



Estácio

CAMPUS: Polo Planalto - BH - MG

CURSO: Desenvolvimento FullStack

DISCIPLINA: Vamos integrar sistemas

TURMA: 2025.1

SEMESTRE LETIVO: Primeiro Semestre (2025)

ALUNO: Bruno Ricardo Viana Venturelli

MATRÍCULA: 202401226726

LINK DO GITHUB:

https://github.com/DevBrN01/Trabalho_n4_BackEnd

Título da prática:

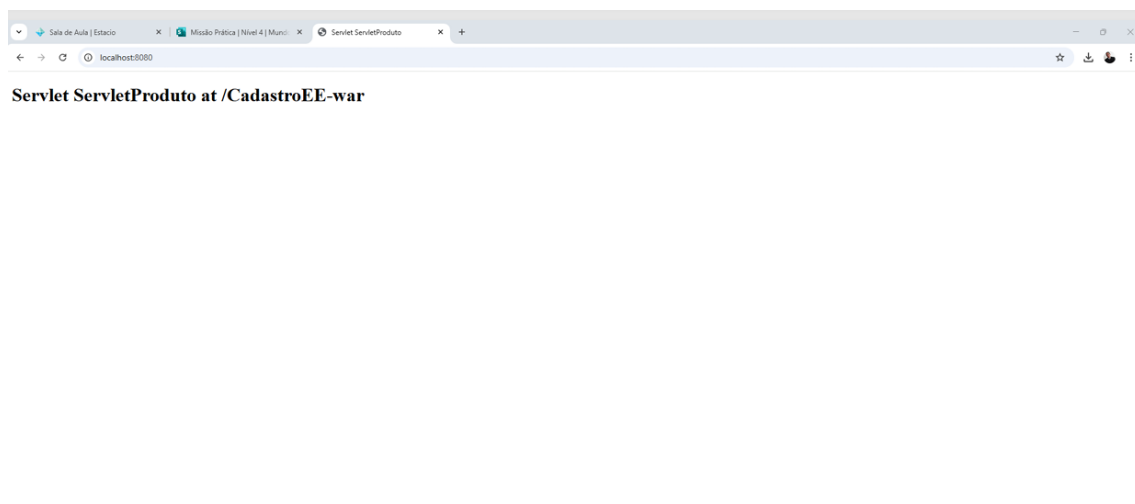
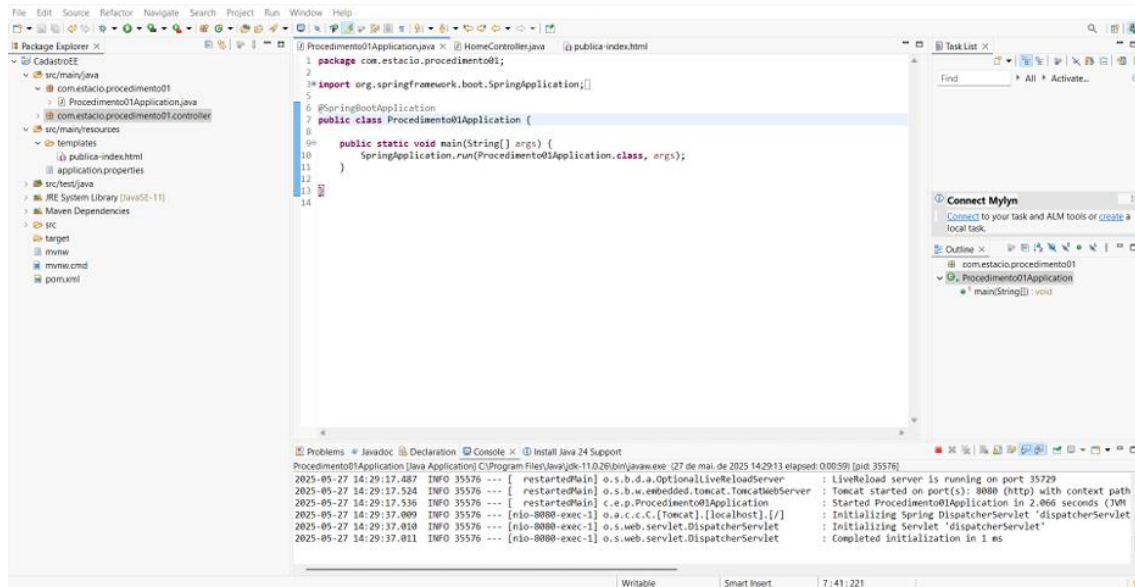
Implementação de sistema cadastral com interface Web, baseado nas tecnologias de Servlets, JPA e JEE..

