

FIAP GRADUAÇÃO

SISTEMAS DE INFORMAÇÃO

MICROSERVICE AND WEB ENGINEERING

Prof^a. Aparecida Castello Rosa
profaparecida.rosa@fiap.com.br

I Agenda

- Continuar com o projeto produto-mvc
- CRUD – *Read* – Listar todos produtos
- View para listar todos os produtos

Objetivos

- Continuar com o projeto produto-mvc
- Criar *fragments* - navbar
- Ajustar Layout
- Apresentar mensagens de sucesso e erro na validação do formulário
- Criar funcionalidade de listagem de produtos
- Criar View para listar todos produtos

Layout

Continuação da aula anterior

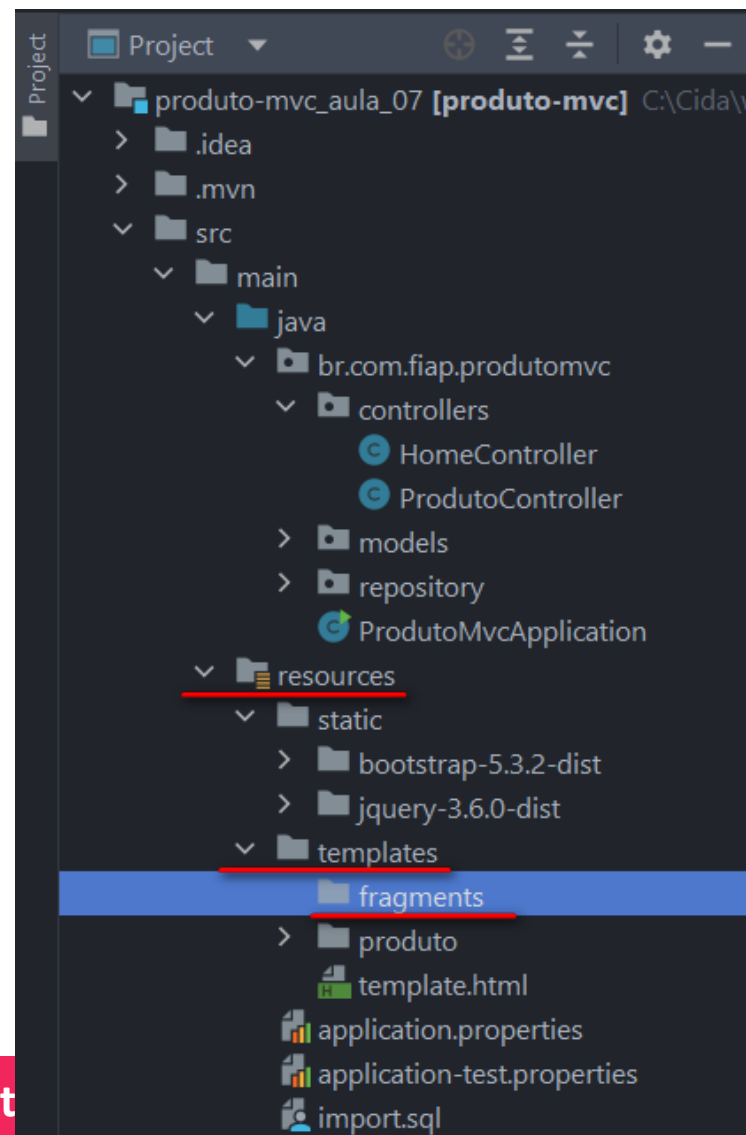
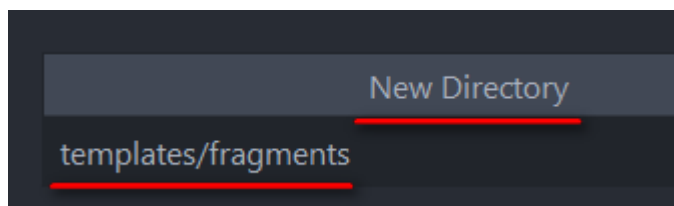
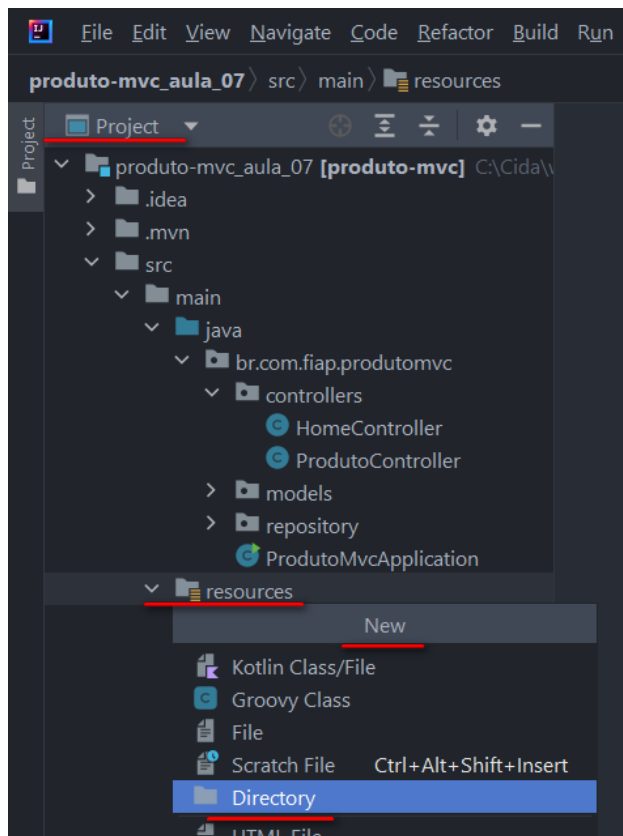
■ *Fragments*

