-end. O objetivo é avaliar a capacidade técnica do desenvolvedor, assim como habilidades com tecnologias complementares e boas práticas de desenvolvimento.

## Requisitos

### Back-end:

1. **Framework:** Utilize um framework de sua escolha em Python ou JavaScript para construir a API.
2. **Usuarios:**
   * Permitir a criação de usuários (nome de usuário ou email e senha)
3. **Autenticação e Autorização:**
   * Implementar login simples (nome de usuário ou email e senha)
   * JWT (JSON Web Tokens) para controle de acesso
4. **Tarefas:**
   * Permitir as seguintes operações:
     + Adicionar uma nova tarefa
     + Atualizar o status de uma tarefa (pendente/completa)
     + Remover uma tarefa
     + Listar todas as tarefas
   * Cada tarefa deve conter:
     + Título
     + Descrição
     + Status
     + Data de criação
     + Data de atualização
5. **Validações:**
   * Implementar validação de dados para garantir integridade
   * Sanitização de entradas para evitar injeção de código
6. **Persistência de Dados:**
   * Utilize um banco de dados para armazenar as tarefas e dados de usuários, podendo ser:
     + Relacional: PostgreSQL, MySQL, etc.
     + Não relacional: MongoDB, etc.
7. **Cache:**
   * Utilize Redis para cache de informações de controle de acesso, garantindo recuperação rápida para autenticação e autorização e reduzindo a carga no banco de dados principal.

### Segurança:

* Implementar autenticação e autorização via JWT.
* Utilizar HTTPS.
* Sanitização de inputs.

### Deploy:

* Implementar a aplicação em uma plataforma de nuvem (AWS, GCP, Azure, Vercel, etc.).
* Aplicação acessível via HTTPS.

### 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.

## Entrega

### Prazo: 4 dias

### A entrega do projeto deve conter:

* **Vídeo:**
  + Vídeo de até 5 minutos contendo:
    - Breve apresentação do candidato, deve mostrar o rosto.
    - Apresentação do projeto, tecnologias escolhidas, estrutura, desafios, deploy, etc.
* **Repositório:**
  + Link para o repositório público no GitHub
  + README detalhado com:
    - Instruções de instalação
    - Documentação das rotas/endpoints
    - Tecnologias utilizadas e justificativas
* **Link da aplicação:**
  + Link para a aplicação rodando em nuvem.
  + Manter disponível por no mínimo 3 dias.

## Avaliação

O projeto será avaliado com base nos seguintes critérios:

* **Qualidade do código:** Organização, clareza, boas práticas (PEP8 para Python, convenções adequadas para JavaScript/Node.js), proeficiencia com as tecnologias escolhidas.
* **Implementação da API:** Estrutura, funcionalidades e gerenciamento adequado das tarefas.
* **Estrutura e persistência de dados:** Uso eficiente do Redis para cache e do banco de dados principal.
* **Práticas de segurança:** Implementação adequada de autenticação, autorização via JWT, e proteção de dados sensíveis.
* **Capacidade de deploy:** Aplicação corretamente implementada em nuvem e acessível via HTTPS.
* **Uso do Git e GitHub:** Controle de versão adequado e repositório organizado.
* **Front-end (se implementado):** Funcionalidade, usabilidade e interação com a API.
* **Documentação:** Clareza e completude do README e documentação da API.