



POLO OLARIA - RIO DE JANEIRO - RJ

DESENVOLVIMENTO FULL STACK

Nível 4: Vamos Integrar Sistemas | Turma 9001

Erik Bastos de Moraes

Missão Prática | Nível 4 | Mundo 3

2º Procedimento | Interface Cadastral com Servlet e JSPs

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.
- 5- 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

```
package cadastroee.servlets;

import cadastroee.model.Produto;
import cadastroee.controller.ProdutoFacadeLocal;
import jakarta.ejb.EJB;
import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.util.List;

@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 = "ProdutoLista.jsp";

        if (null == acao) {
            List<Produto> produtos = facade.findAll();
            request.setAttribute("lista", produtos);
```

```

} else switch (acao) {
    case "listar" -> {
        List<Produto> produtos = facade.findAll();
        request.setAttribute("lista", produtos);
    }
    case "formIncluir" -> destino = "ProdutoDados.jsp";
    case "formAlterar" -> {
        int id = Integer.parseInt(request.getParameter("id"));
        Produto p = facade.find(id);
        request.setAttribute("produto", p);
        destino = "ProdutoDados.jsp";
    }
    case "excluir" -> {
        int id = Integer.parseInt(request.getParameter("id"));
        Produto p = facade.find(id);
        facade.remove(p);
        request.setAttribute("lista", facade.findAll());
    }
    case "alterar" -> {
        int id = Integer.parseInt(request.getParameter("id"));
        Produto p = facade.find(id);
        p.setNome(request.getParameter("nome"));
        p.setQuantidade(Integer.valueOf(request.getParameter("quantidade")));
        p.setPrecoVenda((float)
Double.parseDouble(request.getParameter("precoVenda")));
        facade.edit(p);
        request.setAttribute("lista", facade.findAll());
    }
    case "incluir" -> {
        Produto p = new Produto();

```

```

        p.setNome(request.getParameter("nome"));
        p.setQuantidade(Integer.valueOf(request.getParameter("quantidade")));
        p.setPrecoVenda((float)
Double.parseDouble(request.getParameter("precoVenda")));
        facade.create(p);
        request.setAttribute("lista", facade.findAll());
    }
    default -> {
    }
}

```

```

RequestDispatcher rd = request.getRequestDispatcher(destino);
rd.forward(request, response);
}

```

@Override

```

protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

```

@Override

```

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}
}

```

ProdutoLista.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.util.*, cadastroee.model.Produto"%>
<!DOCTYPE html>
<html>
<head>
    <title>Lista de Produtos</title>
</head>
<body>
    <h1>Lista de Produtos</h1>

    <a href="ServletProdutoFC?acao=formIncluir">Novo Produto</a>

    <table border="1">
        <tr>
            <th>ID</th><th>Nome</th><th>Quantidade</th><th>Preço</th>
        <th>Ações</th>
        </tr>
        <%
            List<Produto> lista = (List<Produto>) request.getAttribute("lista");
            if (lista != null) {
                for (Produto p : lista) {
        %>
        <tr>
            <td><%= p.getIdProduto() %></td>
            <td><%= p.getNome() %></td>
            <td><%= p.getQuantidade() %></td>
            <td><%= p.getPrecoVenda() %></td>
```

<td>

 <a href="ServletProdutoFC?acao=formAlterar&id=<%= p.getIdProduto()
%>">Alterar

 <a href="ServletProdutoFC?acao=excluir&id=<%= p.getIdProduto()
%>">Excluir

</td>

</tr>

<%

 }

}

%>

</table>

</body>

</html>

ProdutoDados.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="cadastroee.model.Produto"%>
<%
    Produto produto = (Produto) request.getAttribute("produto");
    String acao = (produto == null) ? "incluir" : "alterar";
%>
<!DOCTYPE html>
<html>
<head>
    <title>Dados do Produto</title>
</head>
<body>
    <h1><%= (produto == null ? "Novo Produto" : "Alterar Produto") %></h1>

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

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

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

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

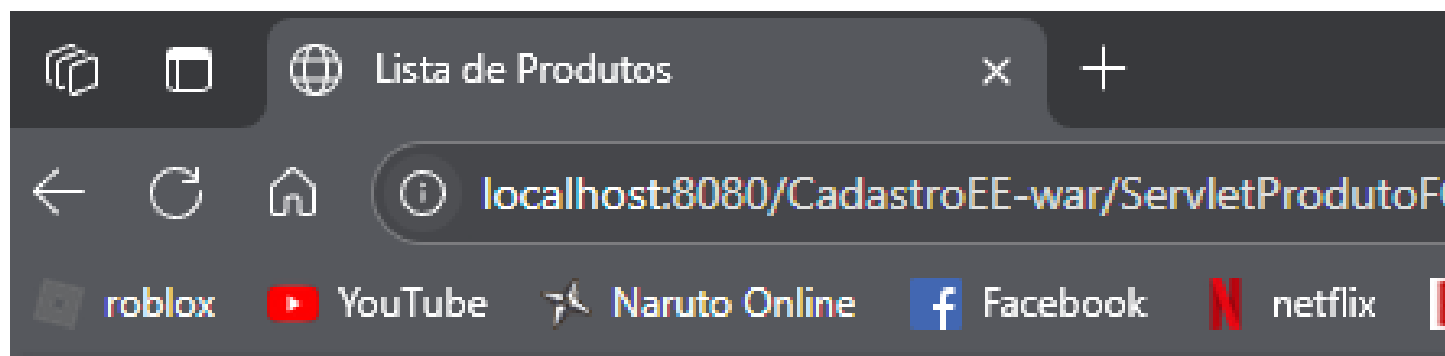
        <input type="submit" value="<%= (produto == null ? "Incluir" : "Alterar") %>" />
    </form>
```

[Voltar](ServletProdutoFC?acao=listar)

</body>

</html>

Resultado



Lista de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço	Ações
1	Mouse	100	50.0	Alterar Excluir
2	Teclado	200	80.0	Alterar Excluir
4	Mouse	27	20.0	Alterar Excluir
5	Monitor	1	1000.0	Alterar Excluir

Análise e Conclusão

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

R: O Front Controller centraliza a entrada de requisições em um único ponto, o controlador frontal, que encaminha para diferentes componentes da aplicação. Ele funciona como um centro de requisições, recebendo todas as solicitações, analisando e direcionando para os controladores específicos

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

R: Servlets são classes java gerenciam requisições e respostas HTTP, enquanto JSPs são apenas paginas web com java incorporado, ambos são usados para gerar conteúdo dinâmico em aplicações web, mas Servlets são muito mais complexos e flexíveis.

C. 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?

R: O redirecionamento simples, apenas redireciona o usuário para outra pagina web, enquanto o método forward encaminha dados para outro JSP ou Servlet, o redirecionamento é pelo lado do cliente, enquanto o método forward pelo servidor. Parâmetros são normalmente usados para armazenar informações, são dados enviados pelo cliente, já Atributos armazenam dados que não foram enviados pelo cliente.