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.

Utilizar a biblioteca Bootstrap para melhoria do design.

1º Procedimento | Camadas de Persistência e Controle Códigos usados neste roteiro



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?

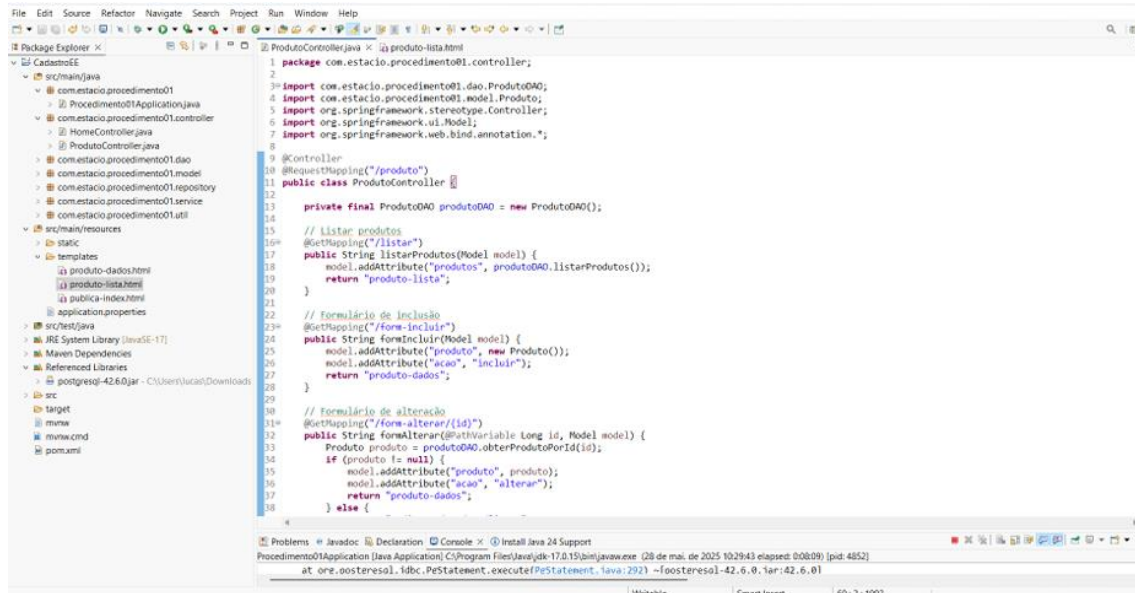
Servlets são componentes Java que processam requisições HTTP em aplicações web, gerando respostas dinâmicas (como HTML).

