

Faculdade Estácio - Polo Curitiba - Centro

Curso: Desenvolvimento Full Stack

Disciplina: Vamos manter as informações

Número da Turma: RPG0017

Semestre Letivo: 3

Integrante: Mariana Lucas Fernandes Onório

Repositório: <https://github.com/MariLFO/estacio-mundo3-missao-nivel-4>

Sumário:

Faculdade Estácio - Polo Curitiba - Centro	1
Sumário:	1
1. Título da Prática:	3
2. Objetivos da Prática:	3
3. Códigos do roteiro:	3
Arquivo: servlets/ServletProdutoFC.java	3
Arquivo: web/ProdutoLista.jsp	6
Arquivo: web/ProdutoDados.jsp	7
4. Resultados da execução dos códigos	8
5. Análise e Conclusão	9
a) Como funciona o padrão Front Controller, e como ele é implementado em um aplicativo Web Java, na arquitetura MVC?	9
b) Quais as diferenças e semelhanças entre Servlets e JSPs?	9
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?	9

1. Título da Prática:

RPG0017 - Vamos integrar sistemas

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

2. 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.

3. Códigos do roteiro:

Arquivo: [servlets/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 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.model.Produto;
import cadastroee.controller.ProdutoFacadeLocal;
import java.io.PrintWriter;

/**
 *
 * @author Mari
 */
public class ServletProdutoFC extends HttpServlet {
    @EJB
    private ProdutoFacadeLocal facade;

    /**
     * Processes requests for both HTTP GET and POST

```

```

* 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 {
    String acao = request.getParameter("acao");
    if (acao == null) {
        acao = "listar";
    }

    Integer idProduto;
    Produto produto;
    String destino = "ProdutoLista.jsp";

    switch (acao) {
        case "listar":
            List<Produto> listaProdutos = facade.findAll();
            request.setAttribute("listaProdutos", listaProdutos);
            break;
        case "excluir":
            idProduto = Integer.parseInt(request.getParameter("idProduto"));
            produto = facade.find(idProduto);
            facade.remove(produto);
            listaProdutos = facade.findAll();
            request.setAttribute("listaProdutos", listaProdutos);
            break;
        case "alterar":
            idProduto = Integer.parseInt(request.getParameter("idProduto"));
            produto = facade.find(idProduto);
            produto.setNome(request.getParameter("nome"));
            produto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));
            produto.setPrecoVenda(Float.parseFloat(request.getParameter("precoVenda")));
            facade.edit(produto);
            listaProdutos = facade.findAll();
            request.setAttribute("listaProdutos", listaProdutos);
            break;
        case "incluir":
            produto = new Produto();
            produto.setNome(request.getParameter("nome"));
            produto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));
            produto.setPrecoVenda(Float.parseFloat(request.getParameter("precoVenda")));
            facade.create(produto);
            listaProdutos = facade.findAll();
            request.setAttribute("listaProdutos", listaProdutos);
            break;
        case "formIncluir":
            destino = "ProdutoDados.jsp";
            break;
        case "formAlterar":
            idProduto = Integer.parseInt(request.getParameter("idProduto"));
            produto = facade.find(idProduto);
    }
}

```

```

        request.setAttribute("produto", produto);
        destino = "ProdutoDados.jsp";
        break;
    }
    request.getRequestDispatcher(destino).forward(request, response);
}

// <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>
}

```

Arquivo: [web/ProdutoLista.jsp](#)

```
<%--
Document    : ProdutoLista
Created on  : Aug 21, 2023, 4:33:08 PM
Author      : Mari
--%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Listagem de Produtos</title>
  </head>
  <body>
    <h1>Listagem de Produtos</h1>

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

    <table>
      <thead>
        <tr>
          <th>#</th>
          <th>Nome</th>
          <th>Quantidade</th>
          <th>Preço de Venda</th>
          <th>Opções</th>
        </tr>
      </thead>

      <tbody>
        <c:forEach items="${listaProdutos}" var="produto">
          <tr>
            <td>${produto.idProduto}</td>
            <td>${produto.nome}</td>
            <td>${produto.quantidade}</td>
            <td>${produto.precoVenda}</td>

            <!-- Links para alteração e exclusão -->
            <td><a href="ServletProdutoFC?acao=formAlterar&idProduto=${produto.idProduto}">Alterar</a> |
              <a href="ServletProdutoFC?acao=excluir&idProduto=${produto.idProduto}">Excluir</a></td>
          </tr>
        </c:forEach>
      </tbody>
    </table>
  </body>
</html>
```

Arquivo: [web/ProdutoDados.jsp](#)

```
<%--
Document    : ProdutoDados
Created on  : Aug 21, 2023, 5:14:12 PM
Author      : Mari
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Dados do Produto</title>
  </head>
  <body>
    <h1>Dados do Produto</h1>
    <form action="ServletProdutoFC" method="post">
      <input type="hidden" name="acao" value="{produto == null ? 'incluir' : 'alterar'}">
      <input type="hidden" name="idProduto" value="{produto != null ? produto.idProduto : ''}">
      <label for="nome">Nome:</label>
      <input type="text" name="nome" id="nome" value="{produto != null ? produto.nome : ''}">
      <br>
      <label for="quantidade">Quantidade:</label>
      <input type="number" name="quantidade" id="quantidade" value="{produto != null ?
produto.quantidade : ''}">
      <br>
      <label for="precoVenda">Preço de Venda:</label>
      <input type="number" name="precoVenda" id="precoVenda" min="0" step="0.01" value="{produto != null
? produto.precoVenda : ''}">
      <br>
      <button type="submit">{produto == null ? 'Adicionar' : 'Alterar'} Produto</button>
    </form>
  </body>
</html>
```

4. Resultados da execução dos códigos

localhost:8080/CadastroEE-war/S ×

Listagem de Produtos ×

localhost:8080/CadastroEE-war/ServletProdutoFC

Listagem de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana	100	5.0	Alterar Excluir
2	Laranja	500	2.0	Alterar Excluir
3	Laranja	800	4.0	Alterar Excluir
1001	Morango	500	4.95	Alterar Excluir

localhost:8080/CadastroEE-war/S ×

Dados do Produto ×

localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formIncluir

Dados do Produto

Nome:

Quantidade:

Preço de Venda:

localhost:8080/CadastroEE-war/S ×

Dados do Produto ×

localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formAlterar&idProd... A

Dados do Produto

Nome:

Quantidade:

Preço de Venda:

5. Análise e Conclusão

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

Resposta: O padrão Front Controller é responsável por receber todas as requisições e encaminhá-las ao recurso apropriado, servindo como um “roteador” das requisições. É possível utilizar um Servlet que atua como Front Controller repassando e carregando outros controladores ou páginas JSP, por exemplo, a partir dele.

- b) Quais as diferenças e semelhanças entre Servlets e JSPs?

Resposta: Os Servlets são classes Java que processam solicitações e geram respostas, enquanto JSPs são páginas HTML que contêm código Java embutido para processar os dados. Os Servlets geralmente são usados para gerar conteúdo dinâmico em tempo real, enquanto JSPs são frequentemente usadas para gerar conteúdo dinâmico que muda com menos frequência.

- 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?

Resposta: O redirecionamento simples é usado para enviar uma resposta de redirecionamento ao cliente, que o levará para uma nova página, o que gera uma nova requisição para a nova página. O método forward é usado para encaminhar a solicitação para outro recurso no servidor, sendo processada ainda no servidor antes de enviar uma nova resposta para o cliente.