

# Exercício: Desenvolvimento de uma Aplicação CRUD

**\*\*Título:\*\*** Desenvolvimento de uma Aplicação CRUD com React, Vite, TypeScript e Json-Server

**\*\*Objetivo:\*\*** O objetivo deste exercício é que o estudante desenvolva uma aplicação web utilizando as tecnologias React, Vite, TypeScript e Json-Server para criar um sistema CRUD (Create, Read, Update, Delete) para gerenciar uma entidade fictícia.

**\*\*Descrição do Projeto:\*\***

Você deverá desenvolver uma aplicação web para gerenciar uma entidade de sua escolha, como por exemplo, "Livros", "Filmes", "Produtos", "Clientes" ou qualquer outra entidade relevante para o contexto da sua área de estudo ou interesse. A entidade deve conter exatamente cinco variáveis (atributos), que serão manipuladas através das funcionalidades de CRUD.

**\*\*Requisitos do Projeto:\*\***

## 1. **\*\*Tecnologias a Utilizar:\*\***

- React para construção da interface de usuário.
- Vite para a criação e desenvolvimento do projeto.
- TypeScript para tipagem estática e melhor desenvolvimento do código.
- Json-Server para simular uma API RESTful que armazene os dados da sua aplicação.

## 2. **\*\*Funcionalidades Obrigatórias:\*\***

- **\*\*Criação (Create):\*\*** Permitir a adição de novos registros à entidade.
- **\*\*Leitura (Read):\*\*** Exibir a lista de registros existentes com detalhes de cada um.

- **\*\*Atualização (Update):\*\*** Permitir a edição de um registro existente.
- **\*\*Exclusão (Delete):\*\*** Permitir a remoção de um registro.

### 3. **\*\*Detalhamento das Variáveis:\*\***

- A entidade deve conter cinco variáveis que serão manipuladas pelo usuário. Exemplo para a entidade "Livros":

- ID (gerado automaticamente)
- Título (string)
- Autor (string)
- Ano de Publicação (number)
- Gênero (string)
- Número de Páginas (number)

### 4. **\*\*Layout e Navegação:\*\***

- Crie uma interface amigável que permita a navegação fácil entre as diferentes funcionalidades de CRUD.
- Utilize componentes reutilizáveis para a criação de formulários e listas.
- O layout deve ser responsivo e adaptado para diferentes tamanhos de tela.

### 5. **\*\*Configuração do Json-Server:\*\***

- Configure o Json-Server para simular a API que armazenará os dados da entidade.
- Os dados devem ser persistidos em um arquivo `db.json` na raiz do projeto.
- Certifique-se de que as operações de CRUD interajam corretamente com a API do Json-Server.

### 6. **\*\*Entrega:\*\***

- O projeto deve ser entregue em um repositório Git, com um README detalhando as etapas para execução do projeto.

- Inclua instruções claras sobre como iniciar o Json-Server e como executar a aplicação.

**\*\*Critérios de Avaliação:\*\***

- Funcionamento completo das funcionalidades de CRUD.
- Uso correto de React, Vite, TypeScript e Json-Server.
- Qualidade e clareza do código.
- Organização do projeto e repositório.
- Interface de usuário funcional e intuitiva.

**\*\*Desafio Extra:\*\***

- Adicione validações nos formulários de criação e edição para garantir que os dados inseridos pelo usuário sejam válidos.
- Implemente uma funcionalidade de pesquisa ou filtro para facilitar a localização de registros específicos.

Este exercício tem como objetivo aprimorar suas habilidades de desenvolvimento front-end, integração com APIs e boas práticas de programação. Bons estudos e bom desenvolvimento!