



Estácio

CAMPUS: Polo Jacareí - SP – Centro

CURSO: Desenvolvimento FullStack

DISCIPLINA: Vamos integrar sistemas

TURMA: 2025.1

SEMESTRE LETIVO: Primeiro Semestre (2025)

ALUNO: Maurício Pereira Campos

MATRÍCULA: 202403843447

Título da prática:

Implementação de sistema cadastral com interface Web, baseado nas tecnologias de Servlets, JPA e JEE..

Objetivos da prática:

- 1.Implementar persistência com base em JPA.
- 2.Implementar regras de negócio na plataforma JEE, através de EJBs.
- 3.Implementar sistema cadastral Web com base em Servlets e JSPs.
- 4.Utilizar a biblioteca Bootstrap para melhoria do design.

1º Procedimento | Camadas de Persistência e Controle

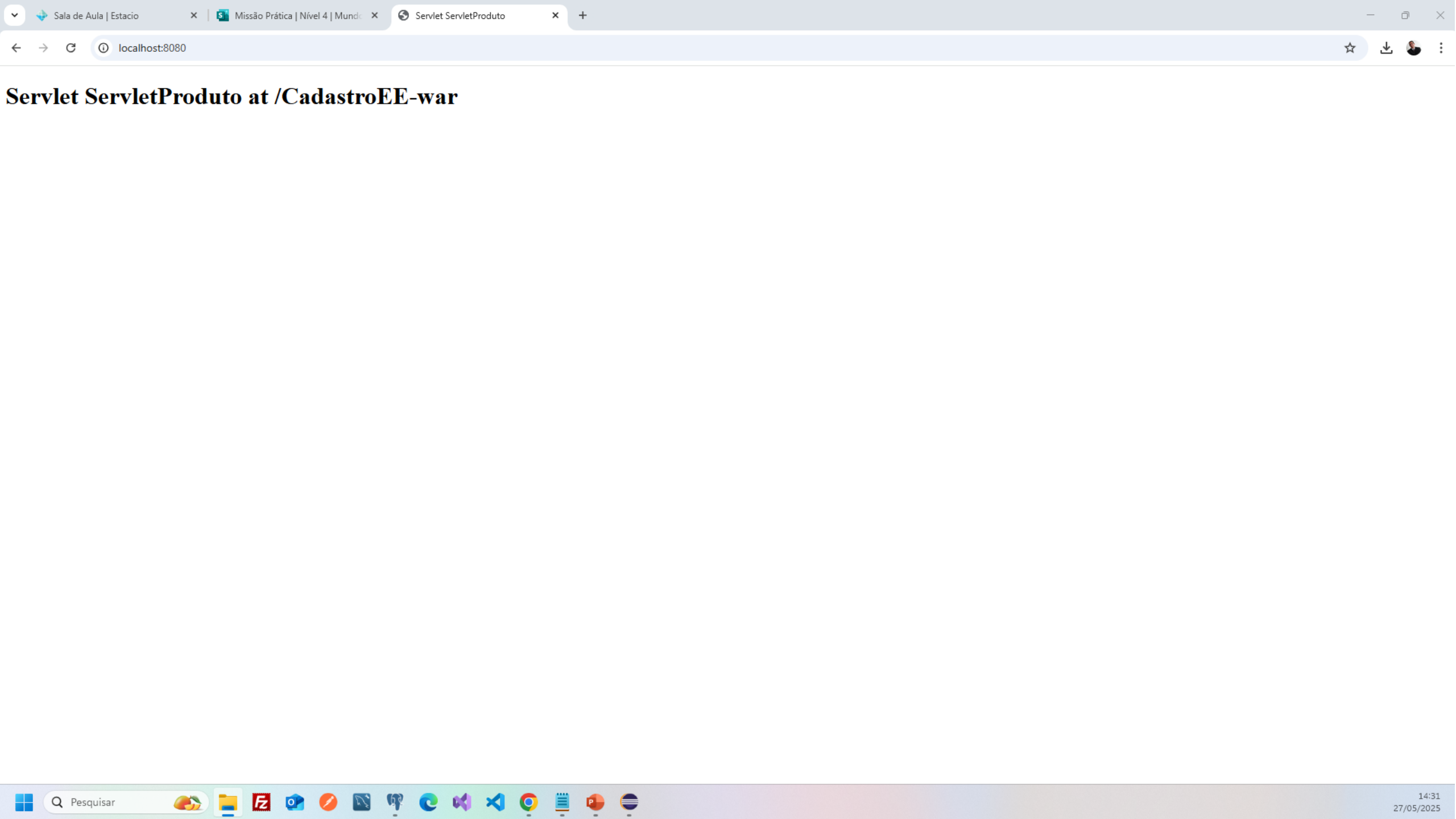
Códigos usados neste roteiro

The screenshot shows the Eclipse IDE interface with the following components:

- Package Explorer:** Displays the project structure for 'CadastroEE'. It includes the 'src/main/java' package containing 'com.estacio.procedimento01', which has 'Procedimento01Application.java' and 'com.estacio.procedimento01.controller'. It also shows 'src/main/resources' with 'templates', 'publica-index.html', and 'application.properties'. Other folders include 'src/test/java', 'JRE System Library [JavaSE-11]', 'Maven Dependencies', and 'src' with 'target', 'mvnw', 'mvnw.cmd', and 'pom.xml'.
- Main Editor:** Shows the code for 'Procedimento01Application.java'. The code is as follows:

```
1 package com.estacio.procedimento01;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class Procedimento01Application {
8
9     public static void main(String[] args) {
10         SpringApplication.run(Procedimento01Application.class, args);
11     }
12
13 }
14
```
- Task List:** Shows a 'Connect Mylyn' task and an 'Outline' view for 'com.estacio.procedimento01' with a method 'main(String[]) : void'.
- Console:** Displays the output of the application running. The logs show that the LiveReload server is running on port 35729, Tomcat is started on port 8080, and the application is initialized successfully in 1 ms.

