



UNIVERSIDADE ESTÁCIO DE SÁ
PROGRAMAÇÃO FULL STACK
PÓLO NORTE SHOPPING

CLEYDSON ASSIS COELHO

**MISSÃO PRÁTICA 3 MUNDO 3
PROCEDIMENTO 3**

RPG0017 - Vamos integrar sistemas

RIO DE JANEIRO

2025

Objetivos da prática

- Implementar persistência com base em JPA.
- Implementar regras de negócio na plataforma JEE, através de EJBs.
- Implementar sistema cadastral Web com base em Servlets e JSPs.
- Utilizar a biblioteca Bootstrap para melhoria do design.
- No final do exercício, o aluno terá criado todos os elementos necessários para exibição e entrada de dados na plataforma Java Web, tornando-se capacitado para lidar com contextos reais de aplicação.

ProdutoDados.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="cadastroe.model.Produto" %>
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Dados do Produto</title>

  <!-- Bootstrap CSS via CDN -->
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/css/bootstrap.min.css"
    rel="stylesheet"
    integrity="sha384-
4Q6Gf2aSP4eDXB8Miphtr37CMZZQ5oXLH2yaXMJ2w8e2ZtHTl7GptT4jmndRuHDT"
    crossorigin="anonymous">
</head>
<body class="container mt-4">

  <%
    String acao = request.getParameter("acao");
    Produto produto = (Produto) request.getAttribute("produto");
    boolean isAlterar = "formAlterar".equals(acao);
  %>

  <h1 class="mb-4">Dados do Produto</h1>

  <form action="produto" method="post" class="form">
    <input type="hidden" name="acao" value="<%= acao %>">
    <% if (isAlterar) { %>
      <input type="hidden" name="id" value="<%= produto.getIdProduto() %>">
    <% } %>

    <div class="mb-3">
      <label for="nome" class="form-label">Nome:</label>
      <input
        type="text"
        class="form-control"
        id="nome"

```

```

        name="nome"
        required
        value="<%= produto != null ? produto.getNome() : "" %>">
    </div>

    <div class="mb-3">
        <label for="quantidade" class="form-label">Quantidade:</label>
        <input
            type="number"
            class="form-control"
            id="quantidade"
            name="quantidade"
            min="0"
            value="<%= produto != null ? produto.getQuantidade() : 0 %>">
    </div>

    <div class="mb-3">
        <label for="precoVenda" class="form-label">Preço de Venda:</label>
        <input
            type="text"
            class="form-control"
            id="precoVenda"
            name="precoVenda"
            required
            value="<%= produto != null ? produto.getPrecoVenda() : "" %>">
    </div>

    <button type="submit" class="btn btn-primary">
        <%= isAlterar ? "Alterar Produto" : "Incluir Produto" %>
    </button>
    <a href="produto?acao=listar" class="btn btn-secondary ms-2">Cancelar</a>
</form>

<!-- Bootstrap JS Bundle via CDN -->
<script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
j1CDi7MgGQ12Z7Qab0qlWQ/Qqz24Gc6BM0thvEMVjHnfYGF0rmFCozFSxQBxwHKO"
    crossorigin="anonymous"></script>
</body>
</html>

```

ProdutoLista.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.util.List, cadastroee.model.Produto" %>
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Listagem de Produtos</title>

    <!-- Bootstrap CSS via CDN -->
    <link

```

```

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
4Q6Gf2aSP4eDXB8Miphtr37CMZZQ5oXLH2yaXMJ2w8e2ZtHTl7GptT4jmndRuHDT"
crossorigin="anonymous">
</head>
<body class="container">

<h1 class="mt-4">Listagem de Produtos</h1>

<!-- Link para abrir o formulário de inclusão -->
<p>
  <a href="produto?acao=formIncluir"
    class="btn btn-primary m-2">
    Novo Produto
  </a>
</p>

<%
  @SuppressWarnings("unchecked")
  List<Produto> produtos = (List<Produto>) request.getAttribute("produtos");
  if (produtos == null || produtos.isEmpty()) {
%>
    <div class="alert alert-info">Nenhum produto encontrado.</div>
<%
  } else {
%>
<table class="table table-striped">
  <thead class="table-dark">
    <tr>
      <th>#</th>
      <th>Nome</th>
      <th>Quantidade</th>
      <th>Preço de Venda</th>
      <th>Opções</th>
    </tr>
  </thead>
  <tbody>
    <%
      for (Produto p : produtos) {
%>
    <tr>
      <td><%= p.getIdProduto() %></td>
      <td><%= p.getNome() %></td>
      <td><%= p.getQuantidade() %></td>
      <td>R$ <%= String.format("%.2f", p.getPrecoVenda()) %></td>
      <td>
        <a href="produto?acao=formAlterar&id=<%= p.getIdProduto() %>"
          class="btn btn-primary btn-sm">
            Alterar
        </a>
        <a href="produto?acao=excluir&id=<%= p.getIdProduto() %>"
          class="btn btn-danger btn-sm"
          onclick="return confirm('Confirma exclusão deste produto?');">
            Excluir