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

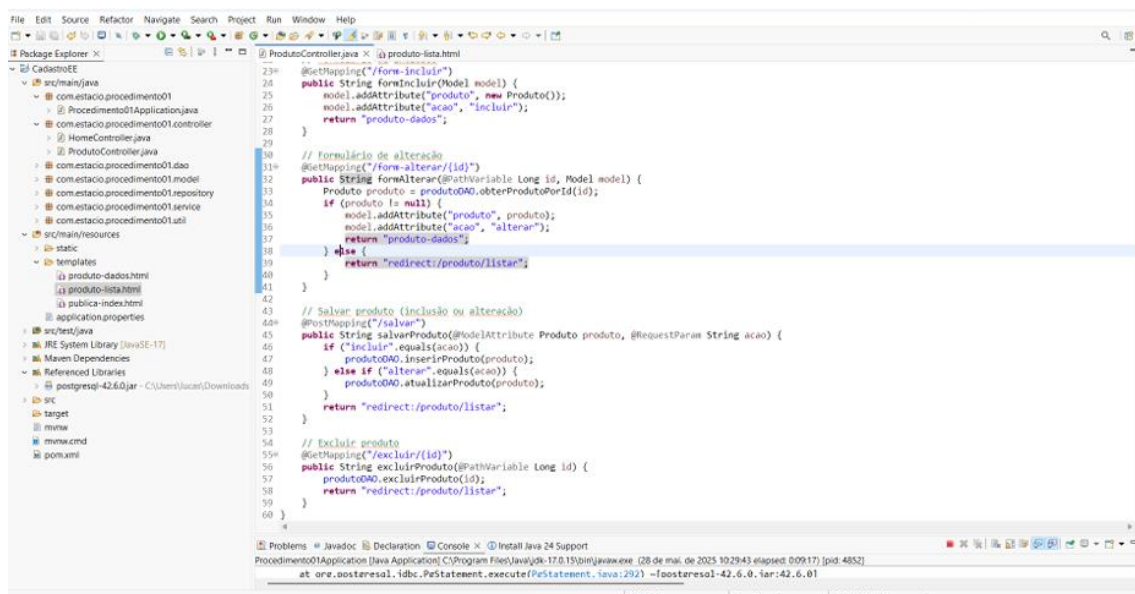
A comunicação entre **Servlets** e **Session Beans (EJBs)** é feita por **injeção de dependência** ou **lookup JNDI**, permitindo que o **Servlet** chame métodos do **EJB**.

2º Procedimento | Interface Cadastral com Servlet e JSPs

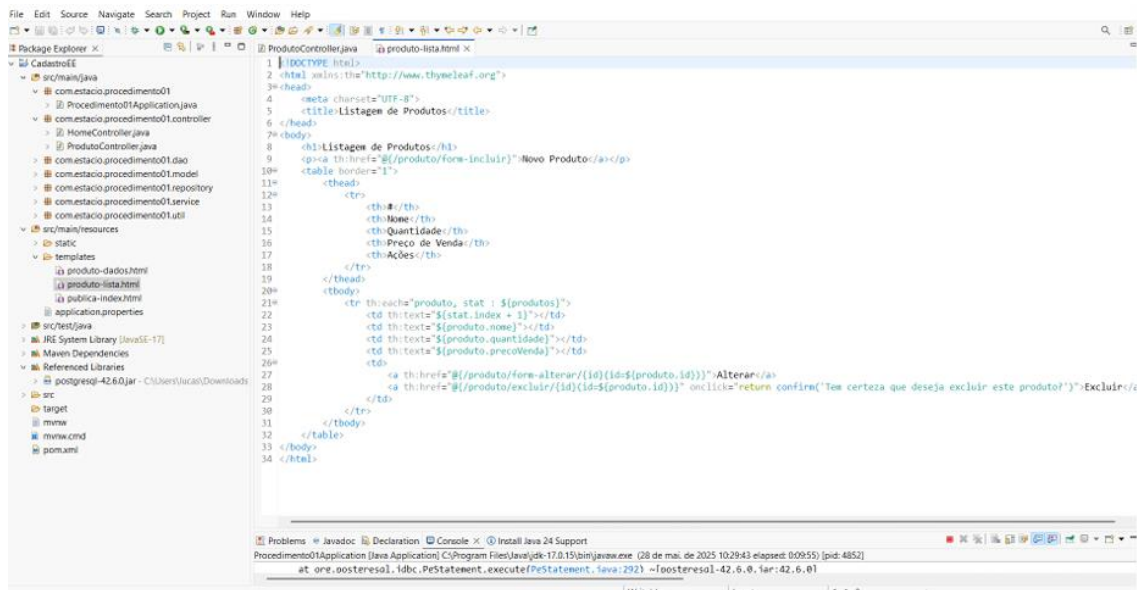
Códigos usados neste roteiro



```
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 {
39             return "redirect:/produto/lista";
40         }
41     }
42
43     // Salvar produto (inclusão ou alteração)
44     @PostMapping("/salvar")
45     public String salvarProduto(@ModelAttribute Produto produto, @RequestParam String acao) {
46         if ("incluir".equals(acao)) {
47             produtoDAO.inserirProduto(produto);
48         } else if ("alterar".equals(acao)) {
49             produtoDAO.atualizarProduto(produto);
50         }
51         return "redirect:/produto/lista";
52     }
53
54     // Excluir produto
55     @GetMapping("/excluir/{id}")
56     public String excluirProduto(@PathVariable Long id) {
57         produtoDAO.excluirProduto(id);
58         return "redirect:/produto/lista";
59     }
60 }
```



```
23 // Formulário de inclusão
24 @GetMapping("/form-incluir")
25 public String formIncluir(Model model) {
26     model.addAttribute("produto", new Produto());
27     model.addAttribute("acao", "incluir");
28     return "produto-dados";
29 }
30
31 // Formulário de alteração
32 @GetMapping("/form-alterar/{id}")
33 public String formAlterar(@PathVariable Long id, Model model) {
34     Produto produto = produtoDAO.obterProdutoPorId(id);
35     if (produto != null) {
36         model.addAttribute("produto", produto);
37         model.addAttribute("acao", "alterar");
38         return "produto-dados";
39     } else {
40         return "redirect:/produto/lista";
41     }
42 }
43
44 // Salvar produto (inclusão ou alteração)
45 @PostMapping("/salvar")
46 public String salvarProduto(@ModelAttribute Produto produto, @RequestParam String acao) {
47     if ("incluir".equals(acao)) {
48         produtoDAO.inserirProduto(produto);
49     } else if ("alterar".equals(acao)) {
50         produtoDAO.atualizarProduto(produto);
51     }
52     return "redirect:/produto/lista";
53 }
54
55 // Excluir produto
56 @GetMapping("/excluir/{id}")
57 public String excluirProduto(@PathVariable Long id) {
58     produtoDAO.excluirProduto(id);
59     return "redirect:/produto/lista";
60 }
```

