

# Documentação do Projeto - Arquitetura e Estrutura

---

## Visão Geral

Este projeto é composto por duas partes principais: o **Frontend** e o **Backend**. O backend é responsável pela manipulação de dados e pela criação de uma API RESTful para interação com um banco de dados MongoDB. O frontend é uma aplicação React Native que interage com essa API para fornecer uma interface amigável ao usuário.

## Estrutura do Projeto

### /project-root

```
|
| └── /frontend - Contém o código do aplicativo React Native (UI/UX)
|   | └── /src
|   |   | └── /components - Componentes reutilizáveis
|   |   |   | └── ProductCard.js - Exibe informações do produto de forma compacta
|   |   |   | └── ProductForm.js - Formulário para adicionar ou editar produtos
|   |   |   | └── /screens
|   |   |   |   | └── HomeScreen.js - Tela inicial que lista os produtos
|   |   |   |   | └── ProductDetailsScreen.js - Tela para visualizar e editar detalhes do produto
|   |   |   |   | └── /services
|   |   |   |   |   | └── api.js - Serviço que faz a integração com a API do backend
|   |   |   |   |   | └── App.js - Configuração da navegação e estrutura do aplicativo
|   |   |   |   |   | └── /backend - Contém o código da API (Node.js/Express)
|   |   |   |   |   |   | └── /models
|   |   |   |   |   |   |   | └── Product.js - Modelo de dados para produtos
|   |   |   |   |   |   |   | └── /controllers
|   |   |   |   |   |   |   |   | └── productController.js - Lógica de controle para as operações de CRUD
|   |   |   |   |   |   |   |   | └── /routes
|   |   |   |   |   |   |   |   |   | └── productRoutes.js - Definição das rotas para a API de produtos
|   |   |   |   |   |   |   |   |   | └── server.js - Arquivo principal do servidor Express
|   |   |   |   |   |   |   |   |   | └── README.md - Documentação do projeto
```

---

## Backend

### 1. Estrutura do Backend

O backend é construído com **Node.js** e **Express**, usando o **MongoDB** para armazenamento de dados.

- **server.js**: Arquivo principal que inicia o servidor e configura a conexão com o banco de dados MongoDB.

- **/models/Product.js**: Define o modelo de dados para o produto, incluindo os campos nome, descrição, quantidade e imagem.
- **/controllers/productController.js**: Contém a lógica de controle das operações de CRUD (Criar, Ler, Atualizar, Deletar).
- **/routes/productRoutes.js**: Define as rotas da API, como a criação de produtos, listagem e exclusão.

## 2. Funções CRUD

As funcionalidades principais implementadas no backend são:

- **Criar Produto**: Permite a criação de novos produtos.
  - **Listar Produtos**: Exibe todos os produtos cadastrados.
  - **Obter Produto por ID**: Retorna os detalhes de um produto específico.
  - **Atualizar Produto**: Atualiza as informações de um produto existente.
  - **Deletar Produto**: Exclui um produto do banco de dados.
- 

## Frontend

### 1. Estrutura do Frontend

O frontend é uma aplicação React Native que consome a API do backend.

- **App.js**: Configura a navegação entre as telas do aplicativo, como a tela inicial e a tela de detalhes do produto.
- **HomeScreen.js**: Tela inicial que lista todos os produtos cadastrados. Ela utiliza o componente **ProductCard** para exibir uma prévia de cada produto.
- **ProductDetailsScreen.js**: Tela para visualizar os detalhes de um produto, com a opção de editá-lo ou adicioná-lo.
- **api.js**: Arquivo que define as funções para realizar as requisições HTTP para o backend.

### 2. Componentes

- **ProductCard.js**: Exibe as informações do produto, como nome e imagem. Pode ser clicado para navegar até a tela de detalhes.
  - **ProductForm.js**: Usado para adicionar ou editar produtos, coletando dados de entrada do usuário e enviando para a API.
-

## Execução do Projeto

### 1. Configuração do Backend

1. Instale as dependências:

bash

```
npm install express mongoose body-parser cors
```

2. Inicie o servidor:

bash

```
node server.js
```

O servidor estará disponível em <http://localhost:5000/api/products>.

### 2. Configuração do Frontend

1. Instale as dependências do React Native com Expo:

bash

```
cd frontend  
npm install axios react-navigation react-navigation-stack  
react-native-screens react-native-safe-area-context  
react-native-gesture-handler react-native-reanimated
```

2. Inicie a aplicação no Expo:

bash

```
npx expo start
```