```

```

        </a>
      </td>
    </tr>
  <%
  }
  %>
</tbody>
</table>
<%
}
%>

<!-- Bootstrap Bundle JS via CDN -->
<script
  src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-
j1CDi7MgGQ12Z7Qab0qIWQ/Qqz24Gc6BM0thvEMVjHnfYGF0rmFCozFSxQBxwHKO"
  crossorigin="anonymous">
</script>
</body>
</html>

```

Produto.java

```

package cadastroee.model;

import java.io.Serializable;
import jakarta.persistence.Basic;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.NamedQueries;
import jakarta.persistence.NamedQuery;
import jakarta.persistence.Table;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import jakarta.xml.bind.annotation.XmlRootElement;

@Entity
@Table(name = "Produto")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Produto.findAll", query = "SELECT p FROM Produto p"),
    @NamedQuery(name = "Produto.findById", query = "SELECT p FROM Produto p
WHERE p.idProduto = :idProduto"),
    @NamedQuery(name = "Produto.findByName", query = "SELECT p FROM Produto p
WHERE p.nome = :nome"),
    @NamedQuery(name = "Produto.findByPrecoVenda", query = "SELECT p FROM Produto p
WHERE p.precoVenda = :precoVenda"),
    @NamedQuery(name = "Produto.findByQuantidade", query = "SELECT p FROM Produto p
WHERE p.quantidade = :quantidade")
})

```

```
public class Produto implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @Basic(optional = false)
    @NotNull
    @Column(name = "id_produto")
    private Integer idProduto;

    @Size(max = 100)
    @Column(name = "nome")
    private String nome;

    @Column(name = "quantidade")
    private Integer quantidade;

    @Column(name = "preco_venda")
    private Float precoVenda;

    public Produto() {}

    public Produto(Integer idProduto) {
        this.idProduto = idProduto;
    }

    // --- getters & setters ---

    public Integer getIdProduto() {
        return idProduto;
    }

    public void setIdProduto(Integer idProduto) {
        this.idProduto = idProduto;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public Integer getQuantidade() {
        return quantidade;
    }

    public void setQuantidade(Integer quantidade) {
        this.quantidade = quantidade;
    }
}
```

```

    public Float getPrecoVenda() {
        return precoVenda;
    }

    public void setPrecoVenda(Float precoVenda) {
        this.precoVenda = precoVenda;
    }

    // hashCode, equals e toString continuam iguais
}

```

servletProdutoFC.java

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="cadastroee.model.Produto" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Dados do Produto</title>
</head>
<body>
    <h1><%= (request.getParameter("acao").equals("formAlterar") ? "Alterar" : "Incluir") %>
Produto</h1>
    <%
        // Ao alterar, o Servlet já colocou o atributo "produto" no request
        Produto produto = (Produto) request.getAttribute("produto");
    %>
    <form action="produto" method="post">
        <!-- Indica se vai incluir ou alterar -->
        <input type="hidden" name="acao" value="<%= request.getParameter("acao") %>" />
        <% if (produto != null) { %>
            <!-- Só precisa do id no formAlterar -->
            <input type="hidden" name="id" value="<%= produto.getIdProduto() %>" />
        <% } %>

        <label for="nome">Nome:</label>
        <input type="text" id="nome" name="nome"
            value="<%= (produto != null ? produto.getNome() : "") %>" required />
        <br />

        <label for="quantidade">Quantidade:</label>
        <input type="number" id="quantidade" name="quantidade" min="0"
            value="<%= (produto != null ? produto.getQuantidade() : 0) %>" />
        <br />

        <label for="precoVenda">Preço de Venda:</label>
        <input type="text" id="precoVenda" name="precoVenda"
            value="<%= (produto != null ? produto.getPrecoVenda() : "") %>" required />
        <br />
    </form>

```