- Os *fragments* (fragmentos) são uma forma de conseguirmos isolar um trecho de código dos nossos *templates* em um arquivo separado e então incluí-los em outros *templates* de acordo com a nossa necessidade.

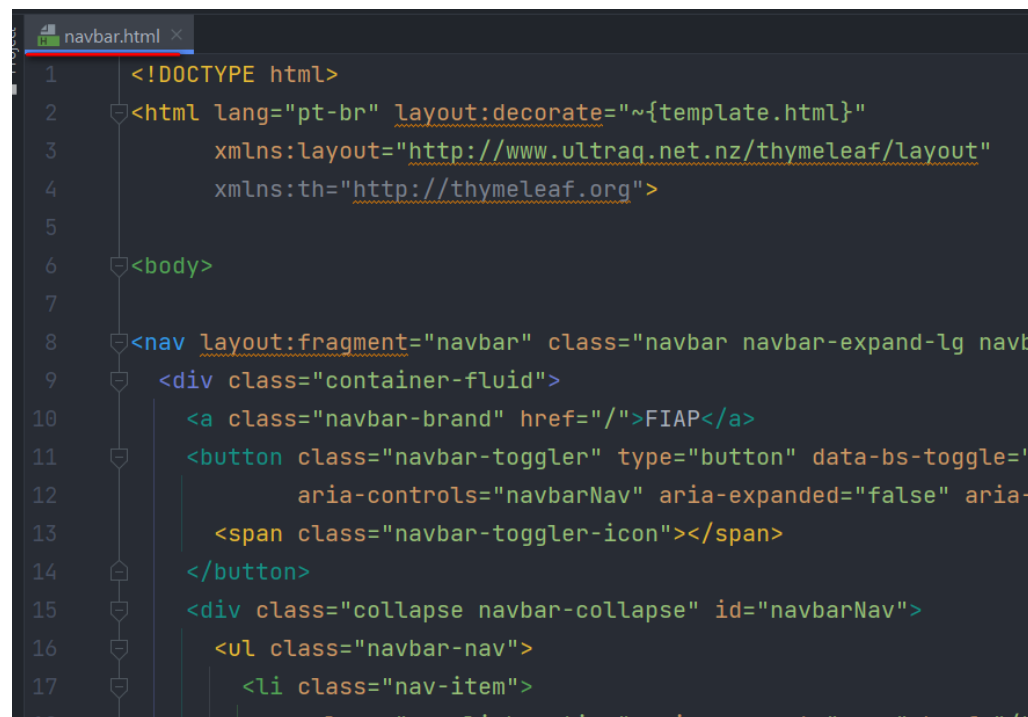
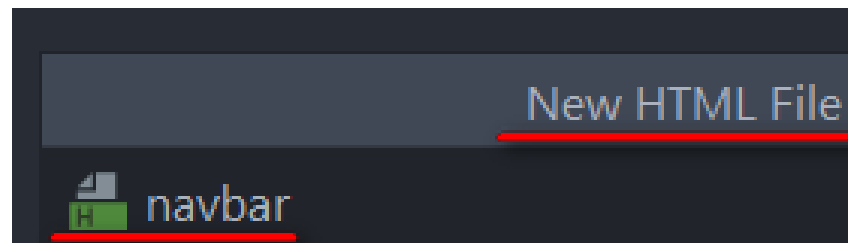
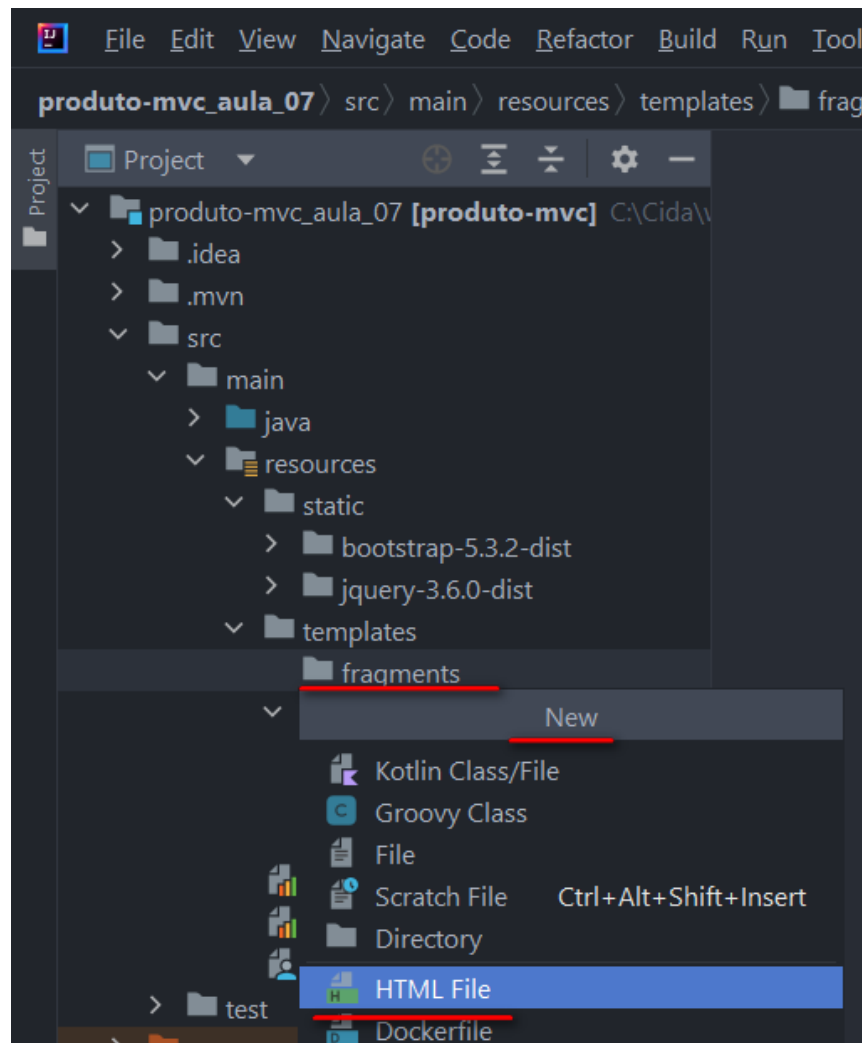
■ Criar um *fragment* para navbar

