

Desafio técnico

Desenvolvedor Back-end Jr

Pessoas
& Cultura

Crie uma conta no Github (caso ainda não tenha criado).

Instruções

- Utilizar apenas JavaScript para resolver os exercícios;
- Entrega: toda a solução deve estar em um único repositório (github, gitlab, bitbucket, etc...) de forma pública;
- Neste repositório deve haver, na raiz, um README.md, com as instruções de execução do código desenvolvido;
- Estilo fica ao critério do candidato, sugerimos o uso de linters para facilitar a correção (airBnB ou outro estilo de código), mas não é o foco principal do teste.

Exercício de Back-end

Desafio

O objetivo deste desafio é implementar uma API RESTful utilizando Node.js para gerenciar posts e comentários. A API deve permitir a criação, listagem, atualização e exclusão de posts e possibilitar que cada post tenha comentários associados. Cada rota deve estar autenticada, exigindo um token JWT para acesso.

Tecnologias e Ferramentas Sugeridas

- Node.js com Express para criação dos endpoints.
- Sequelize como ORM para abstração de dados.
- PostgreSQL como banco de dados relacional.
- JWT (JSON Web Token) para autenticação.

CRUD de Posts

- Inserir Post: Criar um novo post.
- Listar Posts: Retornar uma lista de todos os posts.
- Atualizar Post: Editar o conteúdo de um post existente.
- Deletar Post: Remover um post.

Desafio técnico

Desenvolvedor Back-end Jr

Pessoas
& Cultura

2. CRUD de Comentários para um Post

- Adicionar Comentário a um Post: Inserir um novo comentário associado a um post específico.
- Apagar Comentário de um Post: Remover um comentário específico de um post.

3. Autenticação e Autorização com JWT

- A API deve requerer autenticação via token JWT para todas as rotas de CRUD de posts e comentários.
- O token JWT deve expirar após 1 hora de sua criação.
- Para isso, será necessário implementar um middleware de autenticação que valide o token em cada requisição protegida.

4. Tabela de Usuários

Crie uma tabela chamada users para armazenar as informações de usuário, contendo:

- Nome do usuário
- Usuário (username) único para identificação e login
- Senha (armazenada de forma segura, por exemplo, com hash)

Requisitos Adicionais

Cadastro e Login de Usuários

- Endpoint para registro de usuários (nome, username e senha), permitindo que novos usuários se cadastrem.
- Endpoint para login de usuários que retorna o token JWT ao usuário após a autenticação, com validade de 1 hora.

Desafio técnico

Desenvolvedor Back-end Jr

Pessoas
& Cultura

PLUS

Dockerizar o ambiente.

Migrar o armazenamento e a validação da parte de autenticação (cadastro/login) para o MongoDB, mantendo o PostgreSQL como base principal para os demais dados da aplicação.

Desejamos sucesso!