



Desenvolvimento Full Stack

Aluno: Victor de A. Costa

Missão Prática | Nível 4 | Mundo 3 - 2024.3
VAMOS MANTER AS INFORMAÇÕES!

Interface Cadastral com Servlet e JSPs

2º Procedimento | Alimentando a Base

--> CÓDIGOS UTILIZADOS

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>
    <servlet-name>ServletProduto</servlet-name>
    <servlet-class>cadastroee.servlets.ServletProduto</servlet-class>
  </servlet>

  <servlet>
    <servlet-name>ServletProdutoFC</servlet-name>
    <servlet-class>cadastroee.servlets.ServletProdutoFC</servlet-class>
  </servlet>

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

  <servlet-mapping>
    <servlet-name>ServletProduto</servlet-name>
    <url-pattern>/ServletProduto</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>ServletProdutoFC</servlet-name>
    <url-pattern>/ServletProdutoFC</url-pattern>
  </servlet-mapping>
</web-app>
```

ProdutoDados.jsp

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<html lang="pt-BR">
```

```

<head>
    <meta charset="UTF-8">
    <title>Cadastro de Produto</title>
    <style>
        label {
            display: block;
            margin-bottom: 10px;
        }
        input[type="text"], input[type="number"] {
            width: 100%;
            height: 30px;
            margin-bottom: 20px;
            padding: 10px;
            box-sizing: border-box;
        }
        .btn {
            background-color: #4CAF50;
            color: white;
            padding: 10px 20px;
            text-decoration: none;
            cursor: pointer;
        }
        .btn:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <h1>Cadastro de Produto</h1>

    <form action="${pageContext.request.contextPath}/ServletProdutoFC" method="post">
        <input type="hidden" name="acao" value="${acao}">
        <input type="hidden" name="redirigir" value="true">
        <c:if test="${acao == 'alterar'}">
            <input type="hidden" name="id" value="${produto.idProduto}">
        </c:if>
        <label for="nome">Nome:</label>
        <input type="text" id="nome" name="nome" value="${nomeproduto}">

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

        <label for="precoVenda">Preço de Venda:</label>
        <input type="number" id="precoVenda" name="precoVenda" value="${produto.precoVenda}">

        <c:if test="${acao == 'incluir'}">
            <input type="submit" value="Cadastrar" class="btn">
        </c:if>
        <c:if test="${acao == 'alterar'}">
            <input type="submit" value="Alterar" class="btn">
        </c:if>
    </form>
</body>
</html>

```

ProdutoLista.jsp

```

<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Listagem de Produtos</title>
    <style>
        table {
            border-collapse: collapse;
            width: 100%;
        }
        th, td {
            border: 1px solid black;
            padding: 8px;
            text-align: center;
        }
    </style>

```

```

.btn {
    background-color: #4CAF50;
    color: white;
    padding: 10px 20px;
    text-decoration: none;
    cursor: pointer;
}
.btn:hover {
    background-color: #45a049;
}