```

<button type="submit">
  <%= (request.getParameter("acao").equals("formAlterar") ? "Salvar Alterações" : "Incluir Produto")
%>
</button>
&nbsp;<a href="produto?acao=listar">Cancelar</a>
</form>
</body>
</html>

```

web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="4.0"
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd">

  <!-- Servlet simples para demonstração -->
  <servlet>
    <servlet-name>ServletProduto</servlet-name>
    <servlet-class>cadastroee.servlets.ServletProduto</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ServletProduto</servlet-name>
    <url-pattern>/produtos-simples</url-pattern>
  </servlet-mapping>

  <!-- Front Controller para CRUD de Produto -->
  <servlet>
    <servlet-name>ServletProdutoFC</servlet-name>
    <servlet-class>cadastroee.servlets.ServletProdutoFC</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ServletProdutoFC</servlet-name>
    <url-pattern>/produto</url-pattern>
  </servlet-mapping>

  <session-config>
    <session-timeout>30</session-timeout>
  </session-config>
</web-app>

```

persistence.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0"
  xmlns="http://java.sun.com/xml/ns/persistence"

```

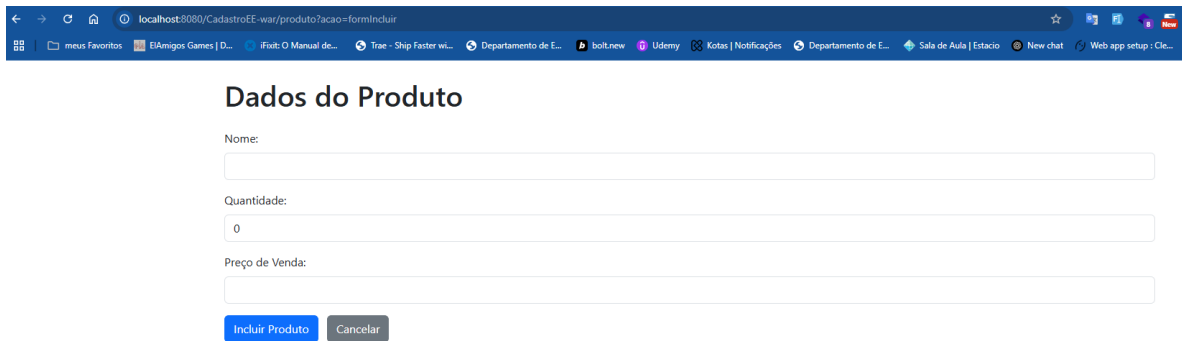


```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
<persistence-unit name="CadastroEE-ejbPU" transaction-type="JTA">
<jta-data-source>jdbc/loja</jta-data-source>
<exclude-unlisted-classes>>false</exclude-unlisted-classes>
<properties/>
</persistence-unit>
</persistence>
```

Resultados:



Listagem de Produtos				
Novo Produto				
#	Nome	Quantidade	Preço de Venda	Opções
2	Teclado	50	R\$ 100,00	Alterar Excluir
3	Mouse	40	R\$ 80,00	Alterar Excluir



Dados do Produto

Nome:

Quantidade:

Preço de Venda:

[Incluir Produto](#) [Cancelar](#)

Dados do Produto

Nome:

Quantidade:

Preço de Venda:

Análise e Conclusão – Procedimento 2

1. Como o framework Bootstrap é utilizado?

O Bootstrap é incluído via CDN (ou arquivos estáticos) adicionando no `<head>` um `<link>` para o CSS e, normalmente antes de `</body>`, um `<script>` para o bundle JavaScript (que já contém o Popper). Uma vez incluído, basta aplicar classes pré-definidas aos elementos HTML—como `container`, `row`, `col`, `btn`, `table`, `form-control`, `mb-3`—para obter imediatamente estilos padronizados de layout, tipografia, cores e componentes interativos (botões, tabelas, formulários, modais etc.). Não há necessidade de escrever CSS personalizado para os componentes básicos; o framework provê todo o conjunto de regras.

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

O Bootstrap centraliza toda a estilização em classes CSS e regras de design system, sem impor marcações proprietárias ou tags especiais. Qualquer elemento HTML semântico (como `<div>`, `<form>`, `<table>`, `<button>`) ganha o estilo correto apenas ao receber as classes correspondentes. Isso separa completamente a estrutura (o HTML semântico e a hierarquia dos elementos) da apresentação (estilos e comportamentos), garantindo que você possa reorganizar ou alterar o HTML sem quebrar o design—desde que mantenha as classes do Bootstrap.

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

O Bootstrap foi desenhado mobile-first e traz um sistema de grid responsivo baseado em 12 colunas, com classes como `col-*-*` e utilitários de breakpoints (`sm`, `md`, `lg`, `xl`). Ao utilizar esse

grid, o layout se ajusta automaticamente a diferentes larguras de tela: colunas empilham-se em dispositivos menores e ficam lado a lado em telas maiores. Além disso, componentes como navbars, cards e formulários incorporam regras de responsividade, garantindo que o conteúdo se apresente de forma legível e usável em smartphones, tablets e desktops sem código extra.