The bottom status bar indicates the file is 'Writable', 'Smart Insert' is enabled, and the time is 7:41:221.



Análise e conclusão:

1.Como é organizado um projeto corporativo no NetBeans?

Um projeto corporativo no NetBeans é geralmente organizado em **módulos separados** para facilitar o desenvolvimento em equipe e a reutilização de código.

2.Qual o papel das tecnologias JPA e EJB na construção de um aplicativo para a plataforma Web no ambiente Java?

Juntas, JPA e EJB permitem criar aplicações web robustas, com separação clara entre lógica de negócio e acesso a dados.

3.Como o NetBeans viabiliza a melhoria de produtividade ao lidar com as tecnologias JPA e EJB?

Esses recursos reduzem o trabalho manual e aceleram o desenvolvimento de aplicações corporativas.

Análise e conclusão:

4.O que são Servlets, e como o NetBeans oferece suporte à construção desse tipo de componentes em um projeto Web?

Servlets são componentes Java que processam requisições HTTP em aplicações web, gerando respostas dinâmicas (como HTML).

5.Como é feita a comunicação entre os Servlets e os Session Beans do pool de EJBs?

A comunicação entre **Servlets** e **Session Beans (EJBs)** é feita por **injeção de dependência** ou **lookup JNDI**, permitindo que o Servlet chame métodos do EJB.

2º Procedimento | Interface Cadastral com Servlet e JSPs

Códigos usados neste roteiro

The screenshot displays an IDE with the following components:

- Package Explorer:** Shows the project structure for 'CadastroEE'. It includes packages like 'com.estacio.procedimento01' and 'com.estacio.procedimento01.controller'. The 'src/main/resources' folder contains templates like 'produto-dados.html', 'produto-lista.html', and 'publica-index.html'.
- Editor:** Displays the code for 'ProdutoController.java'. The code includes imports for 'com.estacio.procedimento01.dao.ProdutoDAO', 'com.estacio.procedimento01.model.Produto', 'org.springframework.stereotype.Controller', 'org.springframework.ui.Model', and 'org.springframework.web.bind.annotation.*'. It defines a 'ProdutoController' class with three methods: 'listarProdutos', 'formIncluir', and 'formAlterar'.
- Console:** Shows the output of the application, indicating that the 'execute' method of 'PeStatement' was called.