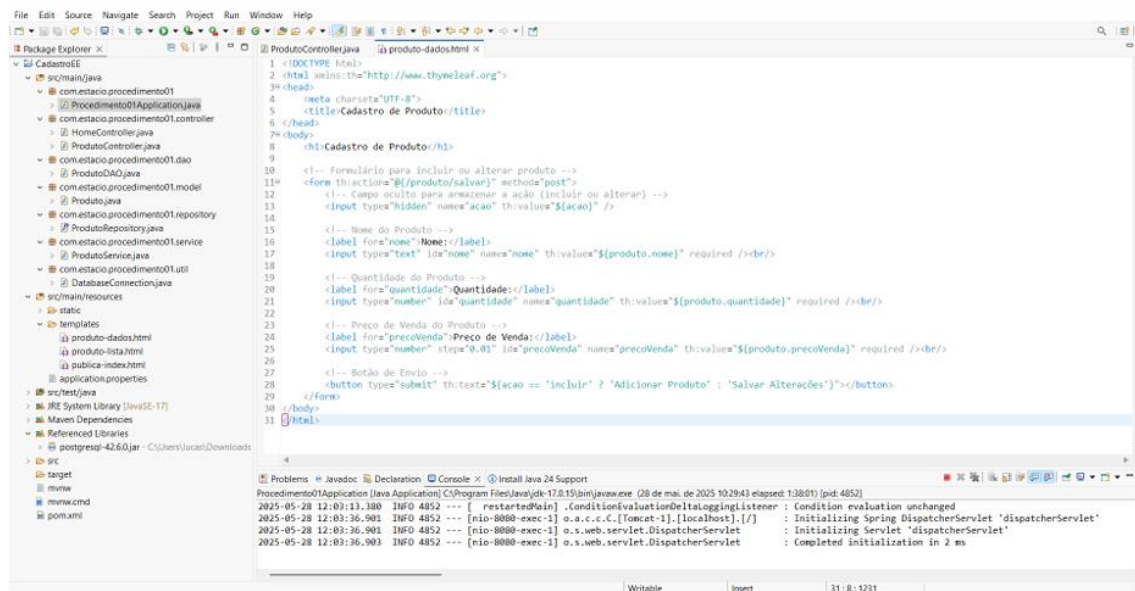
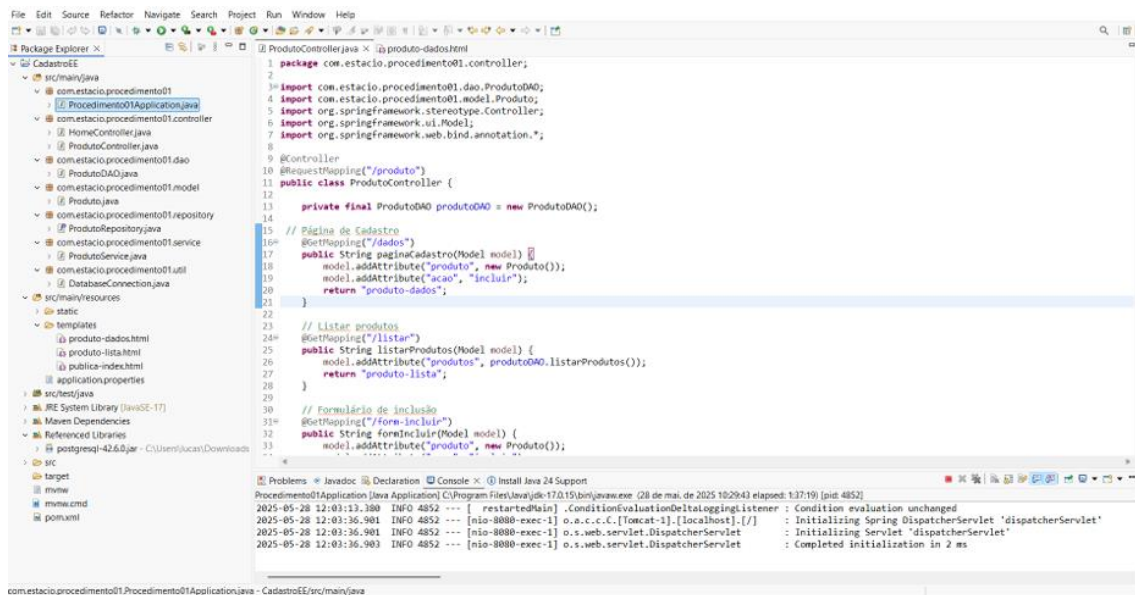


