



Estácio

Faculdade Estácio

Campus Belford Roxo - RJ

Desenvolvimento Full Stack

Disciplina - **Vamos Integrar Sistema**

Turma 2023.2

Semestre - 3

Cristian da Silva de Macena

Repositório: <https://github.com/CristianS34/JavaEE.git>

Interface Cadastral com Servlet e JSPs

Código

```

package cadastroee.servlets;

import cadastroee.controller.ProdutosFacadeLocal;
import cadastroee.model.Produtos;
import jakarta.ejb.EJB;
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.List;

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

    @EJB
    private ProdutosFacadeLocal facade;

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        String acao = request.getParameter("acao");
        String destino = "";

        switch (acao) {
            case "listar":
                List<Produtos> produtos = facade.findAll();
                request.setAttribute("produtos", produtos);
                destino = "ProdutoLista.jsp";
                break;
            case "formIncluir":
            case "formAlterar":
                if ("formAlterar".equals(acao)) {
                    String id = request.getParameter("id");
                    Produtos produto = facade.find(Integer.parseInt(id));
                    request.setAttribute("produto", produto);
                }
                destino = "ProdutoDados.jsp";
                break;
            case "excluir":
                String idExcluir = request.getParameter("id");
                Produtos produtoExcluir = facade.find(Integer.parseInt(idExcluir));
                facade.remove(produtoExcluir);
                request.setAttribute("produtos", facade.findAll());
                destino = "ProdutoLista.jsp";
                break;
        }
    }
}

```

```

        case "alterar":
            String idAlterar = request.getParameter("id");
            Produtos produtoAlterar = facade.find(Integer.parseInt(idAlterar));
            produtoAlterar.setNome(request.getParameter("nome"));

            produtoAlterar.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));
            produtoAlterar.setPrecoVenda(Long.parseLong(request.getParameter("preco")));
            facade.edit(produtoAlterar);
            request.setAttribute("produtos", facade.findAll());
            destino = "ProdutoLista.jsp";
            break;
        case "incluir":
            Produtos novoProduto = new Produtos();
            novoProduto.setNome(request.getParameter("nome"));

            novoProduto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));
            novoProduto.setPrecoVenda(Long.parseLong(request.getParameter("preco")));
            facade.create(novoProduto);
            request.setAttribute("produtos", facade.findAll());
            destino = "ProdutoLista.jsp";
            break;
        default:
            destino = "ProdutoLista.jsp";
            break;
    }

    RequestDispatcher dispatcher = request.getRequestDispatcher(destino);
    dispatcher.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);
}

@Override
public String getServletInfo() {
    return "Servlet Produto Front Controller";
}
}

```

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <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 ? 'alterar' : 'incluir'}">
        <c:if test="{produto != null}">
            <input type="hidden" name="id" value="{produto.idProduto}">
        </c:if>
        <div>
            <label for="nome">Nome:</label>
            <input type="text" id="nome" name="nome" value="{produto != null ? produto.nome
: ""}">
        </div>
        <div>
            <label for="quantidade">Quantidade:</label>
            <input type="number" id="quantidade" name="quantidade" value="{produto != null ?
produto.quantidade : ""}">
        </div>
        <div>
            <label for="preco">Preço:</label>
            <input type="text" id="preco" name="preco" value="{produto != null ?
produto.precoVenda : ""}">
        </div>
        <div>
            <button type="submit">{produto != null ? 'Alterar' : 'Incluir'}</button>
        </div>
    </form>
</body>
</html>

```

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Lista de Produtos</title>
</head>
<body>
    <h1>Lista de Produtos</h1>
    <a href="ServletProdutoFC?acao=formIncluir">Incluir Novo Produto</a>
    <table border="1">
        <thead>
            <tr>
                <th>ID</th>

```

```

        <th>Nome</th>
        <th>Quantidade</th>
        <th>Preço</th>
        <th>Ações</th>
    </tr>
</thead>
<tbody>
    <c:forEach var="produto" items="{produtos}">
        <tr>
            <td>${produto.idProduto}</td>
            <td>${produto.nome}</td>
            <td>${produto.quantidade}</td>
            <td>${produto.precoVenda}</td>
            <td>
                <a
href="ServletProdutoFC?acao=formAlterar&id=${produto.idProduto}">Alterar</a>
                <a
href="ServletProdutoFC?acao=excluir&id=${produto.idProduto}">Excluir</a>
            </td>
        </tr>
    </c:forEach>
</tbody>
</table>
</body>
</html>

```

Lista de Produtos

[Incluir Novo Produto](#)

ID	Nome	Quantidade	Preço	Ações
				Alterar Excluir

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 design pattern que centraliza o tratamento de todas as requisições de um aplicativo web em um único ponto de entrada. Essa centralização oferece diversos benefícios, como maior controle sobre a aplicação, facilidade de manutenção e a possibilidade de aplicar filtros e interceptores.

Quais as diferenças e semelhanças entre Servlets e JSPs?

Servlets: São classes Java que estendem a funcionalidade de um servidor web. Todo o código, incluindo a lógica de negócios e a geração de HTML, é escrito em Java. **JSPs:** São páginas HTML que podem incluir código Java usando tags especiais. A lógica de apresentação é separada da lógica de negócios, tornando o código mais fácil de entender e manter.

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?

Redirecionamento simples faz o navegador enviar uma nova requisição, enquanto forward encaminha a requisição internamente no servidor, mantendo os dados originais.