

# FIAP GRADUAÇÃO

# SISTEMAS DE INFORMAÇÃO

MICROSERVICE AND WEB ENGINEERING

Prof<sup>a</sup>. Aparecida Castello Rosa  
[profaparecida.rosa@fiap.com.br](mailto:profaparecida.rosa@fiap.com.br)

# I Agenda

- Hora de praticar

# Objetivos

- Desenvolver as funcionalidades CRUD no back-end
- O front-end (views), HomeController, fragment navbar, template.html já estão implementados.

# Projeto Loja



## Project

☐ Gradle - Groovy ☐ Gradle - Kotlin

☒ Maven

## Language

☒ Java ☐ Kotlin ☐ Groovy

## Spring Boot

☐ 3.3.0 (SNAPSHOT) ☐ 3.3.0 (M2) ☐ 3.2.4 (SNAPSHOT) ☒ 3.2.3

☐ 3.1.10 (SNAPSHOT) ☐ 3.1.9

## Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 21 ☒ 17

## Dependencies

ADD DEPENDENCIES... CTRL + B

### Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

### H2 Database

SQL

Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

### Spring Data JPA

SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

### Thymeleaf

TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

### Validation

I/O

Bean Validation with Hibernate validator.

### Spring Boot DevTools

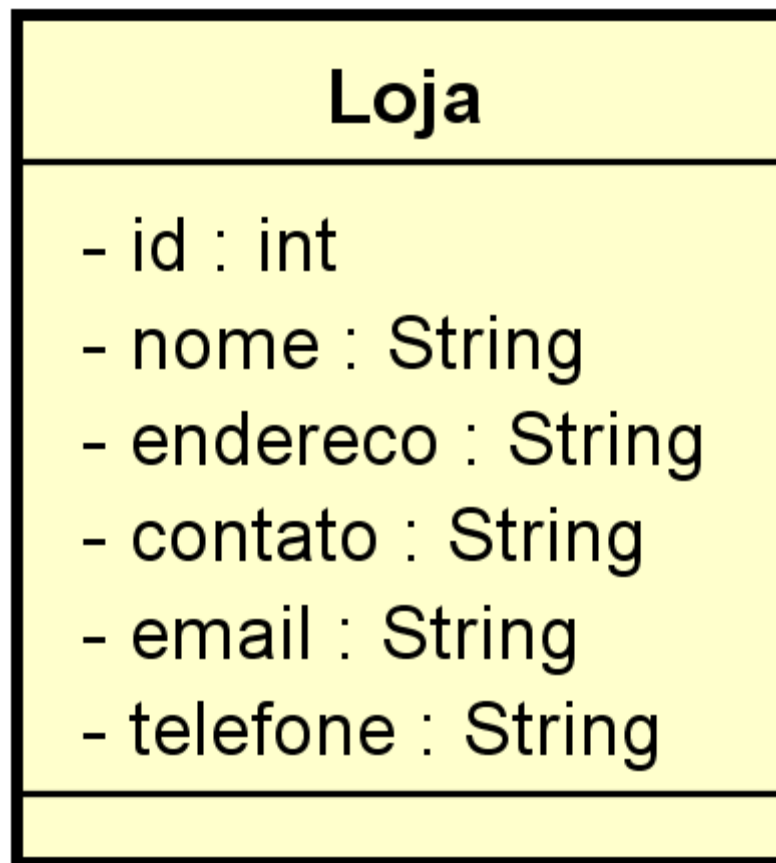
DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

# Mapeamento das requisições

Requisição - Roteamento	Descrição
http://localhost:8080/	Exibe a Home Page – views/index.html
http://localhost:8080/lojas/novo	Exibe a página para cadastro da loja- views/novo-loja.html
http://localhost:8080/lojas/salvar	Salva recurso no banco de dados
http://localhost:8080/lojas/listar	Exibe página com a listagem das lojas- views/listar-lojas.html
http://localhost:8080/lojas/editar/1	Exibe página para editar os dados da loja- views/editar-loja.html (GET)
http://localhost:8080/lojas/editar/1	Salva no banco de dados as alterações (POST)
http://localhost:8080/lojas/deletar/1	Exclui recurso do banco de dados