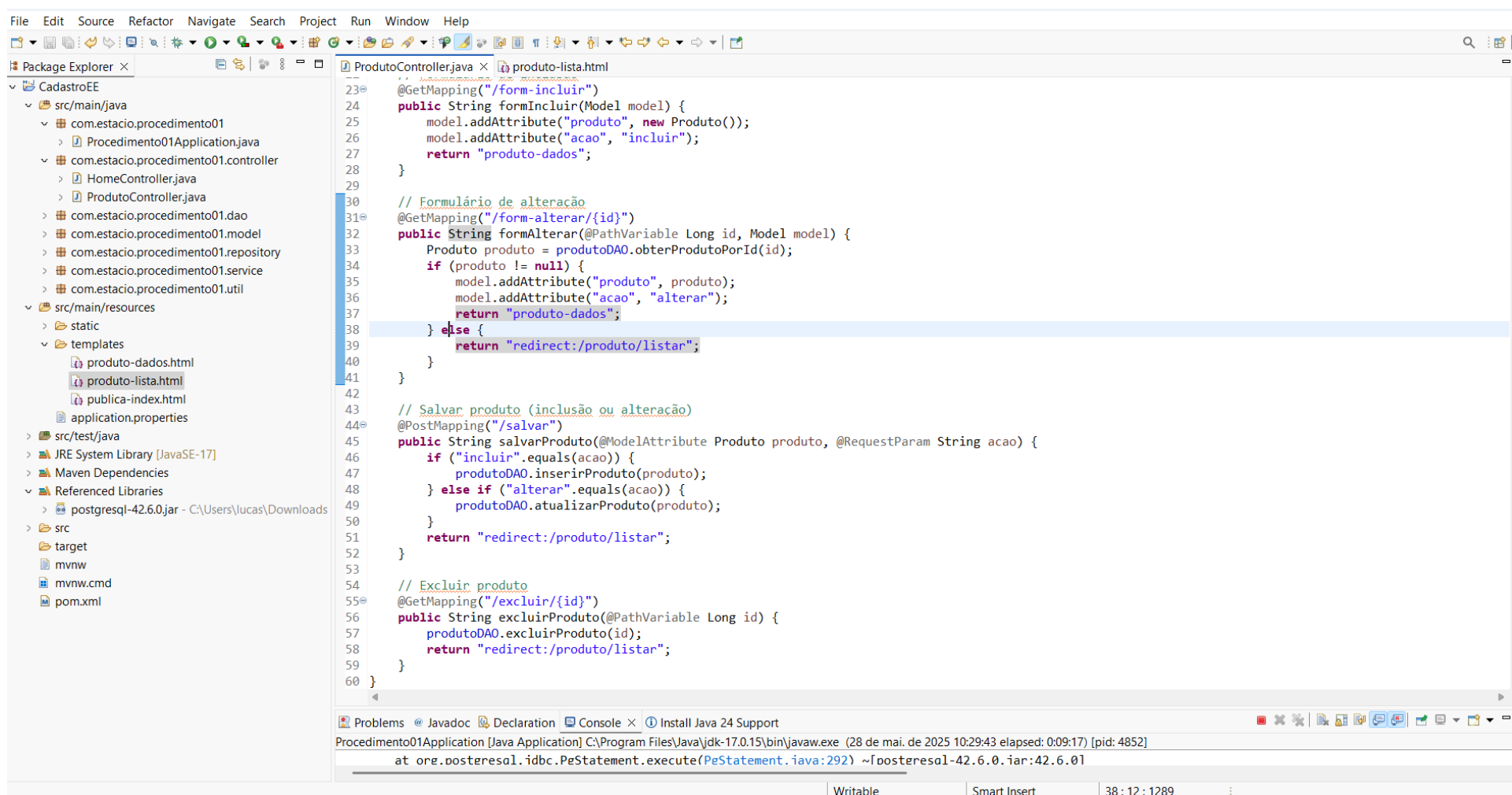
```
1 package com.estacio.procedimento01.controller;
2
3 import com.estacio.procedimento01.dao.ProdutoDAO;
4 import com.estacio.procedimento01.model.Produto;
5 import org.springframework.stereotype.Controller;
6 import org.springframework.ui.Model;
7 import org.springframework.web.bind.annotation.*;
8
9 @Controller
10 @RequestMapping("/produto")
11 public class ProdutoController {
12
13     private final ProdutoDAO produtoDAO = new ProdutoDAO();
14
15     // Listar produtos
16     @GetMapping("/listar")
17     public String listarProdutos(Model model) {
18         model.addAttribute("produtos", produtoDAO.listarProdutos());
19         return "produto-lista";
20     }
21
22     // Formulário de inclusão
23     @GetMapping("/form-incluir")
24     public String formIncluir(Model model) {
25         model.addAttribute("produto", new Produto());
26         model.addAttribute("acao", "incluir");
27         return "produto-dados";
28     }
29
30     // Formulário de alteração
31     @GetMapping("/form-alterar/{id}")
32     public String formAlterar(@PathVariable Long id, Model model) {
33         Produto produto = produtoDAO.obterProdutoPorId(id);
34         if (produto != null) {
35             model.addAttribute("produto", produto);
36             model.addAttribute("acao", "alterar");
37             return "produto-dados";
38         } else {
```

Problems @ Javadoc Declaration Console x Install Java 24 Support
Procedimento01Application [Java Application] C:\Program Files\Java\jdk-17.0.15\bin\javaw.exe (28 de mai. de 2025 10:29:43 elapsed: 0:08:09) [pid: 4852]
at org.postgresql.jdbc.PsStatement.execute(PsStatement.java:292) ~[postgresql-42.6.0.jar:42.6.0]