localhost:8080/produto/listar

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



localhost:8080/produto/dados

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.

2.Forward (RequestDispatcher.forward)

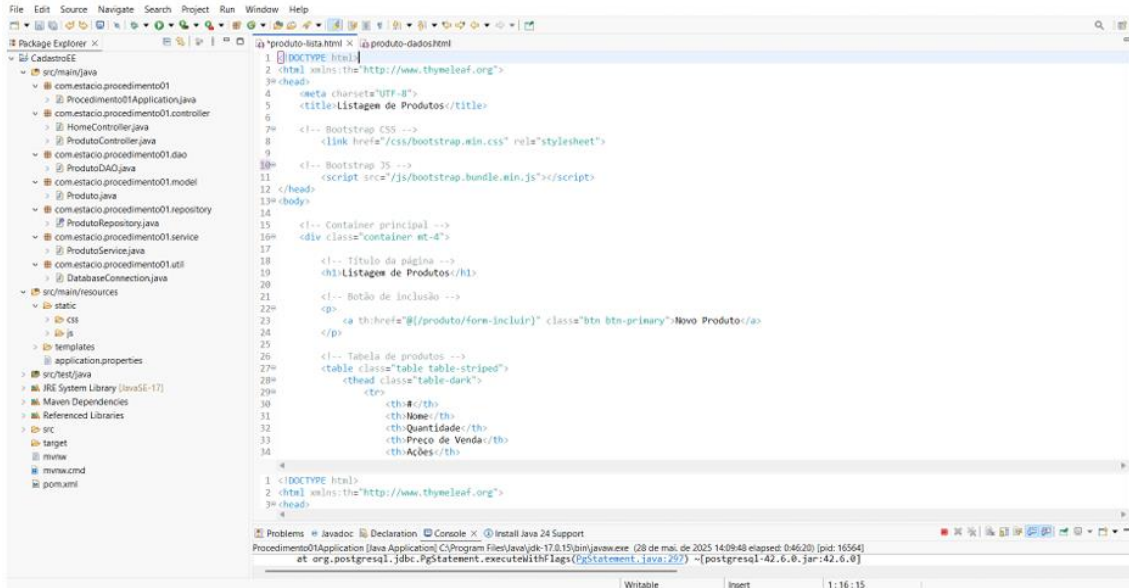
A requisição é encaminhada internamente no servidor, sem envolver o cliente.

