**FATEC “ARTHUR DE AZEVEDO” – MOGI MIRIM**

**Documentação**

**Controle de Clientes**

**Alunos:**

**LUCAS MONTEJANO**

**LEANDRO DIONÍSIO**

**PEDRO AUGUSTO**

**LUCIANO GURJÃO**

### **1. Visão Geral do Projeto**

#### **1.1 Descrição**

O projeto é um sistema de cadastro de clientes desenvolvido utilizando Spring Boot, Thymeleaf, e MySQL. Ele permite realizar operações de CRUD (Create, Read, Update, Delete) em registros de clientes.

#### **1.2 Funcionalidades**

* Adicionar novos clientes
* Listar todos os clientes
* Atualizar informações dos clientes
* Excluir clientes

### **2. Arquitetura do Sistema**

#### **2.1 Tecnologias Utilizadas**

* **Spring Boot**: Framework para criar aplicações web robustas.
* **Thymeleaf**: Template engine para renderização de páginas HTML.
* **MySQL**: Banco de dados relacional.
* **Maven**: Gerenciador de dependências e construção do projeto.

### **3. Estrutura do Projeto**

#### **3.1 Arquivos Principais**

* **pom.xml**: Configurações do Maven e dependências do projeto.
* **ProjectMaromoApplication.java**: Classe principal que inicializa a aplicação.
* **Client.java**: Entidade que representa um cliente.
* **RequestClient.java**: DTO usado para transferência de dados dos clientes.
* **ClientRepository.java**: Interface do repositório JPA para operações com clientes.
* **ClientController.java**: Controlador REST que gerencia as requisições HTTP relacionadas aos clientes.
* **RequestsExceptionHandler.java**: Manipulador de exceções.
* **application.properties**: Configurações do banco de dados e outras propriedades da aplicação.
* **HTML Files**: Páginas HTML utilizadas na interface do usuário.

#### **3.2 Diagrama de Classes**

Diagrama

Descrição gerada automaticamente

### **4. Detalhamento dos Componentes**

#### **4.1 pom.xml**

O arquivo de configuração do Maven que especifica as dependências necessárias para o projeto.

<dependency>

<!-- Exemplo de uma dependência -->

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

#### **4.2 ProjectMaromoApplication.java**

Classe que inicia a aplicação Spring Boot.

@SpringBootApplication

public class ProjectMaromoApplication {

public static void main(String[] args) {

SpringApplication.run(ProjectMaromoApplication.class, args);

}

}

#### **4.3 Client.java**

Classe que define a entidade Cliente.

@Entity(name = "client")

public class Client {

@Id

@GeneratedValue(strategy = GenerationType.SEQUENCE)

private Long id;

// outros campos e métodos

}

#### **4.4 ClientController.java**

Classe que gerencia as operações de CRUD dos clientes via API REST.

@RestController

@RequestMapping("/clients")

public class ClientController {

@Autowired

private ClientRepository clientRepository;

@GetMapping

public ResponseEntity getAllClients() {

// implementação

}

@PostMapping

public ResponseEntity addClient(@RequestBody @Valid RequestClient data) {

// implementação

}

// outros métodos

}

### **5. Configuração do Ambiente**

#### **5.1 Pré-requisitos**

* Java 22
* Maven
* MySQL

#### **5.2 Configuração do Banco de Dados**

Atualize o arquivo application.properties com as configurações do seu banco de dados MySQL.

spring.datasource.url=jdbc:mysql://localhost:3306/mysql\_maromo

spring.datasource.username=root

spring.datasource.password=password

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.thymeleaf.cache=false

### **6. Executando o Projeto**

#### **6.1 Passos para Compilar e Executar**

1. Clone o repositório.
2. Navegue até o diretório do projeto.
3. Execute o comando mvn spring-boot:run.

### **7. Exemplos de Uso**

#### **7.1 Adicionar Cliente**

Exemplo de requisição HTTP para adicionar um novo cliente:

POST /clients

Content-Type: application/json

{

"name": "João Silva",

"email": "joao.silva@example.com",

"cpf": "12345678901",

"phone": "123456789",

"address": "Rua A, 123",

"city": "São Paulo"

}

### **8. Conclusão**

A documentação fornece uma visão abrangente do projeto de cadastro de clientes, detalhando a arquitetura, componentes, e instruções para configuração e execução. Para maiores informações ou dúvidas, consulte a documentação oficial do Spring Boot e Thymeleaf.