Writable Smart Insert 60 : 2 : 1993

2º Procedimento | Interface Cadastral com Servlet e JSPs

Códigos usados neste roteiro

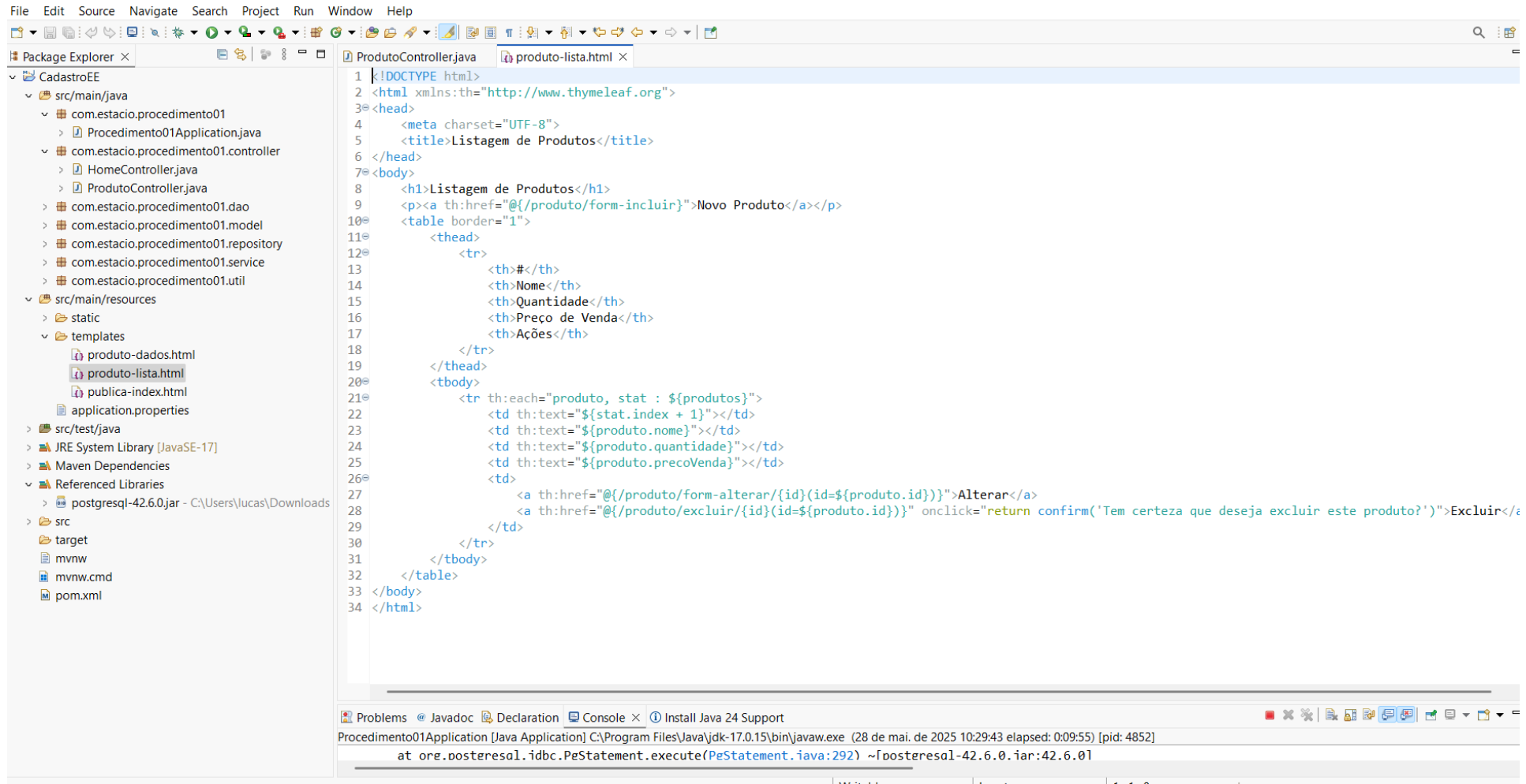


The screenshot displays an IDE window with the following components:

- Package Explorer:** Shows the project structure for 'CadastroEE'. The 'src/main/resources' folder is expanded, showing 'produto-dados.html', 'produto-lista.html', and 'publica-index.html'. The 'src/main/java' folder is also expanded, showing the package structure: 'com.estacio.procedimento01' and 'com.estacio.procedimento01.controller'.
- Editor:** Displays the code for 'ProdutoController.java'. The code includes the following methods:
 - `formIncluir`: A GET method that adds attributes 'produto' and 'acao' to the model and returns 'produto-dados'.
 - `formAlterar`: A GET method that retrieves a product by ID. If the product is not null, it adds attributes 'produto' and 'acao' to the model and returns 'produto-dados'. Otherwise, it returns a redirect to 'produto/listar'.
 - `salvarProduto`: A POST method that inserts or updates a product based on the 'acao' parameter. If 'acao' is 'incluir', it inserts the product. If 'acao' is 'alterar', it updates the product. It then returns a redirect to 'produto/listar'.
 - `excluirProduto`: A GET method that excludes a product by ID and returns a redirect to 'produto/listar'.
- Console:** Shows the output of the application, including the path 'C:\Program Files\Java\jdk-17.0.15\bin\javaw.exe' and the PID '4852'.

2º Procedimento | Interface Cadastral com Servlet e JSPs

Códigos usados neste roteiro



The screenshot shows an IDE with the following components:

- Package Explorer:** Displays the project structure for 'CadastroEE'. It includes packages like 'com.estacio.procedimento01', 'com.estacio.procedimento01.controller', and 'com.estacio.procedimento01.dao'. It also shows 'src/main/resources' with files like 'produto-dados.html', 'produto-lista.html', and 'publica-index.html'.
- Editor:** Displays the code for 'produto-lista.html'. The code is an HTML file using Thymeleaf syntax. It includes a DOCTYPE declaration, a meta charset, a title 'Listagem de Produtos', and a body containing a table with product information. The table has columns for '#', 'Nome', 'Quantidade', 'Preço de Venda', and 'Ações'. The 'Ações' column contains links to 'Alterar' and 'Excluir'.
- Console:** Shows the output of the application. It includes the message 'Procedimento01Application [Java Application] C:\Program Files\Java\jdk-17.0.15\bin\javaw.exe (28 de mai. de 2025 10:29:43 elapsed: 0:09:55) [pid: 4852]' and a line indicating a database operation: 'at org.postgresql.jdbc.PcStatement.execute(PcStatement.java:292) ~[postgresql-42.6.0.jar:42.6.0]'.

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <title>Listagem de Produtos</title>
6 </head>
7 <body>
8   <h1>Listagem de Produtos</h1>
9   <p><a th:href="@{/produto/form-incluir}">Novo Produto</a></p>
10  <table border="1">
11    <thead>
12      <tr>
13        <th>#</th>
14        <th>Nome</th>
15        <th>Quantidade</th>
16        <th>Preço de Venda</th>
17        <th>Ações</th>
18      </tr>
19    </thead>
20    <tbody>
21      <tr th:each="produto, stat : ${produtos}">
22        <td th:text="${stat.index + 1}"></td>
23        <td th:text="${produto.nome}"></td>
24        <td th:text="${produto.quantidade}"></td>
25        <td th:text="${produto.precoVenda}"></td>
26        <td>
27          <a th:href="@{/produto/form-alterar/{id}{id=${produto.id}}}">Alterar</a>
28          <a th:href="@{/produto/excluir/{id}{id=${produto.id}}}" onclick="return confirm('Tem certeza que deseja excluir este produto?')">Excluir</a>
29        </td>
30      </tr>
31    </tbody>
32  </table>
33 </body>
34 </html>
```

2º Procedimento | Interface Cadastral com Servlet e JSPs

Listagem de Produtos

[Novo Produto](#)

#	Nome	Quantidade	Preço de Venda	Ações
1	banana	100	5.0	Alterar Excluir
2	laranja	500	2.0	Alterar Excluir
3	manga	800	4.0	Alterar Excluir

2º Procedimento | Interface Cadastral com Servlet e JSPs

Códigos usados neste roteiro

The screenshot shows an IDE with the following components:

- Package Explorer:** Displays the project structure for 'CadastroEE'. The package 'com.estacio.procedimento01.controller' is expanded, showing 'ProdutoController.java'.
- Editor:** Displays the code for 'ProdutoController.java'. The code is as follows:

