

Desafio Técnico de Seleção de Estagiário Full Stack

Este desafio tem como objetivo avaliar seus conhecimentos técnicos em desenvolvimento web fullstack, incluindo:

- Frontend com ReactJS
- Criação e documentação de APIs RESTful com TypeScript
- Aplicação de padrões SOLID e Clean Architecture
- Autenticação com JWT
- Uso de banco de dados (SQLite + ORM)
- Componentização, hooks e gerenciamento de estado com Redux
- Testes unitários
- Ambiente Dockerizado
- Documentação técnica (README, Swagger)

Instruções:

- **Prazo:** entre 7 a 10 dias após primeira entrevista.
- **Repositório:** crie um mono repositório para o projeto e POC sugeridos, compartilhe com tecnico@everlabs.com.br
- **Domine todas as decisões técnicas:** saiba explicá-las: se usou IA ou algum recurso seja crítico em saber explicar como tal recurso/componente funciona – IA acelera mas não substitui nosso domínio sobre nossas escolhas técnicas.
- **Realize commits de suas interações e solicite apoio em caso de alguma dúvida técnica** - ficaremos felizes em ajudá-lo(a) e isto não afetará nossa decisão – demonstrará seu interesse e empenho – mas apenas se você realmente precisar de apoio – não exigido.

Descrição do Problema

Você deverá desenvolver um sistema de **Gestão de Tarefas Estilo SCRUM/Kanban**, com controle de usuários e permissões, baseado nas seguintes regras:

Usuários

- O sistema deve ter dois perfis de usuário: admin e user.
- O cadastro de usuários deve ser realizado via API (pode ser inserido manualmente no banco ou via rota protegida para admin).

Autenticação

- Implementar autenticação via **JWT**.
- O token deve ser usado para proteger as rotas que exigem login e controle de permissões por tipo de usuário.

Tarefas

- O admin pode:
 - Criar, editar e excluir tarefas.
 - Atribuir tarefas a usuários.
- A tarefa deve ter os seguintes atributos:
 - nome
 - data_inicio e data_fim
 - prioridade (de 0 a 10)
 - status (ver tabela)
- O user somente poderá interagir com tarefas atribuídas a ele, nos seguintes pontos:
 - Atualizar o status.
 - Registrar ocorrências (comentários).
 - Anexar arquivos.

Interações com Tarefas

- Tarefas devem permitir:
 - Comentários (ocorrências).
 - Upload de arquivos (mesmo que apenas salvos localmente).
 - Adição de etiquetas (tags).

Quadro Kanban (Diferencial)

- **Diferencial, não obrigatório:** Criar um quadro estilo Kanban com **drag and drop** entre colunas de status. Exemplos:
 - <https://codesandbox.io/p/sandbox/mui-kanban-forked-zdt8c3?file=%2Fsrc%2FApp.js>
 - <https://www.youtube.com/watch?v=gjrVCY3oPds>

Sugestões de Estudos

Restful API:

- [Complete Guide to Clean Architecture](#)
- [Clean Architecture com TypeScript & Node.js](#)
- [Clean Architecture & Typescript](#) (super recomendo fazer os tutoriais curtos de 1 a 10)

React JS:

- [Curso de React para Completos Iniciantes](#)
- [ReactJS Beginner Course 2025 | Become a React Pro in 1.5 Hours](#)
- [Clone do Netflix com React](#) e parte [2](#) (recomendo ao menos a parte 1)

Outros:

- <https://www.youtube.com/watch?v=CgkZ7MvWUAA>