

# Projeto Front-End: app\_frontprodutos Descrição

Projeto Front-End em ReactJS para consumir a API REST de gerenciamento de Categorias e Produtos. Implementa CRUD completo (Create, Read, Update, Delete) usando Fetch API nativa, integrado ao Back-End desenvolvido em JDBC.

## Autores

**Nome:** Jean Vinicius Rodrigues de Oliveira

**Matrícula:** 25173261

## Sistema de Cadastro de Produtos e Categorias

### Guia Completo de Execução do Projeto

Sistema de Cadastro de Categorias e Produtos Back-End (Java + Spark) | Front-End (React) | Banco de Dados (MySQL)

#### 1. Visão Geral do Projeto

Este repositório contém duas aplicações:

Back-End (Java + Spark + JDBC) Responsável por se conectar ao banco de dados MySQL e fornecer todas as APIs REST de Categorias e Produtos.

Front-End (ReactJS) Interface gráfica do usuário que consome as APIs do Back-End utilizando a Fetch API.

A comunicação entre Front-End e Back-End ocorre via HTTP, utilizando JSON.

#### 2. Configuração do Banco de Dados (MySQL)

Este projeto depende de um banco de dados MySQL. Antes de iniciar qualquer parte da aplicação, é necessário:

Ter o MySQL instalado

Criar o banco

Criar as tabelas

Garantir que o Back-End consiga se conectar

## 2.1. Criando o Banco e as Tabelas

Execute no MySQL Workbench, DBeaver ou terminal:

```
CREATE DATABASE IF NOT EXISTS aulajdbc; USE aulajdbc;
```

```
CREATE TABLE IF NOT EXISTS categorias ( id BIGINT PRIMARY KEY  
AUTO_INCREMENT, nome VARCHAR(255) NOT NULL );
```

```
INSERT INTO categorias (nome) VALUES ('Eletrônicos'), ('Livros'), ('Alimentos');
```

```
CREATE TABLE IF NOT EXISTS produtos ( id BIGINT PRIMARY KEY  
AUTO_INCREMENT, nome VARCHAR(255) NOT NULL, preco DOUBLE NOT  
NULL, estoque INT NOT NULL, id_categoria BIGINT, FOREIGN KEY (id_categoria)  
REFERENCES categorias(id) );
```

## 2.2. Configuração de Acesso ao MySQL no Back-End

No arquivo ConnectionFactory.java

```
private static final String URL = "jdbc:mysql://localhost:3306/aulajdbc"; private static  
final String USUARIO = "root"; private static final String SENHA = "sua_senha_aqui";
```

Altere SENHA para a senha do seu MySQL.

## 3. Executando o Back-End (Java + Spark)

O Back-End fica dentro da pasta:

/Back-End

### 3.1. Instalar dependências

Necessário ter Java 17 e Maven instalados.

### 3.2. Iniciar o servidor Iniciar o arquivo com nome API.java

Da um Run java para rodar o servidor.

Se estiver tudo correto, o console exibirá algo como:

Servidor Spark iniciado na porta 4567

### 3.3. Endereço da API <http://localhost:4567>

Exemplos:

Listar categorias → GET /categoria

Criar categoria → POST /categoria

Editar categoria → PUT /categoria/{id}

Excluir categoria → DELETE /categoria/{id}

## 4. Executando o Front-End (ReactJS)

O Front-End fica dentro de:

/Front-End

### 4.1. Instalar dependências

Você precisa ter Node.js instalado.

Dentro da pasta Front-End:

npm install

### 4.2. Iniciar o projeto npm start

O front abrirá em:

<http://localhost:3000>

## 5. Comunicação Front-End ↔ Back-End

No arquivo:

/src/api/index.js

A URL da API deve ser:

```
const BASE = "http://localhost:4567";
```

Todas as requisições usam o método request:


```
async function request(path, options = {}) { const res = await fetch(`${BASE}${path}`, {  
  headers: { "Content-Type": "application/json" }, ...options, });
```

```
if (res.status === 204) return null; return res.json(); }
```

É esta função que faz GET, POST, PUT e DELETE.

## 6. Estrutura Completa do Projeto

pgsql

 Copiar código

```
app_frontProdutos/  
├── Back-End/  
│   ├── src/main/java/api/  
│   │   ├── ApiCategoria.java  
│   │   ├── ApiProduto.java  
│   │   └── Conexao.java  
│   └── pom.xml  
└── Front-End/  
    ├── public/  
    ├── src/  
    │   ├── api/index.js  
    │   ├── components/Sidebar.js  
    │   ├── pages/  
    │   │   ├── Home/  
    │   │   ├── CadastroCategoria/  
    │   │   ├── CadastroProduto/  
    │   │   └── Lista/  
    │   ├── styles/  
    │   │   ├── Sidebar.css  
    │   │   ├── categoria.css  
    │   │   └── home.css  
    │   └── App.js  
    └── package.json
```

### 6. Ordem Correta para Rodar o Projeto

Criar o banco e tabelas no MySQL

Ajustar usuário e senha no Conexao.java

Rodar o Back-End com Maven

Confirmar que a API está ativa em <http://localhost:4567>

Instalar dependências do Front-End

Rodar o React em <http://localhost:3000>

Usar a interface para cadastrar, editar e excluir categorias e produtos

### 7. Funcionalidades Implementadas Categorias

Listagem de categorias

Cadastro de nova categoria

Edição de categoria existente

Exclusão de categoria

Requisições usando GET, POST, PUT, DELETE

Produtos

Cadastro de produto

Listagem com valores formatados em Real

Edição e exclusão