

TESTE TÉCNICO - FASE 2

BACK-END

Requisitos Técnicos

- A API deve ser desenvolvida utilizando **Deno** com o framework **Hono**;
- O banco de dados deve ser **PostgreSQL**, utilizando **Drizzle ORM**;
- A validação de entrada e saída de dados deve ser feita com **Zod**.

Modelagem do Banco de Dados

As seguintes tabelas devem ser criadas:

- **Users**
- **Organizations**

Regras de Negócio

- Um **usuário** pode pertencer a **múltiplas organizações**;
- Um usuário pode ter **apenas uma organização ativa por vez**;
- As requisições para a API devem refletir os dados da **organização ativa**;
- Caso necessário, crie tabelas intermediárias para gerenciar relacionamentos.

Endpoints Requeridos

- **CRUD de Usuários**
- **CRUD de Organizações**

Observações

- A listagem de usuários deve oferecer **filtros e paginação**;
- A deleção de usuários deve suportar **remoção em lote**.

Funcionalidades Extras

- A implementação de um sistema de **autenticação** é um diferencial;
- Caso opte pela autenticação, defina uma abordagem para **proteger endpoints privados**.

FRONT-END

Requisitos Técnicos

- O front-end deve fornecer interfaces intuitivas e eficientes para interagir com os endpoints da API, permitindo a execução das ações necessárias de forma fluida e acessível;
- O projeto deve ser desenvolvido utilizando **React** com **Vite**;
- Para estilização, utilize **Tailwind CSS** com **shadcn**;
- As requisições à API devem ser feitas com **Axios** e gerenciadas utilizando **Tanstack Query**;
- Listagens e tabelas devem ser implementadas utilizando **Tanstack Table**.

Arquitetura e Modelagem

- Estruture a aplicação visando **reutilização de componentes** e **escalabilidade**;
- Organize o código de maneira modular, facilitando a manutenção e extensibilidade;
- Implementar paginação e filtragem de dados utilizando **parâmetros na API**;
- Utilize **React Hook Form** para manipulação de formulários e **Zod** para validação de dados.

Funcionalidades Extras

- Utilize **Tanstack Router** para gerenciamento de rotas;
- A implementação de um sistema de **autenticação** é um diferencial;
- Caso a autenticação seja implementada, utilize **Zustand** para gerenciamento de estado global e implemente **rotas privadas** na aplicação.