

# Documentação do Projeto

## Visão Geral

Este projeto tem como objetivo realizar a gestão de clientes, isso inclui a realização de cadastro e manipulação dessas informações . Ele foi desenvolvido utilizando tecnologias como Java para o servidor backend, MySQL para o banco de dados e HTML, CSS e JavaScript para o frontend. O sistema permite cadastrar as informações de novos clientes tanto quanto a edição e exclusão desses dados.

## Tecnologias Utilizadas

- **Backend:** Java (Spring Boot e Java)
- **Banco de Dados:** MySQL
- **Frontend:** HTML, CSS e JavaScript

## Estrutura do Projeto

A estrutura do projeto está organizada da seguinte forma:

```
projeto-estagio/  
├── backend/  
│   ├── src/  
│   │   ├── main/  
│   │   │   ├── java/com/projeto/  
│   │   │   └── resources/  
│   │   └── test/  
│   ├── pom.xml (Caso utilize Maven)  
│   └── build.gradle (Caso utilize Gradle)  
├── frontend/  
│   ├── index.html  
│   ├── assets/  
│   │   ├── css/  
│   │   └── js/  
├── database/  
│   ├── schema.sql  
│   └── data.sql  
└── README.md
```

# Configuração do Ambiente

## 1. Configurar o Banco de Dados MySQL

1. Instale o MySQL e crie um banco de dados:

```
CREATE DATABASE nome_do_banco;
```

2. Configure as credenciais no arquivo `application.properties` ou `application.yml` no backend:

```
spring.datasource.url=jdbc:mysql://localhost:3306/nome_do_banco
spring.datasource.username=root
spring.datasource.password=senha
spring.jpa.hibernate.ddl-auto=update
```

**Obs: caminho para localizações dados sensíveis do MySQL:**

Extrair arquivo zip => projeto-estagio => Gestao => backend => config.env

## 2. Configurar o Backend

1. Navegue até a pasta do backend:

```
cd backend
```

2. Compile e execute o servidor:

```
mvn spring-boot:run # Caso utilize Maven
```

Ou

```
./gradlew bootRun # Caso utilize Gradle
```

## 3. Configurar o Frontend

1. Abra o arquivo `index.html` no navegador ou utilize um servidor local:

```
cd frontend
python -m http.server 8080
```

Em seguida, acesse <http://localhost:8080> no navegador.

## Funcionalidades do Sistema

- **Cadastro de Usuários:** Permite criar, editar e excluir usuários.
- **Gerenciamento de Dados:** Permite armazenar e recuperar informações.

## API Endpoints

Método	Rota	Descrição
GET	/api/usuarios	Lista todos os usuários
POST	/api/usuarios	Cria um novo usuário
PUT	/api/usuarios/{id}	Atualiza um usuário
DELETE	/api/usuarios/{id}	Remove um usuário

## Contato

Caso precise de suporte ou queira contribuir, entre em contato pelo email [lavihsf@gmail.com] ou abra uma issue no repositório do projeto.