```
1 package com.estacio.procedimento01.controller;
2
3 import com.estacio.procedimento01.dao.ProdutoDAO;
4 import com.estacio.procedimento01.model.Produto;
5 import org.springframework.stereotype.Controller;
6 import org.springframework.ui.Model;
7 import org.springframework.web.bind.annotation.*;
8
9 @Controller
10 @RequestMapping("/produto")
11 public class ProdutoController {
12
13     private final ProdutoDAO produtoDAO = new ProdutoDAO();
14
15     // Página de Cadastro
16     @GetMapping("/dados")
17     public String paginaCadastro(Model model) {
18         model.addAttribute("produto", new Produto());
19         model.addAttribute("acao", "incluir");
20         return "produto-dados";
21     }
22
23     // Listar produtos
24     @GetMapping("/listar")
25     public String listarProdutos(Model model) {
26         model.addAttribute("produtos", produtoDAO.listarProdutos());
27         return "produto-lista";
28     }
29
30     // Formulário de inclusão
31     @GetMapping("/form-incluir")
32     public String formIncluir(Model model) {
33         model.addAttribute("produto", new Produto());
34     }
35 }
```
- Console:** Displays the output of the application. The output is as follows:

```
Procedimento01Application [Java Application] C:\Program Files\Java\jdk-17.0.15\bin\javaw.exe (28 de mai. de 2025 10:29:43 elapsed: 1:37:19) [pid: 4852]
2025-05-28 12:03:13.380 INFO 4852 --- [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
2025-05-28 12:03:36.901 INFO 4852 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat-1].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-05-28 12:03:36.901 INFO 4852 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-05-28 12:03:36.903 INFO 4852 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
```

The status bar at the bottom indicates the current file is 'com.estacio.procedimento01.Procedimento01Application.java' located in 'CadastroEE/src/main/java'.

2º Procedimento | Interface Cadastral com Servlet e JSPs

Códigos usados neste roteiro

The screenshot shows an IDE with the following structure:

- Package Explorer:
 - CadastroEE
 - src/main/java
 - com.estacio.procedimento01
 - Procedimento01Application.java
 - com.estacio.procedimento01.controller
 - HomeController.java
 - ProdutoController.java
 - com.estacio.procedimento01.dao
 - ProdutoDAO.java
 - com.estacio.procedimento01.model
 - Produto.java
 - com.estacio.procedimento01.repository
 - ProdutoRepository.java
 - com.estacio.procedimento01.service
 - ProdutoService.java
 - com.estacio.procedimento01.util
 - DatabaseConnection.java
 - src/main/resources
 - static
 - templates
 - produto-dados.html
 - produto-lista.html
 - publica-index.html
 - application.properties
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - Referenced Libraries
 - postgresql-42.6.0.jar - C:\Users\lucas\Downloads
 - src
 - target
 - mvnw
 - mvnw.cmd
 - pom.xml

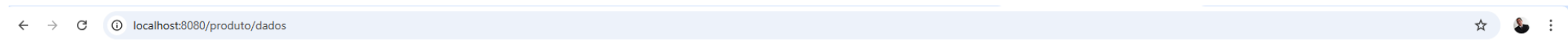
The main editor shows the code for `ProdutoController.java` and `produto-dados.html`:

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <title>Cadastro de Produto</title>
6 </head>
7 <body>
8   <h1>Cadastro de Produto</h1>
9
10  <!-- Formulário para incluir ou alterar produto -->
11  <form th:action="@{/produto/salvar}" method="post">
12    <!-- Campo oculto para armazenar a ação (incluir ou alterar) -->
13    <input type="hidden" name="acao" th:value="${acao}" />
14
15    <!-- Nome do Produto -->
16    <label for="nome">Nome:</label>
17    <input type="text" id="nome" name="nome" th:value="${produto.nome}" required /><br/>
18
19    <!-- Quantidade do Produto -->
20    <label for="quantidade">Quantidade:</label>
21    <input type="number" id="quantidade" name="quantidade" th:value="${produto.quantidade}" required /><br/>
22
23    <!-- Preço de Venda do Produto -->
24    <label for="precoVenda">Preço de Venda:</label>
25    <input type="number" step="0.01" id="precoVenda" name="precoVenda" th:value="${produto.precoVenda}" required /><br/>
26
27    <!-- Botão de Envio -->
28    <button type="submit" th:text="${acao == 'incluir' ? 'Adicionar Produto' : 'Salvar Alterações'}"></button>
29  </form>
30 </body>
31 </html>
```

The bottom console shows the following output:

```
Procedimento01Application [Java Application] C:\Program Files\Java\jdk-17.0.15\bin\javaw.exe (28 de mai. de 2025 10:29:43 elapsed: 1:38:01) [pid: 4852]
2025-05-28 12:03:13.380 INFO 4852 --- [ restartedMain] .ConditionEvaluationDeltaLoggingListener : Condition evaluation unchanged
2025-05-28 12:03:36.901 INFO 4852 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat-1].[localhost].[] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-05-28 12:03:36.901 INFO 4852 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-05-28 12:03:36.903 INFO 4852 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
```

2º Procedimento | Interface Cadastral com Servlet e JSPs



Cadastro de Produto

Nome:

Quantidade:

Preço de Venda:

Análise e conclusão:

1.Como funciona o padrão Front Controller, e como ele é implementado em um aplicativo Web Java, na arquitetura MVC?

