

Teste para Vaga de Fullstack

Desafio Fullstack

Olá, desenvolvedor(a)!

Este desafio tem como objetivo avaliar sua capacidade de desenvolvimento fullstack, com foco em organização, domínio técnico e velocidade de entrega. Esperamos que você demonstre boas práticas de arquitetura, escrita de código limpo e controle de versão.

Descrição do Desafio

Você deverá desenvolver uma aplicação de quadro Kanban, composta por:

- Backend utilizando NestJS com Postgres.
- Frontend utilizando Angular.

Requisitos do Backend (NestJS + Postgres)

A API deve permitir:

1. Kanban

- Listar todas as tarefas.
- Criar uma nova tarefa.
- Editar uma tarefa.
- Alterar o status da tarefa (pending, in_progress, testing, done).
- Deletar uma tarefa.
- Mover uma tarefa entre projetos e colunas.
- Pagar listagens e filtrar tarefas por status/projeto.

2. Usuários e Autenticação

- CRUD de usuários.

- Autenticação JWT.
- Controle de acesso baseado em papéis (RBAC).

Especificações técnicas:

- NestJS com TypeScript.
- Postgres como banco de dados (local ou nuvem) com migrações.
- Organização modular (Clean Architecture é desejável).
- Código bem estruturado, legível e comentado.
- Validação de dados.
- Tratamento global de erros.
- Documentação da API (Swagger/OpenAPI).
- Testes automatizados (mínimo unitários).

Requisitos do Frontend (Angular)

A interface deve permitir:

- Exibir as tarefas separadas por status em colunas estilo Kanban.
- Criar uma nova tarefa por meio de um formulário.
- Editar tarefas existentes.
- Alterar o status das tarefas.
- Deletar tarefas.

Opcional

Implemente um sistema de comentários por tarefa (ex: adicionar/remover comentários), salvando no banco e exibindo no frontend. Isso nos ajudará a avaliar modelagem relacional e uso de relacionamentos no banco.

Especificações técnicas:

- Angular com estrutura de módulos.
- Componentes reutilizáveis.
- Integração com a API via HTTP.
- Layout responsivo.
- Pode utilizar bibliotecas modernas (NgRx, RxJS, etc.) como diferencial.

Entrega

1. Crie um repositório público no GitHub.
2. Suba o código do backend e do frontend, organizados em pastas separadas.
3. Inclua um arquivo README.md com instruções claras de como rodar a aplicação.
4. Envie o link do repositório para nossa equipe.

Critérios de Avaliação

- Organização da estrutura do projeto.
- Separação de responsabilidades (módulos, serviços, controllers, etc.).
- Clareza e legibilidade do código.
- Boas práticas de desenvolvimento (Clean Code, SOLID).
- Histórico de commits (uso contínuo e explicativo).
- Documentação no README.

Prazo de Entrega

Entrega até terça-Feira, 26 de Agosto de 2025, às 12h00.