```

</style>

</head>

<body>

<h1>Listagem de Produtos</h1>

Novo Produto

<table>

ID	Nome	Quantidade	Preço de Venda	Opções		
<c:foreach ><="" items="\${produtos}" td="" var="produto"> <td><c:out ><="" td="" value="\${produto.idProduto}"> <td><c:out ><="" td="" value="\${produto.nome}"> <td><c:out ><="" td="" value="\${produto.quantidade}"> <td><c:out ><="" td="" value="R\$ \${produto.precoVenda}"> <td><c:out ><="" a="" value="alterar&id=\${produto.idProduto}"></c:out></td> <td><c:out ><="" a="" value="excluir&id=\${produto.idProduto}"></c:out></td> </c:out></td></c:out></td></c:out></td></c:out></td></c:foreach>	<c:out ><="" td="" value="\${produto.idProduto}"> <td><c:out ><="" td="" value="\${produto.nome}"> <td><c:out ><="" td="" value="\${produto.quantidade}"> <td><c:out ><="" td="" value="R\$ \${produto.precoVenda}"> <td><c:out ><="" a="" value="alterar&id=\${produto.idProduto}"></c:out></td> <td><c:out ><="" a="" value="excluir&id=\${produto.idProduto}"></c:out></td> </c:out></td></c:out></td></c:out></td></c:out>	<c:out ><="" td="" value="\${produto.nome}"> <td><c:out ><="" td="" value="\${produto.quantidade}"> <td><c:out ><="" td="" value="R\$ \${produto.precoVenda}"> <td><c:out ><="" a="" value="alterar&id=\${produto.idProduto}"></c:out></td> <td><c:out ><="" a="" value="excluir&id=\${produto.idProduto}"></c:out></td> </c:out></td></c:out></td></c:out>	<c:out ><="" td="" value="\${produto.quantidade}"> <td><c:out ><="" td="" value="R\$ \${produto.precoVenda}"> <td><c:out ><="" a="" value="alterar&id=\${produto.idProduto}"></c:out></td> <td><c:out ><="" a="" value="excluir&id=\${produto.idProduto}"></c:out></td> </c:out></td></c:out>	<c:out ><="" td="" value="R\$ \${produto.precoVenda}"> <td><c:out ><="" a="" value="alterar&id=\${produto.idProduto}"></c:out></td> <td><c:out ><="" a="" value="excluir&id=\${produto.idProduto}"></c:out></td> </c:out>	<c:out ><="" a="" value="alterar&id=\${produto.idProduto}"></c:out>	<c:out ><="" a="" value="excluir&id=\${produto.idProduto}"></c:out>
					Alterar	<c:out ><="" c:out="" style="color: red;" value="Excluir"></c:out>

</c:forEach>

</table>

</body>

</html>

ServletProdutoFC.java

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template
 */
package cadastroee.servlets;

```

```

import java.io.IOException;
import jakarta.ejb.EJB;
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import cadastroee.controller.ProdutoFacadeLocal;
import cadastroee.model.Produto;
import java.util.List;

```

```

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 = "";

switch (acao) {
    case "listar":
        destino = "ProdutoLista.jsp";
        break;
    case "incluir":
    case "alterar":
        destino = "ProdutoDados.jsp";
        break;
    default:
        // Caso não seja uma ação válida, redireciona para listar
        destino = "ProdutoLista.jsp";
}

try {
    switch (acao) {
        case "listar":
            listarResultados(request);
            break;
        case "incluir":
            incluirProduto(request, acao);
            if (request.getParameter("redirigir") != null) {
                destino = "ServletProdutoFC?acao=listar";
            }
            break;
        case "excluir":
            excluirProduto(request);
            break;
        case "alterar":
            alterarProduto(request, acao);
            if (request.getParameter("redirigir") != null) {
                destino = "ServletProdutoFC?acao=listar";
            }
            break;
    }
} catch (Exception e) {
    request.setAttribute("mensagemErro", "Ocorreu um erro: " + e.getMessage());
}

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

private void listarResultados(HttpServletRequest request) {
    List<Produto> produtos = facade.findAll();
    System.out.println("N produtos: " + produtos.size());
    request.setAttribute("produtos", produtos);
}

private void incluirProduto(HttpServletRequest request, String acao) {
    Produto produto = new Produto();
    request.setAttribute("acao", acao);
    produto.setNome(request.getParameter("nome"));
    produto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));
    produto.setPrecoVenda(Float.parseFloat(request.getParameter("precoVenda")));
    facade.create(produto);
}

private void alterarProduto(HttpServletRequest request, String acao) {
    int id = Integer.parseInt(request.getParameter("id"));
    request.setAttribute("acao", acao);
    Produto produto = facade.find(id);
    request.setAttribute("nomedoproduto", produto.nome);
    request.setAttribute("produto", produto);
    produto.setNome(request.getParameter("nome"));
    produto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));
    produto.setPrecoVenda(Float.parseFloat(request.getParameter("precoVenda")));
    facade.edit(produto);
}

private void excluirProduto(HttpServletRequest request) {
    int id = Integer.parseInt(request.getParameter("id"));
    Produto produto = facade.find(id);
    facade.remove(produto);
    listarResultados(request);
}

```