- Vamos criar um fragmento de código para incluir uma barra de navegação (navbar).
- Os fragmentos (códigos) devem ficar dentro da pasta *fragments*. Ela é uma subpasta de *templates*.

■ Criar um *fragment* para navbar




■ Criar um *fragment* para navbar



■ Código navbar.html

- Baixar o arquivo navbar anexado na aula.
- Copiar e colar o código, ou copiar e colar o arquivo no diretório *fragments*.
- Analisar o código.

Alterar template.html



The image shows a code editor window titled 'template.html'. The code is as follows:

```
23 </head>
24 <!-- código omitido-->
25 <body>
26
27 <!-- replace do fragment da navbar-->
28 <div layout:replace="~/fragments/navbar :: navbar"></div>
29
30 <!-- conteúdo dinâmico - cada página tem o seu conteúdo-->
31 <div layout:fragment="conteudo"></div>
32
33 </body>
34 </html>
```

Annotations in the image:

- A red box highlights the line `<div layout:replace="~/fragments/navbar :: navbar"></div>`.
- A red arrow points from the text 'path do arquivo navbar' to the `~/fragments/navbar` part of the code.
- A red arrow points from the text 'referência do fragment no arquivo navbar' to the `:: navbar` part of the code.

Entendendo navbar.html



The image shows a code editor with two tabs: 'template.html' and 'navbar.html'. The 'navbar.html' tab is active, displaying the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="pt-br" layout:decorate="~{template.html}"
3   xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
4   xmlns:th="http://thymeleaf.org">
5
6 <body>
7
8 <nav layout:fragment="navbar" class="navbar navbar-expand-lg navbar-light bg-light">
9
10 <!-- código omitido-->
```


Annotations in the image include:

- A red arrow pointing from the `layout:decorate="~{template.html}"` attribute in line 2 to the `layout:fragment="navbar"` attribute in line 8.
- A red arrow pointing from the `layout:fragment="navbar"` attribute in line 8 to a text box.
- A text box with a red border containing the text: "nome para ser referenciado no template.html".

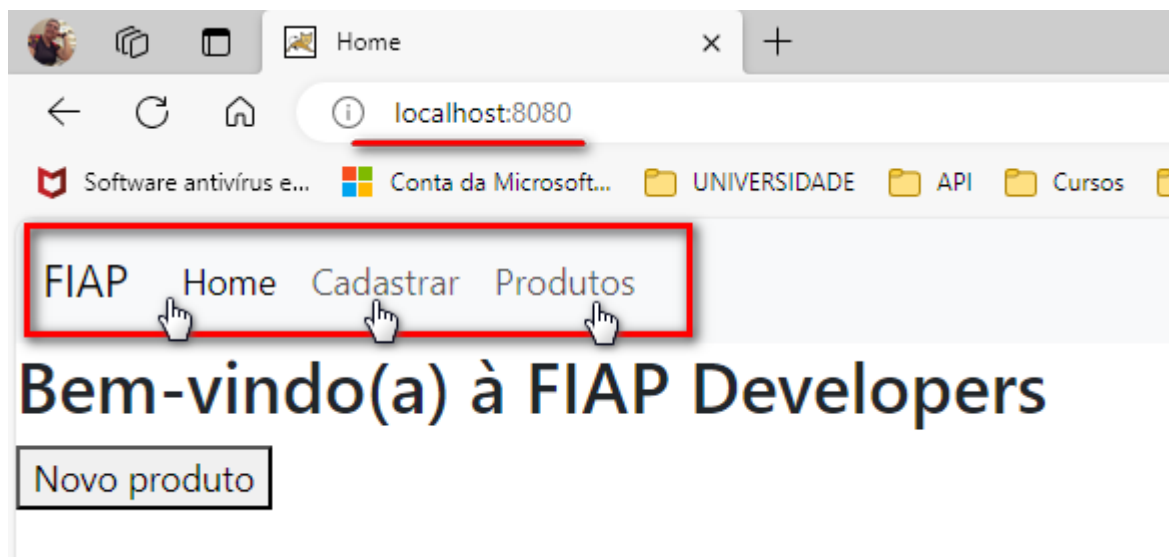
Template e navbar

