

# Documentação app de agendamento.

## 1. Introdução:

- Nome da aplicação: Lorem Ipsum
- Principais tecnologias utilizadas:
  - Frontend: Desenvolvido usando React Native, o frontend é responsável por proporcionar uma interface amigável e intuitiva para o usuário.

Os principais recursos incluem:

- Listagem de contatos: Mostra uma lista de todos os contatos salvos.
  - Formulário de contato: Permite adicionar ou editar informações de contatos.
  - Botões de ação: Permitem editar ou excluir contatos diretamente na lista.
  - As interações realizadas pelo usuário (como adicionar ou editar um contato) são enviadas ao backend por meio de requisições HTTP.
- Backend: Construído com Express.js, o backend gerencia as operações no servidor e interage com o banco de dados MongoDB.

Funcionalidades principais:

- Adicionar contato: Armazena novos contatos no banco de dados.
- Listar contatos: Recupera e retorna todos os contatos armazenados.
- Editar contato: Atualiza informações específicas de um contato.
- Excluir contato: Remove contatos com base no ID.

O backend utiliza Mongoose para modelar os dados no MongoDB, garantindo uma estrutura consistente.

## • Visão Geral do Funcionamento

O aplicativo é uma solução de gerenciamento de contatos que permite ao usuário adicionar, visualizar, editar e excluir contatos. Ele é dividido em duas partes principais: frontend (interface do usuário) e backend (servidor e banco de dados).

## • Arquitetura Geral

Frontend (React Native):

1. Tela principal -> Lista de contatos.
2. Tela de formulário -> Adição/edição de contatos.

Backend (Express.js):

1. Endpoints RESTful gerenciados pelo roteador.
2. Banco de dados MongoDB conectado com Mongoose.

Comunicação:

1. O frontend se comunica com o backend por meio de requisições HTTP (fetch).
2. O backend processa as requisições e responde com JSON.
3. Essa arquitetura modular garante uma separação clara entre a interface do usuário e a lógica do servidor, permitindo uma manutenção mais simples e expansões futuras.

## • Fluxo de Dados

1. Adicionar contato:

- O usuário preenche o formulário e clica em "Salvar".
- O frontend envia os dados para o endpoint POST /contato.
- O backend valida e salva os dados no banco de dados, retornando o novo contato.

2. Editar contato:

- O usuário clica no ícone de edição.
- O frontend carrega os dados do contato no formulário e envia as alterações para o endpoint PUT /contato/:id.
- O backend atualiza o registro no banco de dados e retorna o contato atualizado.

3. Excluir contato:

- O usuário clica no ícone de exclusão.
- O frontend envia uma requisição ao endpoint DELETE /contato/:id.
- O backend remove o registro correspondente no banco de dados.

#### 4. Listar contatos:

- Ao abrir o aplicativo, o frontend faz uma requisição ao endpoint GET /contato.
- O backend retorna todos os contatos, que são exibidos na interface.

## 2. Instalação

### Pré-requisitos

Antes de começar, certifique-se de ter os seguintes itens instalados no seu sistema:

- Node.js (versão 16 ou superior)
- npm ou yarn (gerenciador de pacotes do Node.js)
- Android Studio ou o aplicativo para celular 'Expo GO' (para emular o aplicativo ou testar no dispositivo físico)
- Git (opcional, para clonar o repositório)

## Configuração

- **Passos para clonar o repositório:**

1º Abra o terminal e clone o repositório github:

```
git clone https://github.com/luccaBrasii/agenda-contatos.git  
cd agenda-contatos
```

2º Após acessar a pasta 'agenda-contatos' ainda no terminal instale as dependências com:

```
npm install
```

- **Como executar o aplicativo:**

Será necessário abrir dois terminais, no primeiro executaremos o servidor, e no segundo a aplicação react (backend e frontend respectivamente).

1. Como no passo anterior já abrimos o terminal e estamos dentro da pasta **agendamento-contatos** basta executar no terminal o comando:

```
node server.js
```

Essa mensagem deverá aparecer no terminal:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Toledo\Desktop\lucca\R-native\lista-contatos> node server.js
(node:18740) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in a future version. (Use `node --trace-warnings ...` to show where the warning was created)
(node:18740) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in a future version.
Servidor rodando em http://localhost:3000
DB is Up!
```

2. No segundo terminal, acessamos novamente a pasta

**agendamento-contatos**

E executamos:

```
npm start
```

Esse comando irá executar o aplicativo React junto com a lib 'Expo', e essa mensagem deverá aparecer:

```
PS C:\Users\Toledo\Desktop\lucca\R-native\lista-contatos> npm start

> lista-contatos@1.0.0 start
> expo start

Starting project at C:\Users\Toledo\Desktop\lucca\R-native\lista-contatos
Starting Metro Bundler
```



- **Executar no Android Studio:** Caso tiver o 'Android Studio' instalado na máquina basta iniciar o dispositivo virtual e apertar a tecla 'A' no terminal que ele vai abrir automaticamente o aplicativo no dispositivo virtual,
- **Executar no dispositivo físico (Recomendado):** Caso tenha instalado o aplicativo de celular 'Expo GO' no dispositivo físico basta acessar o aplicativo e escanear o QR code que aparece no terminal (Lembrando que precisam estar conectados na mesma rede, o computador e o dispositivo móvel)

Caso tudo tenha ocorrido corretamente você deverá ver essa tela no seu dispositivo:

## Agendamento App

Adicionar contato

Nome: camila santos

Email: Caca.santtoss@gmail.com

Telefone: 62991838317



