



UNIVERSIDADE ESTÁCIO DE SÁ
PROGRAMAÇÃO FULL STACK
PÓLO NORTE SHOPPING

CLEYDSON ASSIS COELHO

**MISSÃO PRÁTICA 3 MUNDO 3
PROCEDIMENTO 2**

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.

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 />
```

```

<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>
  </persistence-unit>
</persistence>

```

```
<exclude-unlisted-classes>>false</exclude-unlisted-classes>
<properties/>
</persistence-unit>
</persistence>
```

ProdutoDados.jsp

```
<%@ 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 />

<button type="submit">
<%= (request.getParameter("acao").equals("formAlterar") ? "Salvar Alterações" : "Incluir Produto")
%>
</button>
<nbsp;<a href="produto?acao=listar">Cancelar</a>
</form>
</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>
<head>
    <meta charset="UTF-8">
    <title>Listagem de Produtos</title>
    <style>
        table { border-collapse: collapse; width: 100%; }
        th, td { border: 1px solid #ccc; padding: 8px; text-align: left; }
        th { background-color: #f0f0f0; }
        a { text-decoration: none; color: blue; }
    </style>
</head>
<body>
    <h1>Listagem de Produtos</h1>

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

    <%
        @SuppressWarnings("unchecked")
        List<Produto> produtos = (List<Produto>) request.getAttribute("produtos");
        if (produtos == null || produtos.isEmpty()) {
    %>
        <p><em>Nenhum produto encontrado.</em></p>
    <%
        } else {
    %>
    <table>
        <tr>
            <th>#</th>
            <th>Nome</th>
            <th>Quantidade</th>
            <th>Preço de Venda</th>
            <th>Opções</th>
        </tr>
        <%
            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() %>">Alterar</a>
                &nbsp;&nbsp;&nbsp;&|&nbsp;&nbsp;&nbsp;&
                <a href="produto?acao=excluir&id=<%= p.getIdProduto() %>"
                    onclick="return confirm('Confirma exclusão deste produto?');">
                    Excluir
            </td>
        </tr>
        <%
            }
    %>
    </table>
    %>
```

```

        </a>
    </td>
</tr>
<%
    }
%>
</table>
<%
    }
%>
</body>
</html>

```

Index.html

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Cadastro de Produtos</title>
    <!-- Redireciona automaticamente para o Front Controller -->
    <meta http-equiv="refresh" content="0; URL=produto?acao=listar">
</head>
<body>
    <!-- Em navegadores antigos ou se o meta não funcionar, exibe um link -->
    <p>Redirecionando para a lista de produtos...
    <a href="produto?acao=listar">Clique aqui se não for redirecionado</a>.</p>
</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
}

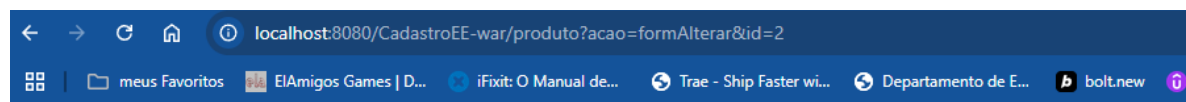
```

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



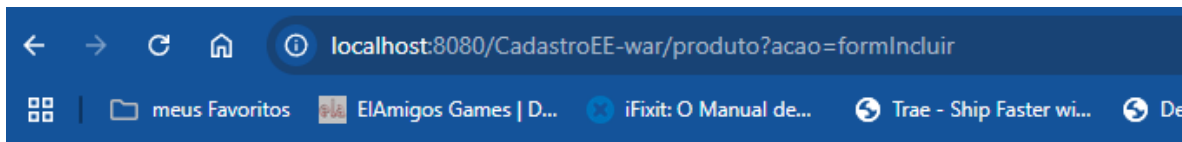
Alterar Produto

Nome:

Quantidade:

Preço de Venda:

[Cancelar](#)



Incluir Produto

Nome:

Quantidade:

Preço de Venda:

[Cancelar](#)

Análise e Conclusão – Procedimento 2

1. Padrão Front Controller em MVC Java

O Front Controller é um único ponto de entrada para todas as requisições da aplicação: em vez de cada página ou servlet ser chamada diretamente, todas as URLs são mapeadas para um servlet central (o controller). Nesse servlet — nosso ServletProdutoFC — lemos um parâmetro (por exemplo, acao), decidimos qual serviço de negócio (EJB/Fachada) invocar e, por fim, encaminhamos (forward) a requisição para o JSP de visualização adequado. Assim, o controller orquestra o fluxo:

Model: entidades JPA + EJBs

View: JSPs que exibem dados

Controller: servlet único que trata URL genérica e dispara ações via switch ou if/else

2. Diferenças e semelhanças entre Servlets e JSPs

Semelhanças:

Ambos rodam no contêiner de servlets (GlassFish) e atendem requisições HTTP.

Compartilham API de HttpServletRequest / HttpServletResponse e podem usar RequestDispatcher.

Podem ser configurados via anotações (@WebServlet) ou em web.xml.

Diferenças:

Servlet: classe Java pura. Ideal para lógica de controle e processamento (Controller). Exige programar a geração de HTML “na mão” (usando PrintWriter).

JSP: página HTML com marcações especiais (scriptlets, EL, taglibs). Foca na apresentação (View). Na primeira vez, o contêiner converte o JSP em servlet Java antes de compilar.

3. Redirect vs. forward e uso de parâmetros e atributos

Redirect (HTTP 302): o servidor responde ao cliente com código 302 e nova URL. O navegador faz uma segunda requisição para esse endereço, mudando a URL visível. É útil para evitar reenvio de formulários (Post/Redirect/Get) ou levar o usuário a outro contexto.

Forward (RequestDispatcher): o contêiner encaminha internamente para outro recurso sem envolver o cliente; a URL no navegador permanece a mesma. Permite compartilhar o mesmo objeto `HttpServletRequest`, inclusive seus atributos.

Parâmetros (`request.getParameter`) vêm diretamente do cliente — query string ou corpo de formulário — e carregam dados de entrada (filtros, campos de formulário, instruções de ação).

Atributos (`request.setAttribute`) são objetos Java colocados no escopo da requisição pelo servidor (controller) para repassar dados complexos (listas, entidades, mensagens) para o JSP durante um forward, sem aparecer na URL.