```
template.html x
23 </head>
24 <!-- código omitido-->
25 <body>
26
27 <!-- replace do fragment da navbar-->
28 <div layout:replace="~/fragments/navbar :: navbar"></div>
29
```

```
template.html x navbar.html x
4 xmlns:th="http://thymeleaf.org">
5
6 <body>
7
8 <nav layout:fragment="navbar" class="navbar navbar-expand-lg na
9
```



Testar a aplicação



Testar a aplicação

```
Run: ProdutoMvcApplication x
Hibernate:
    insert
    into
        tb_produto
        (descricao, nome, valor, id)
    values
        (?, ?, ?, default)
```

localhost:8080/h2-console/login.do?jsessionid=ba237137d74df481dafd297f7b12

Universidade API Imagens e Sons Gr... Ferramentas Facebook Google & YT

Auto commit Max rows: 1000 Auto complete Off Auto select

jdbc:h2:mem:testdb

- TB_PRODUTO
- INFORMATION_SCHEMA
- Users
- H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM TB_PRODUTO|

SELECT * FROM TB_PRODUTO;

VALOR	ID	DESCRICAO	NOME
250.0	1	Mouse sem fio	Mouse Microsoft
1799.0	2	Samsung Galaxy A54 5G	Smartphone Samsung Galaxy A54 5G
3999.0	3	Smart TV LG LED 65 polegadas	Smart TV
250.5	4	Microfone de mesa	Microfone

(4 rows, 2 ms)

Alterar index.html

```
index.html x
11 <!-- código omitido -->
12 <head>
13 <!-- declarada no template
14 <meta charset="UTF-8" -->
15 <title>Home</title>
16 </head>
17 <body>
18 <div layout:fragment="conteudo">
19 <div class="container">
20 <div class="row">
21 <div class="col">
22 <h2 th:text="${msg}"></h2>
23 </div>
24 </div>
25 </div>
26 </div>
27 </body>
28 </html>
```


Alterar novo-produto.html

- Baixar o arquivo anexado na aula, copiar o código e colar.

```
10 <body>
11   <div layout:fragment="conteudo">
12     <div class="container">
13       <div class="row">
14         <div class="col">
15           <h2>Cadastro de Produtos</h2>
16           <!-- verifica se a mensagem não é nula
17           <div class="alert alert-success alert-d
18             <button type="button" class="btn-cl
19             <!-- se existe a mensagem, vai mostr
20             <span th:text="${mensagem}"></span>
21           </div>
22           <!-- th:action - vai encaminhar para o
23           <!-- th:object faz referencia ao obj -
24           <form action="#" th:action="@{/produtos
```

Comentários sobre novo-produto.html

```
<h2>Cadastro de Produtos</h2>
<!-- verifica se a mensagem não é nula, se existe na requisição (chave,valor) -->
<div class="alert alert-success alert-dismissible fade show" th:if="${mensagem}">
  <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
  <!-- se existe a mensagem, vai mostrar aqui -->
  <span th:text="${mensagem}"></span>
</div>
```

Comentários sobre novo-produto.html