# Diagrama de classe – Entidade Loja



# Tutorial para consulta



**Model**

**Produto Entity**

```
15 // -- código omitido
16 // -- código omitido
17 @Entity
18 @Table(name = "tb_produto")
19 public class Produto {
20
21     @Id
22     @GeneratedValue(strategy = GenerationType.IDENTITY)
23     private Long id;
24
25     @NotBlank(message = "Campo requerido")
26     @Size(min = 3, message = "O nome deve ter no mínimo 3 caracteres")
27     private String nome;
28
29     @NotBlank(message = "Campo requerido")
30     @Column(columnDefinition = "TEXT") //para textos longos
31     private String descricao;
32
33     @NotNull(message = "Campo requerido")
34     @Positive(message = "O valor deve ser positivo")
35     private Double valor;
36 // -- código omitido
37 // Construtores, getters and setters, equals and hashCode, toString
```

**Repository**

**ProdutoRepository**

# ProdutoRepository

```
ProdutoRepository.java x
1 package br.com.fiap.produtomvc.repository;
2
3 import br.com.fiap.produtomvc.models.Produto;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface ProdutoRepository extends JpaRepository<Produto, Long> {
7 }
```

**Seed DB**

**import.sql**

# Arquivo import.sql

```
import.sql ×  
1 INSERT INTO tb_produto(nome, descricao, valor) VALUES('Mouse Microsoft', 'Mouse sem fio', 250.0);  
2 INSERT INTO tb_produto(nome, descricao, valor) VALUES('Smartphone Samsung Galaxy A54 5G', 'Samsung Galaxy A54 5G', 1799.0);  
3 INSERT INTO tb_produto(nome, descricao, valor) VALUES('Smart TV', 'Smart TV LG LED 65 polegadas', 3999);
```

**Controller**

**ProdutoController**

```
17 // -- código omitido
18 //URL - localhost:8080/produtos
19 @Controller
20 @RequestMapping("/produtos")
21 public class ProdutoController {
22     @Autowired
23     private ProdutoRepository repository;
24
25     //URL - localhost:8080/produtos/novo
26     @GetMapping("/novo")
27     @PostMapping("/novo")
28     public String adicionarProduto(Model model) {
29         model.addAttribute("produto", new Produto());
30         return "produto/novo-produto";
31     }
32
33     @PostMapping("/salvar")
34     @Transactional
35     public String insertProduto(@Valid Produto produto,
36                                BindingResult result,
37                                RedirectAttributes attributes) {
38         if(result.hasErrors()){
39             return "produto/novo-produto";
40         }
41         repository.save(produto);
42         attributes.addFlashAttribute("mensagem", "Produto salvo com sucesso");
43         return "redirect:/produtos/novo";
44     }
45 }
```



```
ProdutoController.java x
44
45 //URL - localhost:8080/produtos/listar
46 @GetMapping("/listar")
47 @Transactional(readOnly = true)
48 @
49 public String listarProdutos(Model model){
50     model.addAttribute("produtos", repository.findAll());
51     return "/produto/listar-produtos"; //View
52 }
53
54 //URL - localhost:8080/produtos/editar/1
55 @GetMapping("/editar/{id}")
56 @Transactional(readOnly = true)
57 @
58 public String editarProduto(@PathVariable ("id") Long id, Model model ){
59     Produto produto = repository.findById(id).orElseThrow(
60         () -> new IllegalArgumentException("Produto inválido - id: " + id)
61     );
62     model.addAttribute("produto", produto);
63     return "/produto/editar-produto";
64 }
```

64 //URL - localhost:8080/produtos/editar/1

65 @PostMapping("/editar/{id}")

66 @Transactional

67 @  
public String editarProduto(@PathVariable("id") Long id, @Valid Produto produto,  
BindingResult result){

68  
69 if(result.hasErrors()){

70 produto.setId(id);

71 return "/produto/editar-produto";

72 }

73 repository.save(produto);

74 return "redirect:/produtos/listar";

75 }

76  
77 //URL - localhost:8080/produtos/deletar/1

78 @GetMapping("/deletar/{id}")

79 @Transactional

80 public String deletarProduto(@PathVariable("id") Long id, Model model){

81 if(!repository.existsById(id)){

82 throw new IllegalArgumentException("Produto inválido - id: " + id);

83 }

84 try {

85 repository.deleteById(id);

86 } catch (Exception e){

87 throw new IllegalArgumentException("Produto inválido - id: " + id);

88 }

89 return "redirect:/produtos/listar";

90 }



**Copyright © 2024**  
**Prof<sup>a</sup>. Aparecida de Fátima Castello Rosa**

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).