



# Estácio

Faculdade Estácio - Polo Centro - Canela - RS

Curso: Desenvolvimento Full Stack

Disciplina: RPG0017 - Vamos Integrar Sistemas

Turma: 9001 - Semestre Letivo: 2025.1 - 3º semestre

Integrante: Pedro Henrique Marques Medeiros Pinho

Matrícula: 202402031831

IDE: Sql Server Management Studio

Repositório Git: <https://github.com/PedroPinho23/Vamos-Integrar-Sistemas.git>

# Vamos Integrar Sistemas

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

## Objetivos da prática

- Implementar persistência com base em JPA.
- Implementar regras de negócio na plataforma JEE, através de EJBs.
- Implementar sistema cadastral Web com base em Servlets e JSPs.
- Utilizar a biblioteca Bootstrap para melhoria do design.
- 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.

## Códigos Utilizados

### 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 cadastroee.controller.ProdutoFacadeLocal;
import jakarta.ejb.EJB;
import java.io.IOException;
import java.io.PrintWriter;
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 cadastroee.model.Produto;
/**
 *
 * @author Pedro
 */
@WebServlet("/ServletProduto")
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()) {
            /* TODO output your page here. You may use following sample code.
            */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet ServletProduto</title>");
            out.println("</head>");
            out.println("<body>");
            // out.println("<h1>Servlet ServletProduto at " +
            request.getContextPath() + "</h1>");
            //out.println(facade.findAll().getClass());
            //out.println(facade.find(1).getClass());
            for (Produto p : facade.findAll()) {
                out.println("<li> " + p.getNome() + "</li>");
            }
            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>
}
```

## 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 cadastroee.controller.ProdutoFacadeLocal;
import cadastroee.model.Produto;
import jakarta.ejb.EJB;
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.annotation.WebServlet;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
/**
 *
 * @author Pedro
 */
@WebServlet(name="ServletProdutoFC", urlPatterns = {"/ServletProdutoFC"})
public class ServletProdutoFC extends HttpServlet {
    @EJB
    ProdutoFacadeLocal facade;
    int idAtual =6;
    public int aleatorio() {
        // Math.random() gera um número aleatório entre 0.0 e 0.999
        // Assim, Math.random()*5 estará entre 0.0 e 4.999
        double doubleRandomNumber = Math.random() * 100;
        int randomNumber = (int)doubleRandomNumber;
        return randomNumber;
    }
}
/**
 * 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 {
    String acao = request.getParameter("acao");
    String destino = "";
    if(acao != null){
        switch(acao){
            case "listar":
                request.setAttribute("lista", facade.findAll());
                destino = "ProdutoLista.jsp";
                break;
            case "excluir":
                int idProduto =
                    Integer.valueOf(request.getParameter("idproduto"));
                facade.remove(facade.find(idProduto));
                request.setAttribute("lista", facade.findAll());
                RequestDispatcher rd =
                    request.getRequestDispatcher("ProdutoLista.jsp");
                rd.forward(request, response);
                break;
            case "formIncluir":
                destino = "ProdutoDados.jsp";
                break;
            case "formAlterar":
                int id_produto =
                    Integer.valueOf(request.getParameter("idproduto"));
                request.setAttribute("lista", facade.find(id_produto));
                destino = "ProdutoDados.jsp";
                break;
        }
        RequestDispatcher rd =
            request.getRequestDispatcher(destino);
        rd.forward(request, response);
    } else{
        request.setAttribute("lista", facade.findAll());
        RequestDispatcher rd =
            request.getRequestDispatcher("ProdutoLista.jsp");
        rd.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 {
String acao = request.getParameter("acao");
switch(acao){
case("alterar"):
int produtoID =
Integer.valueOf(request.getParameter("idproduto"));
String nome = request.getParameter("nome");
int quantidadeProduto =
Integer.valueOf(request.getParameter("quantidade"));
float precoProduto =
Float.valueOf(request.getParameter("preco"));
Produto produtoAlterar = facade.find(produtoID);
produtoAlterar.setNome(request.getParameter("nome"));
produtoAlterar.setQuantidade(quantidadeProduto);
produtoAlterar.setPrecoVenda(precoProduto);
facade.edit(produtoAlterar);
request.setAttribute("lista", facade.findAll());
RequestDispatcher rd =
request.getRequestDispatcher("ProdutoLista.jsp");
rd.forward(request, response);

```

```

break;
case("incluir"):
int idNext = aleatorio();
if (idNext != idAtual){
float preco =
Float.valueOf(request.getParameter("preco"));
String nome2 = request.getParameter("nome");
int quantidade =
Integer.valueOf(request.getParameter("quantidade"));
Produto produto = new Produto(idNext,
request.getParameter("nome"),
quantidade,
preco);
facade.create(produto);
request.setAttribute("lista",
facade.findAll());
idAtual = idNext;
RequestDispatcher rd2 =
request.getRequestDispatcher("ProdutoLista.jsp");
rd2.forward(request, response);
break;
} else {
idNext = idNext++;
idAtual = idNext;
float preco =
Float.valueOf(request.getParameter("preco"));
String nome2 = request.getParameter("nome");
int quantidade =
Integer.valueOf(request.getParameter("quantidade"));
Produto produto = new Produto(idNext,
request.getParameter("nome"),
quantidade,
preco);
facade.create(produto);
request.setAttribute("lista",
facade.findAll());
idAtual = idNext;
RequestDispatcher rd2 =
request.getRequestDispatcher("ProdutoLista.jsp");
rd2.forward(request, response);
break;
}
}
}
}