```
<!-- th:action - vai encaminhar para o mapeamento /produto/salvar -->
<!-- th:object faz referencia ao obj - th:object="${produto} - é o objeto enviado pelo controller - chave -->
<form action="#" th:action="@{/produtos/salvar}" th:object="${produto}" method="post">
  <div class="row mb-3">
    <label for="nome" class="col-sm-1 col-form-label">Nome:</label>
    <div class="col-sm-5">
      <!-- th:field associa input com cada campo com o atributo do obj-->
      <input id="nome" class="form-control" type="text" th:field="*{nome}" />
    </div>
    <div class="col-sm-5">
      <span class="mensagem" th:if="${#fields.hasErrors('nome')}" th:errors="*{nome}"></span>
      <!-- th:if="${#fields.hasErrors('nome')} - associado com o controller no BindingResult result -->
      <!-- th:errors="*{nome}" mostra o erro-->
    </div>
  </div>
</div>
```

Exercício

Implementar as funcionalidades do método
`insertProduto`

■ *Annotation* @Valid

- A *annotation* @Valid vem da especificação Bean Validation e é utilizada por diversos frameworks para indicar o disparo do processo de validação.

■ **BindingResult**

- Para conseguirmos verificar se no formulário teve erro ou não, utilizamos o objeto do tipo **BindingResult**. Essa classe possui vários métodos, tanto para verificar se existem erros de validação, quanto para acrescentar novos erros, caso seja preciso efetuar alguma lógica de validação diretamente no método do controller.
- Sempre que algum erro acontece, o Spring MVC disponibiliza um objeto do tipo **BindingResult**, contendo todas as informações necessárias. Esse objeto é o mesmo que usamos no método do controller para verificar se foram encontrados erros de validação.

■ *Redirect Messages*

- ***RedirectAttributes*** é utilizado para passar objetos quando usamos redirecionamento.
- ***FlashAttribute*** é uma maneira de armazenar atributos a serem utilizados em outro método do controller durante o redirecionamento.

■ Para saber mais

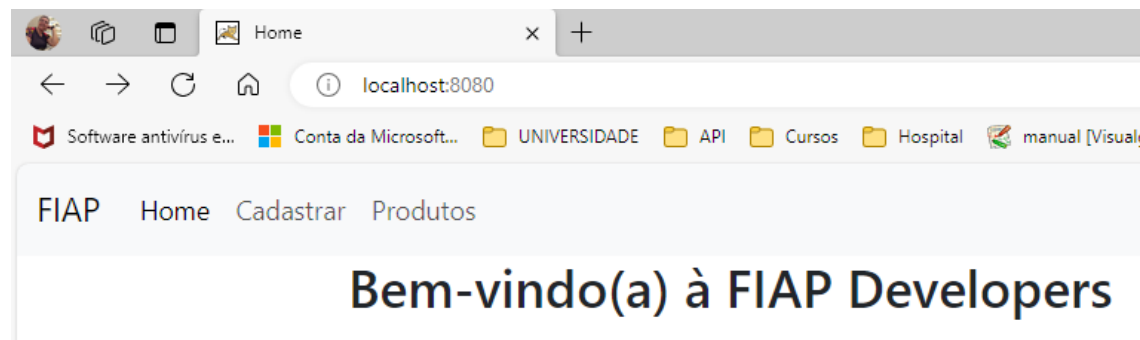
<https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/validation/BindingResult.html>

<https://www.baeldung.com/spring-boot-bean-validation>

Alterar método insertProduto - class ProdutoController