O Front Controller é um padrão de design arquitetural amplamente utilizado em aplicativos Web para centralizar o tratamento de requisições HTTP. Ele atua como um único ponto de entrada para todas as requisições recebidas pelo sistema, delegando a execução de tarefas específicas a outros componentes do sistema.

2.Quais as diferenças e semelhanças entre Servlets e JSPs?

Os Servlets e JSPs (JavaServer Pages) são tecnologias amplamente utilizadas no desenvolvimento de aplicações web em Java. Ambos fazem parte do ecossistema Java EE (Jakarta EE) e são usados para criar aplicativos dinâmicos. No entanto, eles têm propósitos e características distintas.

3.Qual a diferença entre um redirecionamento simples e o uso do método forward, a partir do RequestDispatcher? Para que servem parâmetros e atributos nos objetos HttpRequest?

Diferença entre Redirecionamento e Forward

1.Redirecionamento (sendRedirect)

O servidor envia uma resposta ao cliente (browser) com um novo endereço (URL).

O cliente faz uma nova requisição para o novo endereço.

Altera a URL no navegador.

Usado para redirecionar para diferentes servidores ou domínios.

2.Forward (RequestDispatcher.forward)

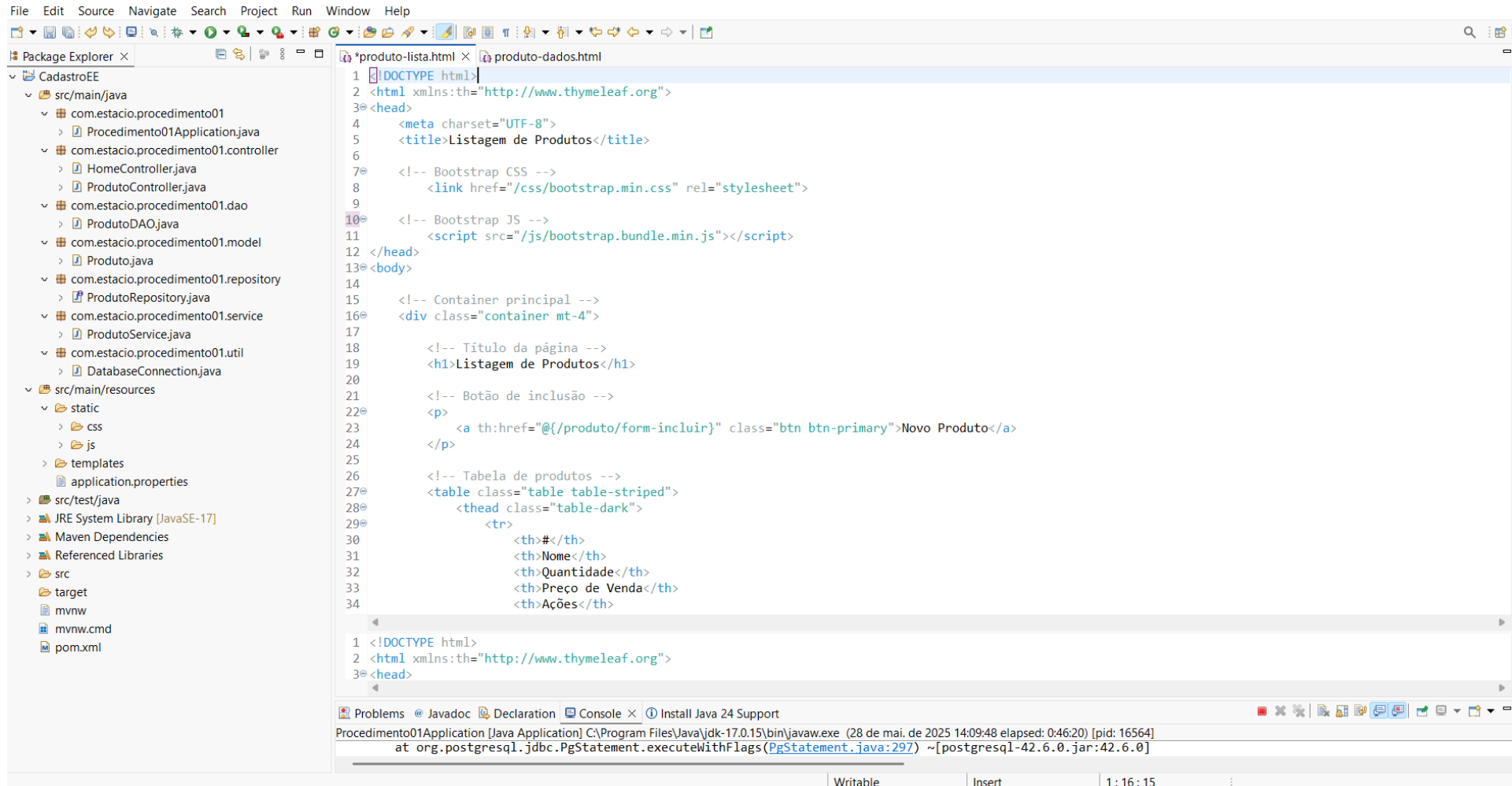
A requisição é encaminhada internamente no servidor, sem envolver o cliente.

