

Aluno: Altair Silva Filho Matrícula: 202403254409 Unidade: Palhoça – SC

Curso: Desenvolvimento Full Stack Disciplina: Vamos integrar sistemas

Período: 2025.1

Relatório da Missão Prática - Nível 4 - Mundo 3

Vamos integrar sistemas

2º Procedimento | Interface Cadastral com Servlet e JSPs

Objetivo

Implementar um sistema cadastral com interface Web, baseado nas tecnologias de Servlets, JPA e JEE.

Neste procedimento, foram criados os Servlets e as páginas JSP's (Java Server Pages)

Códigos do Projeto

Arquivos Servlets e JSP's

ServletProdutoFC.java

package cadastroee.servlets;

import cadastroee.controller.ProdutoFacadeLocal;

import cadastroee.model.Produto;

import jakarta.ejb.EJB;

import jakarta.servlet.RequestDispatcher;

import java.io.IOException;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name = "ServletProdutoFC", urlPatterns = {"/ServletProdutoFC"})
public class ServletProdutoFC extends HttpServlet {

@EJB

private ProdutoFacadeLocal facade;

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

```
throws ServletException, IOException {
request.setCharacterEncoding("UTF-8");
response.setContentType("text/html;charset=UTF-8");
String acao = request.getParameter("acao");
String destino;
if ("formIncluir".equals(acao) || "formAlterar".equals(acao)) {
  destino = "ProdutoDados.jsp";
  if ("formAlterar".equals(acao)) {
   try {
     int id = Integer.parseInt(request.getParameter("idProduto"));
     Produto produto = facade.find(id);
     request.setAttribute("produto", produto);
   } catch (NumberFormatException e) {
     request.setAttribute("erro", "ID inválido para alteração");
   }
 }
} else {
  destino = "ProdutoLista.jsp";
 if (acao == null) {
   acao = "listar"; // ação padrão
 }
 try {
   switch (acao) {
     case "listar":
       request.setAttribute("produtos", facade.findAll());
       break;
     case "excluir":
       int idExclusao = Integer.parseInt(request.getParameter("idProduto"));
       Produto prodExclusao = facade.find(idExclusao);
       if (prodExclusao != null) {
         facade.remove(prodExclusao);
       request.setAttribute("produtos", facade.findAll());
       request.setAttribute("mensagem", "Produto excluído com sucesso!");
       break;
     case "alterar":
       int idAlt = Integer.parseInt(request.getParameter("idProduto"));
       Produto prodAlt = facade.find(idAlt);
       if (prodAlt != null) {
         prodAlt.setNome(request.getParameter("nome"));
         prodAlt.setQuantidade(Integer.valueOf(request.getParameter("quantidade")));
         prodAlt.setPrecoVenda(Integer.valueOf(request.getParameter("preco")));
         facade.edit(prodAlt);
```

```
request.setAttribute("produtos", facade.findAll());
           request.setAttribute("mensagem", "Produto alterado com sucesso!");
          break;
         case "incluir":
           Produto novoProd = new Produto();
           novoProd.setIdProduto(facade.getProximoIdProduto());
           novoProd.setNome(request.getParameter("nome"));
           novoProd.setQuantidade(Integer.valueOf(request.getParameter("quantidade")));
           novoProd.setPrecoVenda(Integer.valueOf(request.getParameter("preco")));
          facade.create(novoProd);
           request.setAttribute("produtos", facade.findAll());
           request.setAttribute("mensagem", "Produto incluído com sucesso!");
           break;
         default:
           request.setAttribute("produtos", facade.findAll());
           break;
       }
     } catch (Exception e) {
       request.setAttribute("erro", "Erro ao processar a ação: " + e.getMessage());
     }
   }
   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"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
 <head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>JSP Page</title>
 </head>
 <body>
   <h2>Listagem de Produtos</h2>
   <!-- Link para o formulário de inclusão -->
   >
    <a href="?acao=formIncluir">Novo Produto</a>
   <!-- Dados -->
   ID
    Nome
    Quantidade
    Preço de Venda
    Opções
   <c:forEach var="p" items="${produtos}">
    ${p.idProduto}
      ${p.nome}
      ${p.quantidade}
      R$ ${p.precoVenda}
      <a href="?acao=formAlterar&idProduto=${p.idProduto}">Alterar</a> |
        <a href="?acao=excluir&idProduto=${p.idProduto}" onclick="return confirm('Deseja excluir
este produto?')">Excluir</a>
      </c:forEach>
 </body>
</html>
ProdutoDados.jsp
<@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
 <head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <title>JSP Page</title>
 </head>
 <body>
   <h2>Dados do Produto</h2>
   <form action="ServletProdutoFC" method="post">
    <c:set var="acao" value="${produto != null ? 'alterar' : 'incluir'}" />
    <input type="hidden" name="acao" value="${acao}" />
```

```
<c:if test ="${produto != null }">
       <input type="hidden" name="idProduto" value="${produto.idProduto}" />
       </c:if>
       <label for ="nome">Nome:</label><br/>
       <input type="text" name="nome" id="nome" value="${produto.nome}" required /><br/>
       <label for="quantidade">Quantidade:</label><br/>
       <input type="number" name="quantidade" id="quantidade" value="${produto.quantidade}"
required /><br/><br/>
       <label for="preco">Preço de Venda:</label><br/>
       <input type="number" name="preco" id="preco" value="${produto.precoVenda}" required
/><br/>
       <!-- Botão -->
       <input type="submit" value="${produto!= null? 'Salvar Produto': 'Cadastrar Produto'}" />
   </form>
 </body>
</html>
```