```
ProdutoController.java x
34 // código omitido
35 // receber dados do form da View novo-produto.html
36 //URL - localhost:8080/produtos/salvar
37
38 // @Valid - especificação Bean Validation - dispara o processo de validação
39 // class BindingResult - Para verificar se o formulário teve erro ou não
40 // RedirectAttributes - Redireciona objeto
41 @PostMapping("/salvar")
42 @
43 public String insertProduto(@Valid Produto produto,
44                             BindingResult result,
45                             RedirectAttributes attributes){
46     if(result.hasErrors()){
47         return "/produto/novo-produto";
48     }
49     repository.save(produto);
50     attributes.addFlashAttribute("mensagem", "Produto salvo com sucesso");
51     //redireciona para localhost:8080/produtos/novo
52     return "redirect:/produtos/novo";
53 }
```

Testar a aplicação

A screenshot of a web browser window showing the 'Cadastro de Produtos' (Product Registration) form. The browser's address bar displays 'localhost:8080/produtos/salvar'. The form includes three input fields: 'Nome:', 'Descrição:', and 'Valor:'. Each field is accompanied by a red error message: 'Campo requerido' (Required field) and 'O nome deve ter no mínimo 3 caracteres' (The name must have at least 3 characters) for the 'Nome' field, and 'Campo requerido' for the 'Descrição' and 'Valor' fields. A blue 'Salvar' (Save) button is located at the bottom of the form.

Testar a aplicação

Cadastro de Produto

localhost:8080/produtos/salvar

FIAP Home Cadastrar Produtos

Cadastro de Produtos

Nome: O nome deve ter no mínimo 3 caracteres

Descrição: Campo requerido

Valor: Campo requerido

Salvar

Cadastro de Produto

localhost:8080/produtos/salvar

FIAP Home Cadastrar Produtos

Cadastro de Produtos

Nome:

Descrição:

Valor: Campo requerido

Salvar

Testar a aplicação

Cadastro de Produto

localhost:8080/produtos/salvar

FIAP Home Cadastrar Produtos

Cadastro de Produtos

Nome:

Descrição:

Valor: Campo requerido

Cadastro de Produto

localhost:8080/produtos/novo;jsessionid=D4C25F69D3D45A6F6512CF071A338751

FIAP Home Cadastrar Produtos

Cadastro de Produtos

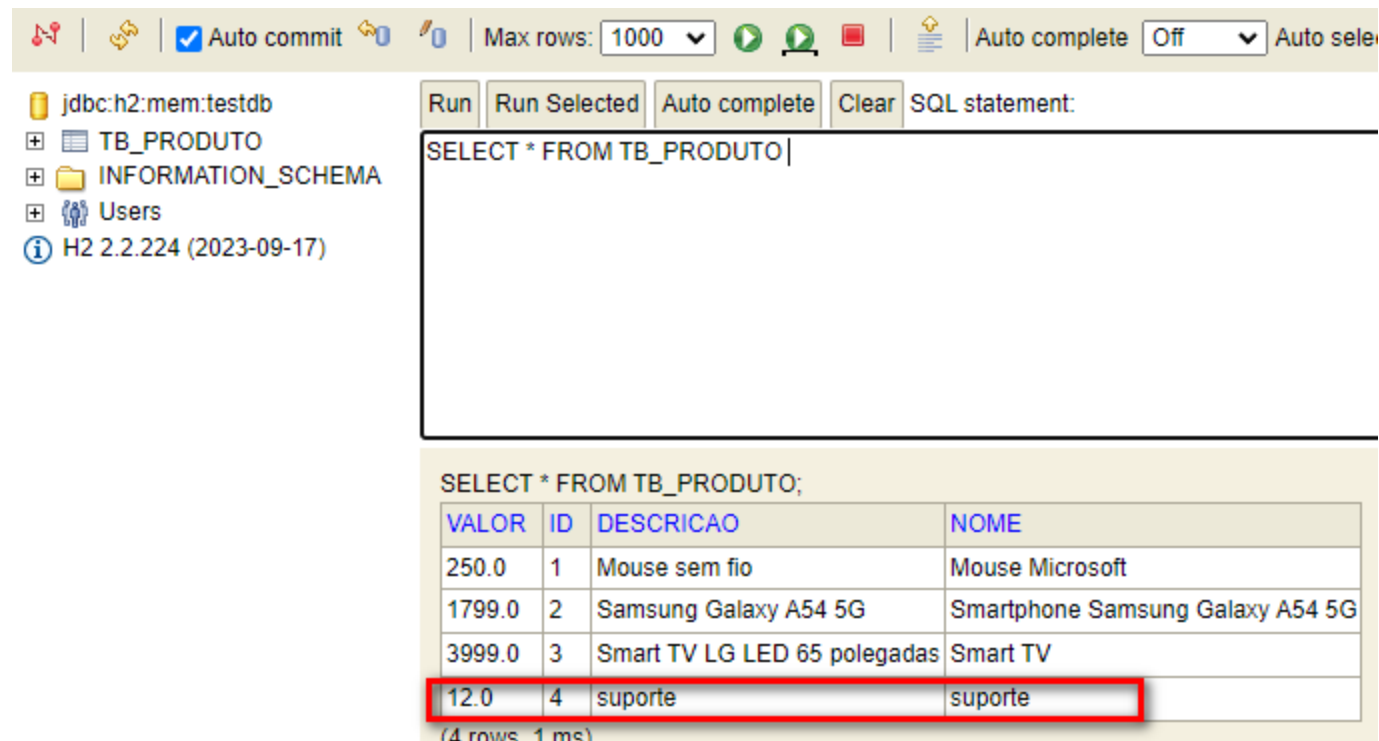
Produto salvo com sucesso

Nome:

Descrição:

Valor:

Testar a aplicação