A URL no navegador não muda.

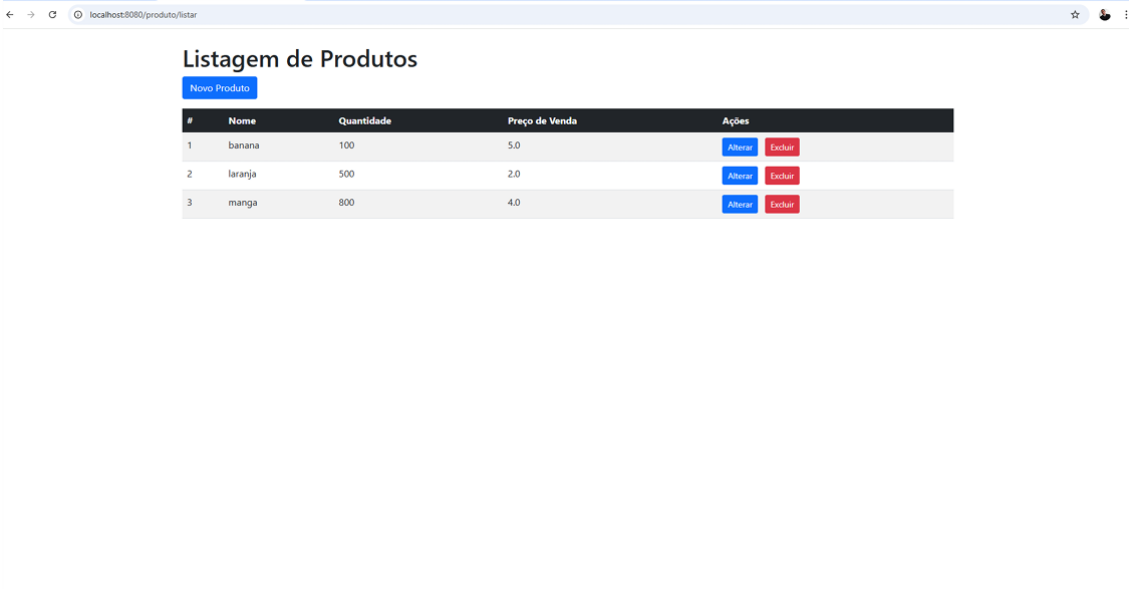
Usado para transferir o controle dentro do mesmo servidor.