```

@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);
}
}

```

ServletProduto.java

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template
 */
package cadastroee.servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.List;
import jakarta.ejb.EJB;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import cadastroee.controller.ProdutoFacadeLocal;
import cadastroee.model.Produto;

/**
 *
 * @author victorcosta
 */
public class ServletProduto extends HttpServlet {

    @EJB
    ProdutoFacadeLocal facade;

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet Produto</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Lista de Produtos</h1>");
            out.println("<ul>");

            // Utilizar o facade para recuperar os dados
            List<Produto> produtos = facade.findAll();

            // Apresentar os dados em uma lista HTML
            for (Produto produto : produtos) {
                out.println("<li>" + produto.getNome() + "</li>");
            }
            out.println("</ul>");
            out.println("</body>");
            out.println("</html>");
        }
    }
}

```

```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
}// </editor-fold>
}

```

---> RESULTADO

The screenshot displays two windows side-by-side. On the left is a web browser window titled 'Listagem de Produtos' showing a table of products with columns: ID, Nome, Quantidade, Preço de Venda, and Opções. The table contains 13 rows of data. On the right is a SQL Server Management Studio (SSMS) window titled 'SQLQuery1.sql - GW...SS55.master (sa (67))'. It shows a query that selects top 1000 rows from the 'Produto' table in the 'Loja2.dbo' schema, returning columns: idProduto, nome, quantidade, and precoVenda.

ID	Nome	Quantidade	Preço de Venda	Opções
1	nome produto	50	R\$ 5000.0	Alterar Excluir
2	nome produto	50	R\$ 5000.0	Alterar Excluir
3	Camiseta Azul	564	R\$ 7000.0	Alterar Excluir
4	Calça jeans	15	R\$ 400.0	Alterar Excluir
8	asdadas	2423	R\$ 32432.0	Alterar Excluir
9	sadasd	242	R\$ 4314.0	Alterar Excluir
10	edit produto	999	R\$ 99999.0	Alterar Excluir
11	sdadas	2	R\$ 2.0	Alterar Excluir
12	2132	213	R\$ 12321.0	Alterar Excluir
13	sqdas	2	R\$ 134.0	Alterar Excluir

```

SELECT TOP (1000) [idProduto]
      ,[nome]
      ,[quantidade]
      ,[precoVenda]
  FROM [Loja2].[dbo].[Produto]

```

---> ANÁLISE E CONCLUSÃO

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

O padrão Front Controller é um padrão de design usado em arquiteturas web para centralizar o controle das requisições do usuário. Em vez de cada página ou recurso ter seu próprio controlador, um único controlador, é responsável por gerenciar todas as requisições. Isso simplifica a estrutura da aplicação, melhora a reutilização de código e facilita a manutenção. Na arquitetura MVC (Model-View-Controller), o padrão Front Controller é frequentemente implementado usando Servlets.

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

Servlets: São classes Java que tratam requisições e respostas HTTP diretamente, geralmente usadas para controlar a lógica de negócios e o fluxo da aplicação.

JSPs (JavaServer Pages): São páginas que permitem a mistura de código Java com HTML para criar conteúdo dinâmico. São mais focadas na apresentação e na geração de respostas HTML para o cliente.

Semelhanças: Ambos são componentes da tecnologia Java EE e são executados em servidores de aplicação compatíveis com Java EE. Tanto Servlets quanto JSPs tratam requisições e respostas HTTP, permitindo a criação de aplicações web dinâmicas e Servlets e JSPs podem ser usados em conjunto.

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 HttpServletRequest?

Redirecionamento Simples: Utiliza o método sendRedirect do objeto HttpServletResponse para enviar uma resposta ao cliente (navegador) instruindo-o a fazer uma nova requisição para uma URL diferente.

Forward com RequestDispatcher: Utiliza o método forward do RequestDispatcher para encaminhar a requisição e a resposta do cliente para outro recurso (servlet, JSP, etc.) no servidor.

Os parâmetros são dados enviados pelo cliente (navegador) para o servidor como parte da requisição HTTP. Eles podem vir na URL (para métodos GET) ou no corpo da requisição (para métodos POST).

Já, os Atributos são objetos que você pode adicionar à requisição HTTP para compartilhar dados entre componentes do servidor (servlets, JSPs, etc.) durante o processamento da mesma requisição.