The screenshot shows a SQL client interface with a toolbar at the top containing icons for undo, redo, auto-commit, and other functions. Below the toolbar, on the left, is a tree view showing the database structure: jdbc:h2:mem:testdb, TB_PRODUTO, INFORMATION_SCHEMA, Users, and H2 2.2.224 (2023-09-17). The main area on the right has a 'SQL statement:' input field with the text 'SELECT * FROM TB_PRODUTO;'. Below this, the query results are displayed in a table format. The table has four columns: VALOR, ID, DESCRICAO, and NOME. The data rows are: (250.0, 1, Mouse sem fio, Mouse Microsoft), (1799.0, 2, Samsung Galaxy A54 5G, Smartphone Samsung Galaxy A54 5G), (3999.0, 3, Smart TV LG LED 65 polegadas, Smart TV), and (12.0, 4, suporte, suporte). The last row is highlighted with a red border. At the bottom of the table, it says '(4 rows, 1 ms)'.

VALOR	ID	DESCRICAO	NOME
250.0	1	Mouse sem fio	Mouse Microsoft
1799.0	2	Samsung Galaxy A54 5G	Smartphone Samsung Galaxy A54 5G
3999.0	3	Smart TV LG LED 65 polegadas	Smart TV
12.0	4	suporte	suporte

(4 rows, 1 ms)

CRUD - *Read*

Ler/consultar registros no DB
`findAll()`

Exercício

Implementar o método `listarProdutos`

■ Inserir método listarProdutos - class ProdutoController^{FIAP}

- O método findAll() retorna uma lista de produtos que será enviada para a View.

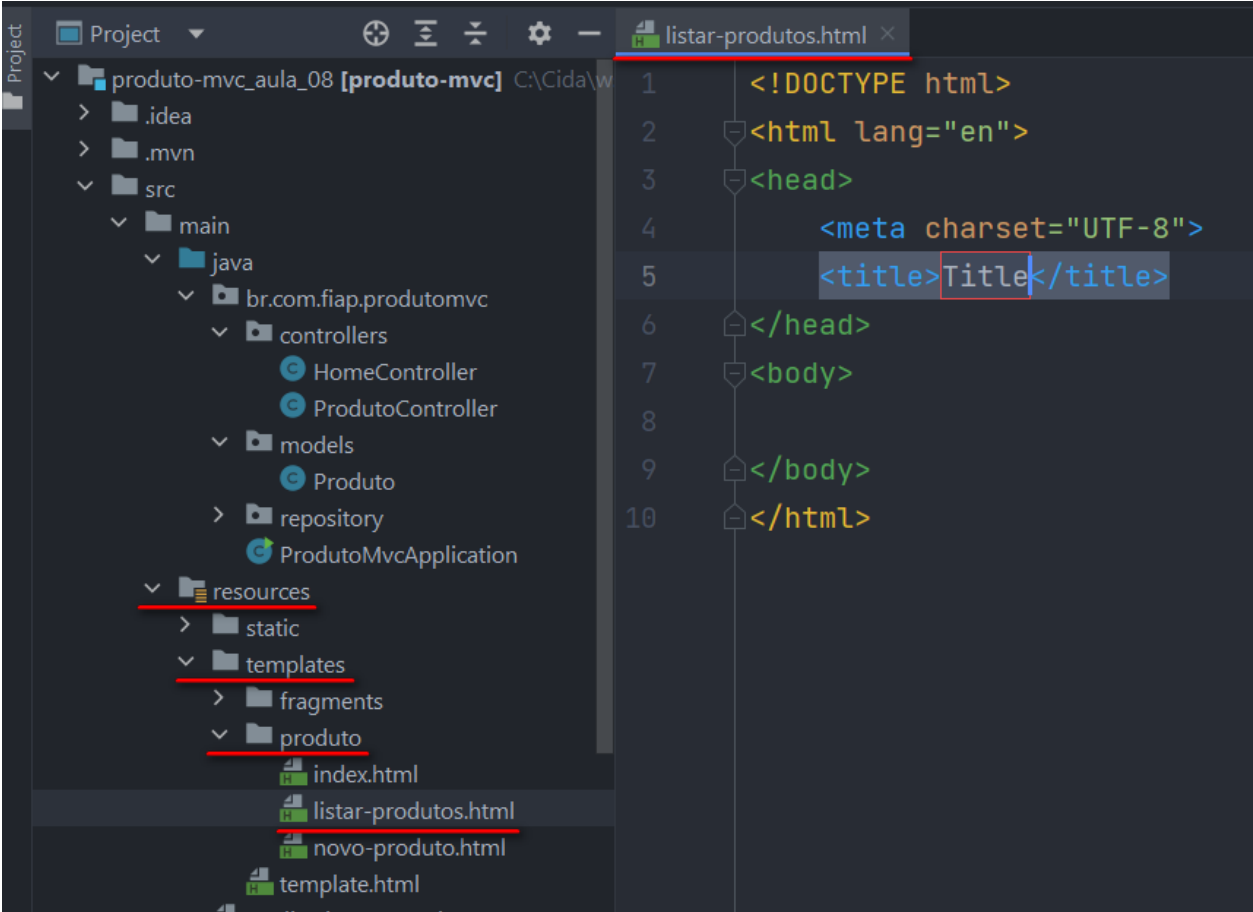
```
ProdutoController.java x
54 //código omitido
55
56 //URL - localhost:8080/produtos/listar
57 @GetMapping("/listar")
58 @ public String listarProdutos(Model model){
59     model.addAttribute("produtos", repository.findAll());
60     return "/produto/listar-produtos"; //View
61 }
```


Exercício

Implementar a View listar-produtos

I Criar a View listar-produtos

- Baixar o arquivo **listar-produtos.html** anexado a aula para complementar.
- Copiar e colar o código ou copiar e colar o arquivo.

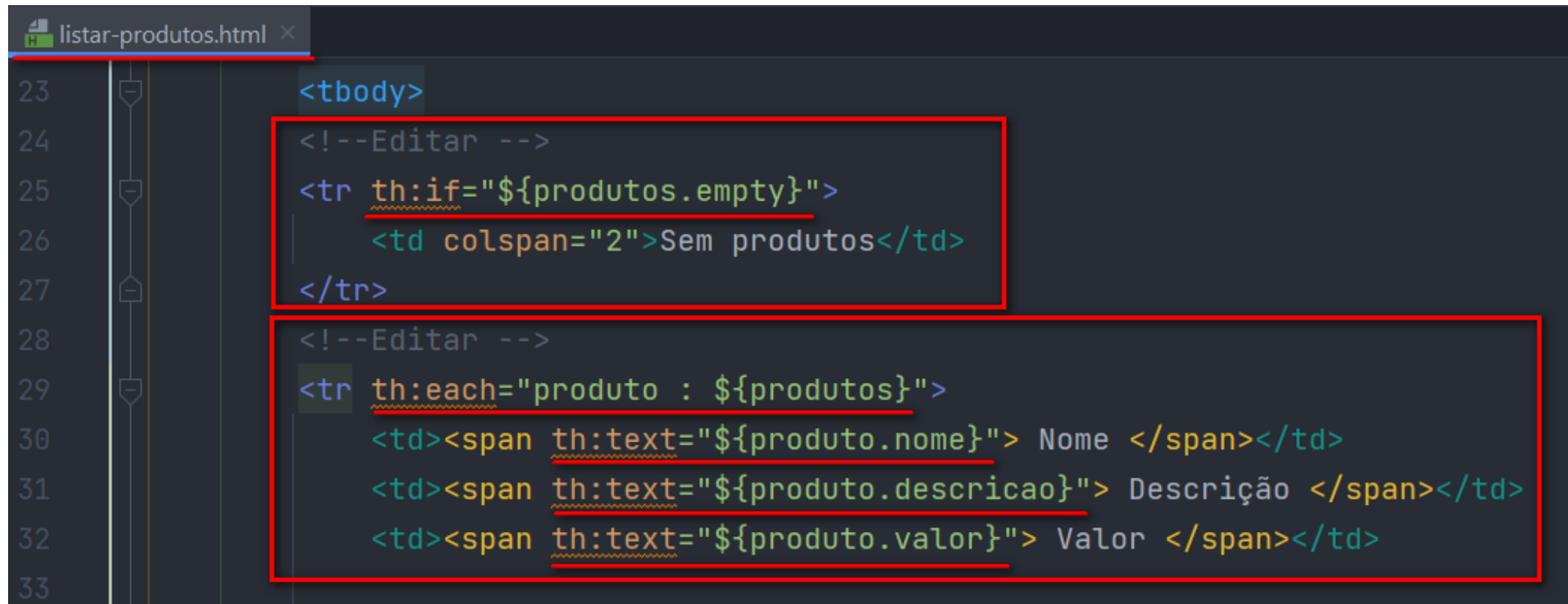


The screenshot shows an IDE with a project named 'produto-mvc_aula_08 [produto-mvc]'. The project structure on the left includes folders for '.idea', '.mvn', 'src', 'main', 'java', 'br.com.fiap.produtoMvc', 'controllers' (with HomeController and ProdutoController), 'models' (with Produto), 'repository' (with ProdutoMvcApplication), 'resources', 'static', 'templates', 'fragments', and 'produto'. The 'produto' folder contains 'index.html', 'listar-produtos.html', 'novo-produto.html', and 'template.html'. The 'listar-produtos.html' file is selected and its content is displayed in the editor on the right. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6 </head>
7 <body>
8
9 </body>
10 </html>
```