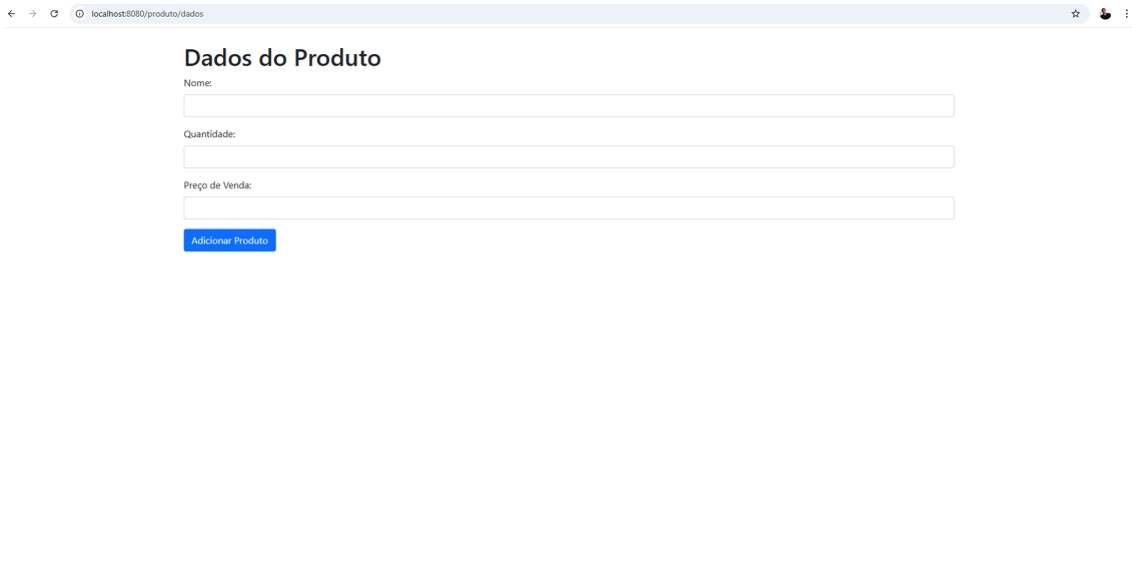
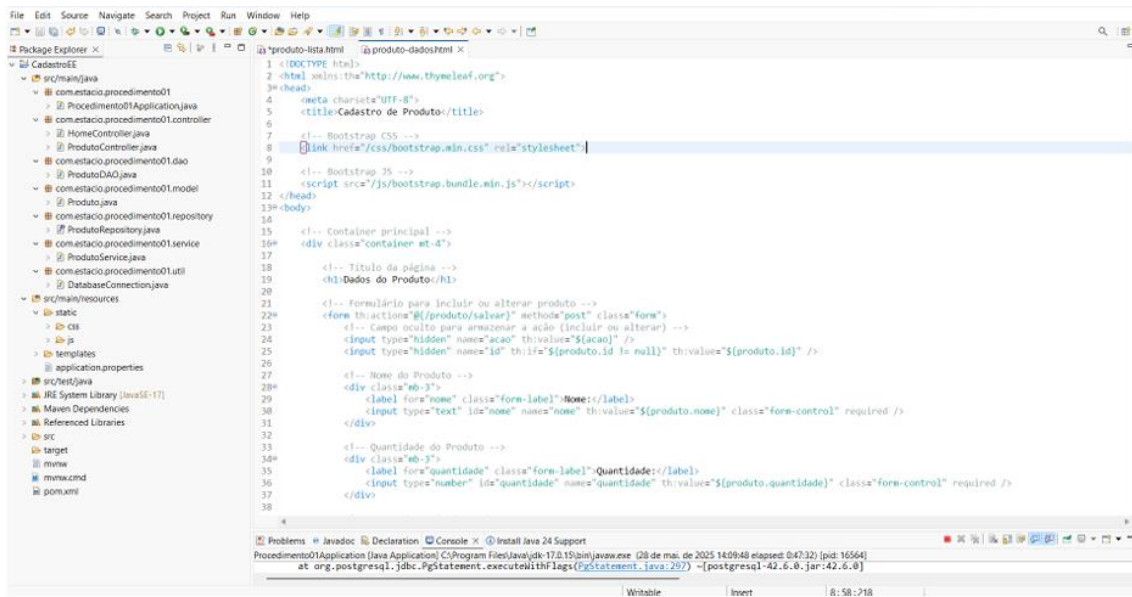
3º Procedimento | Melhorando o Design da Interface

Códigos usados neste roteiro



```
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 <h3>Listagem de Produtos</h3>
20
21 <!-- Botão de inclusão -->
22 <a th:href="@{/produto/form-incluir}" class="btn btn-primary">Novo Produto</a>
23
24 </div>
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 </body>
40 </html>
```





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.