```



```

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

```

### **ProdutoLista.jsp**

```

<%@page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@page import="cadastroee.model.Produto"%>
<%@page import="java.util.ArrayList"%>
<%@page import="java.util.List"%>
<%@page import="cadastroee.controller.ProdutoFacadeLocal"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<div>
<h1>Listagem de Produtos</h1>
<h3>
<a href="ServletProdutoFC?acao=formIncluir"> Novo Produto
</a>
<!-- <a href="ServletProdutoFC"> Atualizar Dados </a> !-->
</h3>
<table border="1" width="100%">
<tr>
<td> ID </td>
<td> Nome </td>
<td> Quantidade </td>
<td> Preço de Venda </td>
<td> Opções </td>
</tr>
<%
try{
List<Produto> lista = (List)

```

```

request.getAttribute("lista");
for(Produto p: lista){
%>
<tr>
<td>
<%=p.getIdproduto()%>
</td>
<td>
<%=p.getNome()%>
</td>
<td>
<%=p.getQuantidade()%>
</td>
<td>
<%=p.getPrecoVenda()%>
</td>
<td>
<a
href="ServletProdutoFC?acao=formAlterar&idproduto=<%=p.getIdproduto()%>">
Alterar </a>
<a
href="ServletProdutoFC?acao=excluir&idproduto=<%=p.getIdproduto()%>"> Excluir
</a>
</td>
</tr>
<% }
} catch(NullPointerException nexc){
out.print("<h1>"+nexc.getMessage()+"</h1>");
}
%>
</table>
</div>
</body>
</html>

```

## ProdutoDados.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="cadastroee.model.Produto"%>
<%@page import="java.util.ArrayList"%>
<%@page import="java.util.List"%>
<%@page import="cadastroee.controller.ProdutoFacadeLocal"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<%
try{
Produto produto = (Produto) request.getAttribute("lista");
if (produto != null){
%>
<h1> Dados do Produto </h1>
<form action="ServletProdutoFC" method="post" >
<input type="hidden" name="acao" value="alterar">
<input type="hidden" name="idproduto"
value="<%=produto.getIdproduto()%>">
Nome: <input name="nome" value="<%=produto.getNome()%>" />
Quantidade: <input name="quantidade"
value="<%=produto.getQuantidade()%>" />
Preco de Venda: <input name="preco"
value="<%=produto.getPrecoVenda()%>" />
<input type="submit" value="Alterar Produto" />
</form>
<%
} else {
%>
<h1> Dados do Produto </h1>
<form action="ServletProdutoFC" method="post" >
<input type="hidden" name="acao" value="incluir">
Nome: <input name="nome" />
Quantidade: <input name="quantidade" />
Preco de Venda: <input name="preco" />
<input type="submit" value="Adicionar Produto" />
</form>
<%
}
}
```

```
} catch(ClassCastException nexc){  
out.print("<h1>"+nexc.getMessage()+"</h1>");}  
%>  
</body>  
</html>
```

## Resultados:

Foram implementadas e testadas as funcionalidades de persistência com JPA, regras de negócio via EJBs e interface web utilizando Servlets, JSPs e Bootstrap. Os testes confirmaram que os dados foram corretamente armazenados, recuperados e manipulados no banco SQL Server, além de garantir a correta interação entre as camadas da aplicação, comprovando o funcionamento integrado da plataforma Java EE.

## Conclusão:

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

O padrão Front Controller centraliza todas as requisições do usuário em um único servlet, que decide qual lógica aplicar e qual página exibir. Em uma aplicação Java Web com arquitetura MVC, esse servlet atua como controlador, chamando os modelos (regras de negócio) e direcionando para as views (JSPs), organizando melhor o fluxo da aplicação.

Quais as diferenças e semelhanças entre Servlets e JSPs?

Servlets e JSPs são usados para criar páginas dinâmicas em aplicações web. Servlets focam na lógica de controle usando Java puro, enquanto JSPs são voltadas para a apresentação, misturando HTML com Java. Ambos funcionam no servidor e se complementam na arquitetura MVC.

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?

O redirecionamento simples envia uma nova requisição do cliente para outra página, mudando a URL no navegador. Já o método forward com RequestDispatcher repassa a requisição internamente no servidor, mantendo a mesma URL. Parâmetros no HttpServletRequest servem para receber dados enviados pelo cliente. Atributos servem para compartilhar dados entre componentes no servidor durante o processamento da mesma requisição.