■ Editar a View listar-produtos

- Editar os trechos de código



```
listar-produtos.html x
23 <tbody>
24 <!--Editar -->
25 <tr th:if="{produtos.empty}">
26 <td colspan="2">Sem produtos</td>
27 </tr>
28 <!--Editar -->
29 <tr th:each="produto : {produtos}">
30 <td><span th:text="{produto.nome}"> Nome </span></td>
31 <td><span th:text="{produto.descricao}"> Descrição </span></td>
32 <td><span th:text="{produto.valor}"> Valor </span></td>
33
```

Entendendo a passagem da informação

listar-produtos.html

```
<tr th:if="${produtos.empty}">
    <td colspan="2">Sem produtos</td>
</tr>

<tr th:each="produto : ${produtos}" >
    <td><span th:text="${produto.nome}"> Nome </span></td>
    <td><span th:text="${produto.descricao}"> Descrição </span></td>
    <td><span th:text="${produto.valor}"> Valor </span></td>
    <td>
```

class ProdutoController

```
@GetMapping("/listar")
public String listarProdutos(Model model){
    model.addAttribute("produtos", repository.findAll());
    return "produto/listar-produtos"; //View
}
```

Atualizar endpoint na navbar

```
navbar.html x
19      </li>
20      <li class="nav-item">
21      |   <a class="nav-link" href="/produtos/novo">Cadastrar</a>
22      |
23      |   <li class="nav-item">
24      |   |   <a class="nav-link" href="/produtos/listar">Produtos</a>
25      |   |
26      |   </li>
27      </ul>
28  </div>
```

Testar aplicação



FIAP Home Cadastrar **Produtos**

Lista de Produtos

Nome	Descrição	Valor	Ações	
Mouse Microsoft	Mouse sem fio	250.0	Excluir	Editar
Smartphone Samsung Galaxy A54 5G	Samsung Galaxy A54 5G	1799.0	Excluir	Editar
Smart TV	Smart TV LG LED 65 polegadas	3999.0	Excluir	Editar



Copyright © 2024
Prof^a. 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).