



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

3º Procedimento | Melhorando o Design da Interface

--> 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>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNllyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
    <meta charset="UTF-8">
    <title>Cadastro de Produto</title>
    <style>
    </style>
</head>
<body class="container">
    <h1>Dados do Produto</h1>

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

        <div class="mb-3">
            <label for="nome" class="form-label">Nome:</label>
            <input type="text" id="nome" name="nome" value="${nomeproduto}" class="form-control">
        </div>

        <div class="mb-3">
            <label for="quantidade" class="form-label">Quantidade:</label>
            <input type="number" id="quantidade" name="quantidade" value="${produto.quantidade}" class="form-control">
        </div>

        <div class="mb-3">
            <label for="precoVenda" class="form-label">Preço de Venda:</label>
            <input type="number" id="precoVenda" name="precoVenda" value="${produto.precoVenda}" class="form-control">
        </div>
        <c:if test="${acao == 'incluir'}">
            <input type="submit" value="Adicionar Produto" class="btn btn-primary">
        </c:if>
        <c:if test="${acao == 'alterar'}">
            <input type="submit" value="Alterar Produto" class="btn btn-primary">
        </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>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNllyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
    <meta charset="UTF-8">
    <title>Listagem de Produtos</title>
    <style>
    </style>
</head>
<body class="container">
    <h1>Listagem de Produtos</h1>
    <a href="${pageContext.request.contextPath}/ServletProdutoFC?acao=incluir"
        class="btn btn-primary m-2">Novo Produto</a>
    <table class="table table-striped">
        <tr class="table-dark">
            <th>#</th>
            <th>Nome</th>
            <th>Quantidade</th>
            <th>Preço de Venda</th>
            <th>Opções</th>
        </tr>

```

```

<c:forEach var="produto" items="${produtos}">
    <tr>
        <td>${produto.idProduto}</td>
        <td>${produto.nome}</td>
        <td>${produto.quantidade}</td>
        <td>R$ ${produto.precoVenda}</td>
        <td>
            <a href="#">\${pageContext.request.contextPath}/ServletProdutoFC?acao=alterar&id=\${produto.idProduto}" class="btn btn-primary btn-sm">Alterar
            <a href="#">\${pageContext.request.contextPath}/ServletProdutoFC?acao=excluir&id=\${produto.idProduto}" onclick="return confirm\('Tem certeza que deseja excluir esse produto?'\)" class="btn btn-danger btn-sm">Excluir
        </td>
    </tr>
</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);
            destino = "ServletProdutoFC?acao=listar";
            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("nomeproduto", 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

Listagem de Produtos

[Novo Produto](#)

#	Nome	Quantidade	Preço de Venda	Opções
1	nome produto	50	R\$ 5000.0	Alterar Excluir
2	213213	50	R\$ 5000.0	Alterar Excluir
3	Camiseta Azul	564	R\$ 7000.0	Alterar Excluir
4	Calça jeans	15	R\$ 400.0	Alterar Excluir
9	sadasd	242	R\$ 4314.0	Alterar Excluir
10	aaaaaaaaaa	999	R\$ 99999.0	Alterar Excluir
12	2132	213	R\$ 12321.0	Alterar Excluir
14	sqdas	2	R\$ 134.0	Alterar Excluir
15	adsdas	43	R\$ 32423.0	Alterar Excluir
16	adsdas	43	R\$ 32423.0	Alterar Excluir

Dados do Produto

Nome:

Quantidade:

Preço de Venda:

[Adicionar Produto](#)

---> ANÁLISE E CONCLUSÃO

A. Como o framework Bootstrap é utilizado?

O Bootstrap é um dos frameworks mais populares para desenvolvimento front-end, utilizado para criar sites e aplicações web responsivos e visualmente atraentes. Ele facilita o trabalho com HTML, CSS e JavaScript, oferecendo uma grande variedade de componentes prontos e sistemas de layout baseados em grids.

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

O Bootstrap é uma excelente ferramenta para criar sites responsivos e consistentes sem modificar a estrutura básica do HTML. Aqui estão alguns motivos pelos quais o Bootstrap garante a independência estrutural do HTML: Utilização de Classes CSS, Sistema de Grid Flexível, Componentes Reutilizáveis, Compatibilidade e Responsividade Integradas.

C. Qual a relação entre o Boostrap e a responsividade da página?

O Bootstrap é conhecido por sua capacidade de tornar as páginas web responsivas, ou seja, adaptáveis a diferentes tamanhos de tela, como desktops, tablets e celulares, através de ferramentas como: Sistema de Grid Flexível, Classes de Utilidade, Media Queries, Componentes Reutilizáveis e Imagens responsivas.