Resultados da execução dos códigos

As janelas com os resultados dos procedimentos e execuções do código são mostradas a seguir:

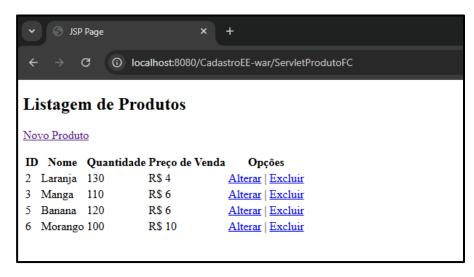


Figura 1 – Página com a lista de produtos - ServletProdutoFC



Figura 2 – Formulário para inclusão de produtos



Figura 3 – Resultado da inclusão de novo produto

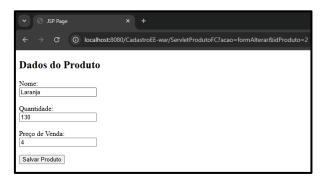


Figura 4 – Formulário para alteração



Figura 5 – Resultado da alteração



Figura 6 – Formulário para exclusão



Figura 7 – Resultado da exclusão

Análise e Conclusão

Este procedimento mostrou como usar o padrão *Front Controller* em aplicativos Web com o uso do DAO (Data Access Object). Nesse padrão todas as requisições serão recebidas em apenas um Servlet, onde ocorrerão as conversões e os processamentos necessários, com o redirecionamento para a visualização correta ao final. No projeto, o padrão foi implementado através do ServletProdutoFC mapeado para uma URL padrão, que, através do parâmetro "acao", cuidou de todas as requisições.

Para mapear as camadas da arquitetura MVC (Model-View-Controller), o projeto usou os componentes Servlets e JSP's, fundamentais para a criação de aplicações Web dinâmicas. Podemos dizer que ambos trabalham em conjunto. Enquanto os Servlets geram os recursos em resposta a uma solicitação ao servidor, os JSP's tornam mais fácil o desenvolvimento dos Servlets. Assim, o primeiro cuida da lógica de negócios, e o segundo é usado para a apresentação visual.

A missão mostrou também a diferença entre os dois métodos de redirecionamento do Java para Web. No método simples, a interface envia um sinal de redirecionamento para o navegador, gerando uma nova requisição. O projeto usou a interface RequestDispatcher, que faz parte do método *Forward*, que envia um sinal de redirecionamento ao nível do servidor, que ocorre de forma interna, com a manutenção da requisição original e suas informações. Isto se deu na realização de uma operação (incluir, excluir e alterar) evitando o reenvio do formulário ao recarregar a página. Deste modo, temos um processamento mais rápido.

No projeto vimos também, que para haver a transmissão de dados durante os ciclos de requisição, as aplicações Web em Java fazem uso de parâmetros e atributos nos objetos HttpRequest. Os parâmetros se referem aos valores enviados pelo navegador ou cliente para o servidor como nas requisições GET e POST. Os atributos são informações adicionais associadas à requisição e são usados especialmente para a troca de dados entre os Servlets e JSP's.

Armazenado no Github: https://github.com/altairsf/CadastroEE.git