A URL no navegador não muda.

Usado para transferir o controle dentro do mesmo servidor.

3º Procedimento | Melhorando o Design da Interface

Códigos usados neste roteiro



The screenshot displays an IDE interface with a Package Explorer on the left and a code editor on the right. The Package Explorer shows a project named 'CadastroEE' with a package structure including 'src/main/java' and 'src/main/resources'. The code editor shows the content of 'produto-lista.html', which is an HTML file using Bootstrap for styling. The HTML includes a title 'Listagem de Produtos', a link to Bootstrap CSS, and a script for Bootstrap JS. It also features a button 'Novo Produto' and a table with columns for product details.

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <title>Listagem de Produtos</title>
6
7   <!-- Bootstrap CSS -->
8   <link href="/css/bootstrap.min.css" rel="stylesheet">
9
10  <!-- Bootstrap JS -->
11  <script src="/js/bootstrap.bundle.min.js"></script>
12 </head>
13 <body>
14
15   <!-- Container principal -->
16   <div class="container mt-4">
17
18     <!-- Título da página -->
19     <h1>Listagem de Produtos</h1>
20
21     <!-- Botão de inclusão -->
22     <p>
23       <a th:href="@{/produto/form-incluir}" class="btn btn-primary">Novo Produto</a>
24     </p>
25
26     <!-- Tabela de produtos -->
27     <table class="table table-striped">
28       <thead class="table-dark">
29         <tr>
30           <th>#</th>
31           <th>Nome</th>
32           <th>Quantidade</th>
33           <th>Preço de Venda</th>
34           <th>Ações</th>
35         </tr>
36       </thead>
37     </table>
38
39   </div>
40 </body>
41 </html>
```

The bottom of the IDE shows a console window with the following output:

