

# BJJ School – Sistema de Cadastro de Escola de Jiu-Jitsu

## Relatório Técnico Acadêmico

**Autor:** Domingos Caldas de Oliveira Junior

**Professor:** Willian Almeida Rodrigues

**Disciplina:** Desenvolvimento de Aplicações Interativas com React [25E4\_2]

**Instituição:** INFNET – Instituto de Tecnologia

**Ano:** 2025

---

## Sumário

1. Resumo Executivo
  2. Objetivo do Projeto
  3. Features Implementadas
  4. Tecnologias Utilizadas
  5. Arquitetura do Projeto
  6. Instruções de Execução
  7. Critérios de Avaliação (Rubrica)
  8. Considerações Finais
  9. Créditos e Fontes
- 

## 1. Resumo Executivo

O presente relatório apresenta o desenvolvimento do sistema **JJ School**, uma aplicação CRUD completa criada com **React + Vite**, aplicada ao contexto de uma escola de jiu-jitsu.

O sistema permite cadastrar, editar, listar e excluir alunos, integrar dados externos e navegar entre páginas de forma fluida e responsiva.

O projeto sintetiza os conhecimentos adquiridos nas cinco primeiras aulas, aplicando boas práticas de **componentização, gerenciamento de estado, integração com APIs e estilização com Material UI**.

---

## 2. Objetivo do Projeto

O objetivo do projeto é desenvolver uma aplicação web moderna e funcional, capaz de gerenciar o cadastro de alunos de uma escola de jiu-jitsu.

A aplicação demonstra a aplicação prática dos seguintes conceitos:

- **Hooks e reatividade (useState, useEffect, Context API)**
- **Boas práticas de JavaScript ES6+ (arrow functions, destructuring, spread)**

- Reutilização de componentes e modularização
  - Integração com API pública (JSONPlaceholder)
  - Navegação entre páginas com React Router
  - Interface estilizada e responsiva com Material UI
  - Gerenciamento global de estado com Redux Toolkit
- 

### 3. Features Implementadas

#### Feature I – JavaScript moderno e Interface com React

- Estrutura modular com **componentes reutilizáveis** (StudentFormFields, StudentsTable, NavLinkButton).
  - Sintaxe moderna: **arrow functions, destructuring e template literals**.
  - Interface com **Material UI**, garantindo responsividade e acessibilidade.
  - Separação entre lógica de negócio e interface (arquitetura em camadas).
- 

#### Feature II – Gerenciamento de dados, reatividade e manipulação de listas

- **Context API** e **Redux Toolkit** para gerenciamento de estado.
  - **Hook customizado** (`useLocalStorage`) para persistência dos dados localmente.
  - **Renderização condicional** entre modos de cadastro e edição.
  - **Formulários controlados com React Hook Form**, com validações e mensagens de erro.
- 

#### Feature III – Integração com API externa e navegação

- Consumo da API pública **JSONPlaceholder** via **Axios**.
  - Uso de **Promise.race** e **AbortController** para controle de requisições assíncronas.
  - Tratamento de erros e estados de carregamento (**Alert**, **CircularProgress** do **MUI**).
  - Navegação entre páginas com **React Router** e **rota privada simulada** (`/admin`).
- 

### 4. Tecnologias Utilizadas

Categoria	Ferramenta / Biblioteca
Framework principal	<a href="#">React</a> (via <a href="#">Vite</a> )
UI / Estilo	<a href="#">Material UI</a>
Formulários	<a href="#">React Hook Form</a>

Categoria	Ferramenta / Biblioteca
Estado Local	Hooks e Context API
Estado Global	<a href="#">Redux Toolkit</a>
Persistência Local	localStorage (hook customizado <code>useLocalStorage</code> )
API Externa	<a href="#">Axios</a> + <a href="#">JSONPlaceholder</a>
Navegação	<a href="#">React Router DOM</a>
IDE / Ambiente	Visual Studio Code + Node.js + Vite Dev Server

