



Campus: POLO FSP - RO

Curso: Desenvolvimento Full Stack

Disciplina: Nível 4: Vamos integrar sistemas

Turma: POO-01

Semestre Letivo: 2025.1

Aluno: Eliton Rodrigues de Oliveira

Título da Prática

Desenvolvimento de uma aplicação corporativa Java EE com NetBeans, JPA, EJB e Servlets

Objetivo da Prática

Construir uma aplicação corporativa utilizando o NetBeans, empregando as tecnologias Java Persistence API (JPA), Enterprise Java Beans (EJB) e Servlets. O objetivo é integrar essas tecnologias para criar um sistema básico de cadastro e listagem de produtos, compreendendo como cada componente colabora no desenvolvimento de sistemas distribuídos para a plataforma Java EE.

Relatório de Prática

Título da Prática

Desenvolvimento de Interface Cadastral com Servlet e JSP utilizando o padrão Front Controller

Objetivo da Prática

Esta prática tem como objetivo implementar um sistema web baseado em Java EE utilizando os conceitos de Servlets, JSP, EJB, JPA e o padrão de arquitetura MVC. Utilizaremos o padrão **Front Controller** para centralizar o controle das ações do usuário, separando as responsabilidades entre controle, modelo e visualização. A aplicação permitirá operações de CRUD em produtos armazenados no banco de dados **SQL Server** via JPA.

Código-Fonte da Prática

Os arquivos principais utilizados nesta prática foram:

1. **Servlet Front Controller – ServletProdutoFC.java**

```
@WebServlet(name = "ServletProdutoFC", urlPatterns = {"/ServletProdutoFC"})

public class ServletProdutoFC extends HttpServlet {

    @EJB

    private ProdutoFacadeLocal facade;

    ...

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        String acao = request.getParameter("acao");

        String destino;

        if ("formIncluir".equals(acao) || "formAlterar".equals(acao)) {

            destino = "ProdutoDados.jsp";

            if ("formAlterar".equals(acao)) {

                int id = Integer.parseInt(request.getParameter("id"));

                Produto p = facade.find(id);

                request.setAttribute("produto", p);

            }

        } else {

            destino = "ProdutoLista.jsp";

            ...

        }

        RequestDispatcher rd = request.getRequestDispatcher(destino);

        rd.forward(request, response);

    }

}
```

2. Interface EJB – ProdutoFacadeLocal.java

@Local

```
public interface ProdutoFacadeLocal {  
  
    void create(Produto produto);  
  
    void edit(Produto produto);  
  
    void remove(Produto produto);  
  
    Produto find(Object id);  
  
    List<Produto> findAll();  
  
    ...  
}
```

3. Entidade – Produto.java

@Entity

@Table(name = "produto")

```
public class Produto implements Serializable {  
  
    @Id  
  
    private Integer idProduto;  
  
    private String nome;  
  
    private Integer quantidade;  
  
    private Float precoVenda;  
  
    ...  
}
```

4. Página JSP – ProdutoLista.jsp

<h1>Listagem de Produtos</h1>

Novo Produto

<table>

<tr><th>#</th><th>Nome</th><th>Quantidade</th><th>Preço</th><th>Opções</th></tr>

```

<c:forEach var="p" items="${produtos}">

<tr>

<td>${p.idProduto}</td>

<td>${p.nome}</td>

<td>${p.quantidade}</td>

<td>${p.precoVenda}</td>

<td>

<a href="ServletProdutoFC?acao=formAlterar&id=${p.idProduto}">Alterar</a>

<a href="ServletProdutoFC?acao=excluir&id=${p.idProduto}">Excluir</a>

</td>

</tr>

</c:forEach>

</table>

```

5. Página JSP – ProdutoDados.jsp

```

<h1>Dados do Produto</h1>

<form action="ServletProdutoFC" method="post">

<input type="hidden" name="acao" value="${produto != null ? 'alterar' : 'incluir'}" />

<input type="hidden" name="id" value="${produto.idProduto}" />

Nome: <input type="text" name="nome" value="${produto.nome}" />

Quantidade: <input type="number" name="quantidade" value="${produto.quantidade}" />

Preço: <input type="text" name="precoVenda" value="${produto.precoVenda}" />

<input type="submit" value="${produto != null ? 'Alterar Produto' : 'Adicionar Produto'}" />

</form>

```

Análise e Conclusão

✅ Como funciona o padrão Front Controller, e como ele é implementado?

O **Front Controller** centraliza todas as requisições do cliente em um único servlet. No projeto, o ServletProdutoFC atua como ponto único de entrada, interpretando o parâmetro acao e encaminhando para o JSP ou EJB adequado. Isso facilita a manutenção e organização da lógica de controle.

✓ Diferenças e semelhanças entre Servlets e JSPs

- **Servlets** são classes Java que processam requisições HTTP e controlam o fluxo da aplicação.
- **JSPs** são páginas de visualização com HTML e comandos Java embutidos.
- Ambos são parte da especificação Java EE, mas **servlets são para controle** e **JSPs para visualização**.

✓ Diferença entre redirecionamento e forward com RequestDispatcher

- **forward (RequestDispatcher)**: envia o controle internamente para outra página sem mudar a URL do navegador.
- **sendRedirect**: envia uma resposta ao cliente pedindo uma nova requisição, mudando a URL.
- O forward é mais eficiente dentro da mesma aplicação.

✓ Para que servem parâmetros e atributos no HttpServletRequest?

- **Parâmetros** (getParameter) carregam dados enviados pelo cliente, como dados de formulários.
- **Atributos** (setAttribute) armazenam objetos para compartilhar dados entre Servlets e JSPs, como a lista de produtos ou a entidade editada.

Repositório GIT do Projeto

🔗 **GitHub:** <https://github.com/Elitonr65/CadastroEE>