```
Procedimento01Application [Java Application] C:\Program Files\Java\jdk-17.0.15\bin\javaw.exe (28 de mai. de 2025 14:09:48 elapsed: 0:46:20) [pid: 16564]
at org.postgresql.jdbc.PgStatement.executeQueryWithFlags(PgStatement.java:297) ~[postgresql-42.6.0.jar:42.6.0]
```

3º Procedimento | Melhorando o Design da Interface

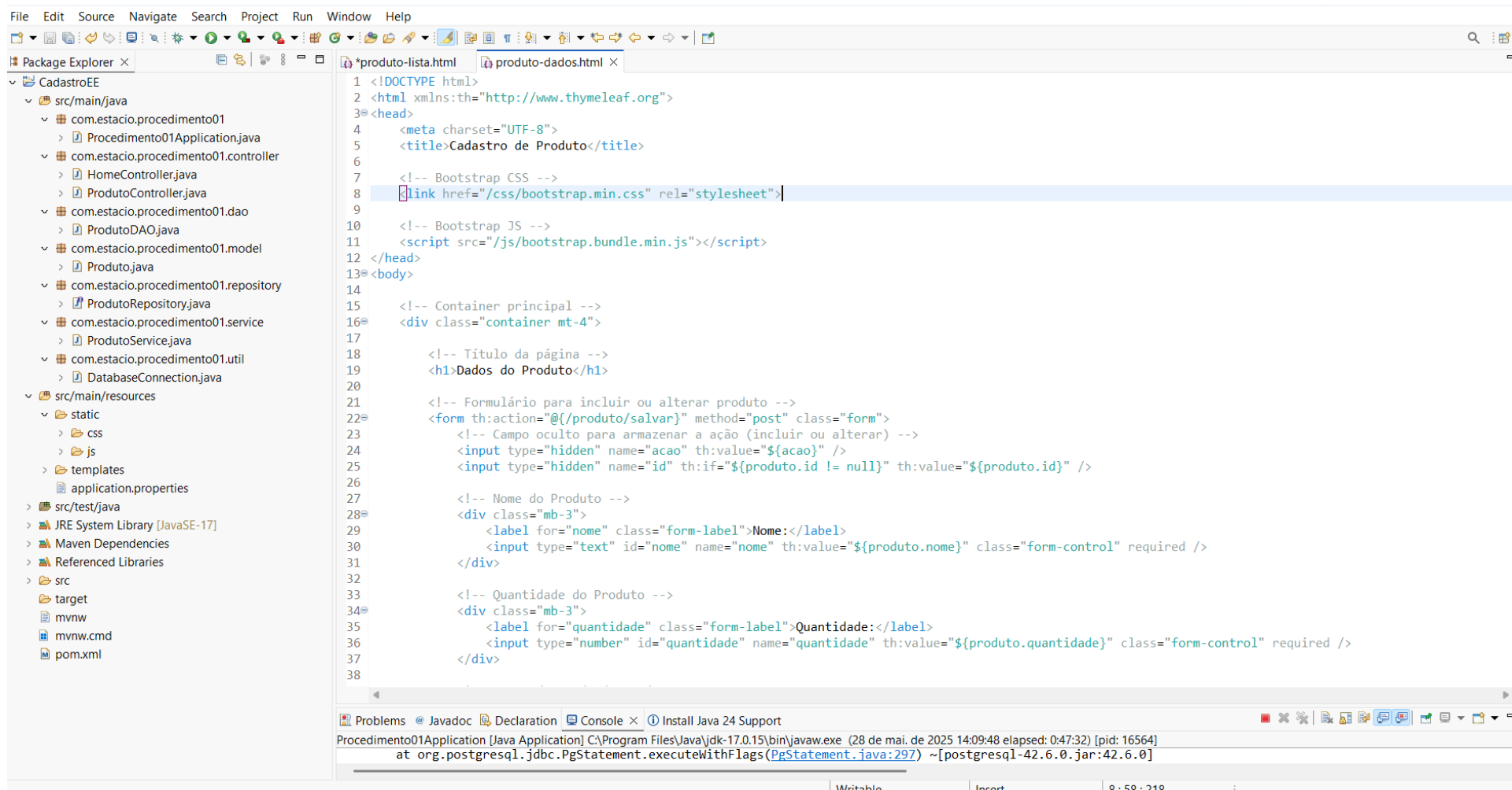
Listagem de Produtos

Novo Produto

#	Nome	Quantidade	Preço de Venda	Ações	
1	banana	100	5.0	Alterar	Excluir
2	laranja	500	2.0	Alterar	Excluir
3	manga	800	4.0	Alterar	Excluir

3º Procedimento | Melhorando o Design da Interface

Códigos usados neste roteiro



The screenshot shows an IDE with the following components:

- Package Explorer:** Displays the project structure for 'CadastroEE'. It includes a 'src/main/java' directory with sub-packages like 'com.estacio.procedimento01' and 'com.estacio.procedimento01.controller'. It also shows 'src/main/resources' with 'static' and 'templates' folders, and 'src/test/java'.
- Editor:** Shows the content of 'produto-lista.html'. The code is an HTML document using Bootstrap and Thymeleaf. It includes a meta charset, a title 'Cadastro de Produto', and links to Bootstrap CSS and JS. The main content area contains a form for adding or updating a product, with fields for 'acao' (hidden), 'id' (hidden), 'nome' (text), and 'quantidade' (number). The form uses Thymeleaf expressions like `\${produto.id}` and `\${produto.nome}`.
- Console:** Shows the output of the application. It indicates that the application is running on 'C:\Program Files\Java\jdk-17.0.15\bin\javaw.exe' and that it is using 'org.postgresql.jdbc.PgStatement.executeWithFlags(PgStatement.java:297)'.

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <title>Cadastro de Produto</title>
6
7   <!-- Bootstrap CSS -->
8   <link href="/css/bootstrap.min.css" rel="stylesheet">
9
10  <!-- Bootstrap JS -->
11  <script src="/js/bootstrap.bundle.min.js"></script>
12 </head>
13 <body>
14
15   <!-- Container principal -->
16   <div class="container mt-4">
17
18     <!-- Título da página -->
19     <h1>Dados do Produto</h1>
20
21     <!-- Formulário para incluir ou alterar produto -->
22     <form th:action="@{/produto/salvar}" method="post" class="form">
23       <!-- Campo oculto para armazenar a ação (incluir ou alterar) -->
24       <input type="hidden" name="acao" th:value="${acao}" />
25       <input type="hidden" name="id" th:if="${produto.id != null}" th:value="${produto.id}" />
26
27       <!-- Nome do Produto -->
28       <div class="mb-3">
29         <label for="nome" class="form-label">Nome:</label>
30         <input type="text" id="nome" name="nome" th:value="${produto.nome}" class="form-control" required />
31       </div>
32
33       <!-- Quantidade do Produto -->
34       <div class="mb-3">
35         <label for="quantidade" class="form-label">Quantidade:</label>
36         <input type="number" id="quantidade" name="quantidade" th:value="${produto.quantidade}" class="form-control" required />
37       </div>
38     </form>
39   </div>
40 </body>
41 </html>
```

3º Procedimento | Melhorando o Design da Interface

Dados do Produto

Nome:

Quantidade:

Preço de Venda:

Adicionar Produto



Pesquisar



Análise e conclusão:

1.Como o framework Bootstrap é utilizado?

O Bootstrap é um framework front-end utilizado para desenvolver interfaces web responsivas e estilizadas de forma rápida e eficiente. Ele oferece componentes, classes CSS e utilitários prontos que facilitam a criação de layouts modernos e compatíveis com diferentes dispositivos (desktops, tablets e smartphones).

2.Por que o Bootstrap garante a independência estrutural do HTML?

O Bootstrap promove a independência estrutural do HTML ao permitir que o desenvolvedor use classes e componentes pré-construídos para estilizar e organizar o conteúdo, sem precisar alterar a estrutura base do HTML ou escrever CSS/JS customizado. Isso resulta em um código mais limpo, modular e fácil de manter.

3.Qual a relação entre o Bootstrap e a responsividade da página?

A relação entre o Bootstrap e a responsividade é direta: o framework fornece ferramentas prontas (grid system, breakpoints e componentes responsivos) que simplificam a criação de layouts adaptáveis, garantindo que a página seja visualmente consistente e funcional em qualquer dispositivo, sem necessidade de codificação adicional. Isso torna o Bootstrap uma escolha popular para desenvolvimento web moderno.

LINK DO GITHUB:

https://github.com/mauriciocampos1234/Missao_Pratica_4_BackEnd_Estacio