---

## 5. Arquitetura do Projeto

```

src/
  └── components/
    ├── StudentFormFields.jsx
    ├── StudentsTable.jsx
    └── NavLinkButton.jsx
  └── context/
    └── AlunosContext.jsx
  └── hooks/
    └── useLocalStorage.js
  └── layout/
    └── AppShell.jsx
  └── pages/
    ├── Home.jsx
    ├── Students.jsx
    ├── Comunicados.jsx
    └── Admin.jsx
  └── services/
    └── api.js
  └── store/
    ├── index.js
    └── uiSlice.js
  └── theme/
    └── theme.js
  └── App.jsx
  └── main.jsx
  └── index.html

```

---

## 6. Instruções de Execução

### 1 Clonar o repositório

```
git clone https://github.com/domigosjr/bjj-school.git
cd bjj-school
```

### 2 Instalar dependências

```
npm install
```

### 3 Executar em modo desenvolvimento

```
npm run dev
```

Acesse <http://localhost:5173>.

## 4 Build de produção

```
npm run build
```

## 5 Pré-visualização do build

```
npm run preview
```

---

## 7. Critérios de Avaliação (Rubrica)

Critério	Evidência
Configuração do ambiente	Projeto criado com Vite e estrutura modular
Uso de JSX e ES6	Arrow functions, destructuring e spread
Reutilização de componentes	Form, Tabela, Navegação
Redux e Hooks	useSlice, useAlunos, useLocalStorage
Estilo e UI	Material UI aplicado integralmente
Consumo de API	JSONPlaceholder via Axios
Tratamento de erros	Alertas e loaders com MUI
Navegação	React Router com rota privada
Acessibilidade	Labels e responsividade do Material UI

---

## 8. Considerações Finais

O desenvolvimento do **JJ School** consolidou os principais fundamentos do React aprendidos durante o módulo.

O projeto atende aos critérios de arquitetura, código moderno, responsividade e boas práticas de integração com APIs.

Durante o processo, foram aplicados **conceitos de Hooks, Context API, Redux, React Router, React Hook Form e MUI**, resultando em uma aplicação completa e didaticamente robusta.

O aprendizado adquirido reforça a capacidade de planejar, estruturar e desenvolver aplicações web modernas e escaláveis, integrando teoria e prática com excelência técnica.

---

## 9. Créditos e Fontes

**Autor:** Domingos Caldas de Oliveira Junior

**Professor:** Willian Almeida Rodrigues

**Disciplina:** Desenvolvimento de Aplicações Interativas com React [25E4\_2]

**Instituição:** INFNET – Instituto de Tecnologia

**Ano:** 2025

---

## Fontes e Referências Técnicas

1. **React Official Documentation** – <https://react.dev>
  2. **Vite – Next Generation Frontend Tooling** – <https://vitejs.dev>
  3. **Material UI (MUI)** – <https://mui.com>
  4. **React Hook Form** – <https://react-hook-form.com>
  5. **Redux Toolkit** – <https://redux-toolkit.js.org>
  6. **React Router DOM** – <https://reactrouter.com>
  7. **Axios HTTP Client** – <https://axios-http.com>
  8. **JSONPlaceholder API** – <https://jsonplaceholder.typicode.com>
  9. **MDN Web Docs (ES6+, Hooks, Fetch API)** – <https://developer.mozilla.org>
  10. **OpenAI ChatGPT (GPT-5)** – Ferramenta utilizada para apoio técnico e geração de documentação, respeitando diretrizes acadêmicas de transparência e citação.
- 

## Agradecimentos

Agradeço ao professor **Willian Almeida Rodrigues** pelas orientações e feedbacks durante o desenvolvimento das atividades práticas, e ao **Instituto INFNET** pela qualidade metodológica e técnica do programa de pós-graduação em Engenharia de Software com Java.