

# Apresentação do Projeto - Sistema de Gerenciamento de Quadras de Tênis

## Visão Geral

Este projeto implementa um sistema completo de gerenciamento de quadras de tênis, seguindo os requisitos do desafio técnico. O sistema permite:

- Cadastrar novas quadras
- Listar todas as quadras, com opção de filtrar apenas as disponíveis
- Editar informações de quadras existentes
- Alterar a disponibilidade de uma quadra
- Excluir quadras

## Stack Tecnológica

### Backend

- **Node.js** com **Express** para a API RESTful
- **Sequelize** como ORM para interação com o banco de dados
- **PostgreSQL** como sistema de gerenciamento de banco de dados
- **Swagger** para documentação da API
- **Jest** para testes automatizados

### Frontend

- **React** para a interface do usuário
- **React Router** para navegação
- **Bootstrap** para estilização
- **Axios** para comunicação com a API
- **React Testing Library** para testes de componentes

## Estrutura do Projeto

O projeto está organizado em duas partes principais:

## Backend

```
backend/  
├── src/  
│   ├── config/           # Configurações (banco de dados, Swagger)  
│   ├── controllers/      # Lógica de negócio  
│   ├── middlewares/      # Validação e tratamento de erros  
│   ├── models/           # Modelos do Sequelize  
│   ├── routes/           # Definição de rotas  
│   └── app.js             # Configuração do Express  
├── tests/                 # Testes automatizados  
├── database.sql           # Script SQL para criação da tabela  
└── server.js              # Ponto de entrada da aplicação
```

## Frontend

```
frontend/  
├── src/  
│   ├── components/       # Componentes React  
│   ├── services/         # Serviços para comunicação com a API  
│   ├── tests/            # Testes de componentes  
│   └── App.js             # Componente principal e rotas  
└── public/                # Arquivos estáticos
```

## Funcionalidades Implementadas

### 1. Banco de Dados

- Tabela `courts` com todos os campos requeridos
- Trigger para atualização automática do campo `updated_at`

### 2. API RESTful

- Endpoints para todas as operações CRUD
- Validação de dados de entrada
- Tratamento de erros
- Documentação com Swagger

### 3. Frontend

- Interface responsiva com Bootstrap
- Listagem de quadras com filtro de disponibilidade
- Formulário para criação e edição de quadras

- Botões para alterar disponibilidade e excluir quadras

## Diferenciais Implementados

- **Documentação com Swagger:** API completamente documentada e interativa
- **Middlewares para validação e tratamento de erros:** Garantindo consistência e segurança
- **Testes automatizados:** Cobertura de testes para backend e frontend
- **Estrutura organizada:** Seguindo padrões de arquitetura e boas práticas

## Demonstração

### Listagem de Quadras

A tela principal exibe todas as quadras cadastradas, com opção para filtrar apenas as disponíveis. Cada quadra mostra seu nome, localização e status de disponibilidade.

### Criação e Edição de Quadras

O formulário permite adicionar novas quadras ou editar as existentes, com validação de campos obrigatórios.

### Gerenciamento de Disponibilidade

Botões para alternar rapidamente a disponibilidade de cada quadra, facilitando o gerenciamento.

## Conclusão

O sistema desenvolvido atende a todos os requisitos do desafio, implementando um CRUD completo e funcional para gerenciamento de quadras de tênis. A arquitetura escolhida permite fácil manutenção e extensão, seguindo boas práticas de desenvolvimento de software.