



**POLO BARRA WORLD**  
**CURSO DESENVOLVIMENTO FULL STACK**

**Disciplina:** Nível 4 – Vamos Integrar Sistemas

**Turma:** 2024.3 Flex / 3º Semestre

**Aluno:** Paola Savedra Barreiros

**Matrícula:** 2023.0701.4731

**Repositório Github:** [pasavedra/RPG0017-Vamos-integrar-sistemas](https://github.com/pasavedra/RPG0017-Vamos-integrar-sistemas)

**Missão Prática | Nível 4 | Mundo 3**

**Título da prática:** RPG0017 - Vamos integrar sistemas

**Objetivo 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 SOLICITADOS NO ROTEIRO

### PROCEDIMENTO 01: Camadas de Persistência e Controle

- **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 pasav  
  
 */  
  
  
  
  
  
  
  
  
  
@WebServlet("/ServletProduto")  
  
public class ServletProduto extends HttpServlet {
```

@EJB

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 {

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>

}

```

## PROCEDIMENTO 02: INTERFACE CADASTRAL COM SERVLET E JSPS

- **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 pasav
```

```
*/
```

```
@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";
    }
}

```



- **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 pasav
 */

@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";  
}  
}  
}
```

- **ProdutoLista.jsp**

```
<%--
```

```
Document : ProdutoLista
```

```
Created on : 3 de nov. de 2024, 21:00:09
```

```
Author : pasav
```

```
--%>
```

```
<%@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>
```





```

        </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**

```
<%--
```

```
Document : ProdutoDados
```

```
Created on : 3 de nov. de 2024, 13:26:07
```

```
Author : pasav
```

```
--%>
```

```
<%@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()%>" />

        Preço 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" />

    Preço de Venda: <input name="preco" />

    <input type="submit" value="Adicionar Produto" />

    </form>

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

</body>

</html>

```

### Procedimento 3: Melhorando o Design da Interface

- **ProdutoLista.jsp**

```
<%--
```

```
Document : ProdutoLista
```

```
Created on : 3 de nov. de 2024, 21:00:09
```

```
Author : pasav
```

```
--%>
```

```
<%@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">
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6Coli6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous"> <title>JSP Page</title>
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
```

```
</head>
```

```

<body class="container">

    <div>

        <h1>Listagem de Produtos</h1>

        <h3>

            <a class="btn btn-primary m-2" href="ServletProdutoFC?acao=formIncluir"> Novo
Produto </a>

            <!-- <a href="ServletProdutoFC"> Atualizar Dados </a> !-->

        </h3>

        <table class="table table-striped" border="1" width="100%">

            <tr class="table-dark">

                <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>

                        </tr>

```

```

        <td>

            <%=p.getQuantidade()%>

        </td>

        <td>

            <%=p.getPrecoVenda()%>

        </td>

        <td>

            <a class="btn btn-primary btn-sm"
href="ServletProdutoFC?acao=formAlterar&idproduto=<%=p.getIdproduto()%>"> Alterar
</a>

            <a class="btn btn-danger btn-sm"
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**

<%--

Document : ProdutoDados

Created on : 3 de nov. de 2024, 13:26:07

Author : pasav

--%>

<%@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">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6Coli6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous"> <title>JSP Page</title>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrmMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

</head>

<body class="container">

<%

try{

Produto produto = (Produto) request.getAttribute("lista");

```

        if (produto != null){
%>

        <h1> Dados do Produto </h1>

        <form class="form" action="ServletProdutoFC" method="post" >

            <input type="hidden" name="acao" value="alterar">

            <input type="hidden" name="idproduto"
value="<%=produto.getIdproduto()%>">

            <div class=" mb-3">

                <label class="form-label"> Nome: </label>

                <input class="form-control" name="nome"
value="<%=produto.getNome()%>" />

            </div>

            <div class=" mb-3">

                <label class="form-label">Quantidade: </label>

                <input class="form-control" name="quantidade"
value="<%=produto.getQuantidade()%>" />

            </div>

            <div class=" mb-3">

                <label class="form-label"> Preço de Venda: </label>

                <input class="form-control" name="preco"
value="<%=produto.getPrecoVenda()%>" />

            </div>

            <div class=" mb-3">

                <input class=" btn btn-primary" type="submit" value="Alterar Produto"/>

            </div>

        </form>

    <%

        } else {

    %>

```



```
<h1> Dados do Produto </h1>

<form class="form" action="ServletProdutoFC" method="post" >

    <input type="hidden" name="acao" value="incluir">

    <div class=" mb-3">

        <label class="form-label">Nome: </label>

        <input class="form-control" name="nome"/>

    </div>

    <div class=" mb-3">

        <label class="form-label">Quantidade: </label>

        <input class="form-control" name="quantidade"/>

    </div>

    <div class=" mb-3">

        <label class="form-label">Preco de Venda: </label>

        <input class="form-control" name="preco"/>

    </div>

    <input class=" btn btn-primary" type="submit" value="Adicionar Produto"/>

</form>

<%

    }

    } catch(ClassCastException nexc){

        out.print("<h1>"+nexc.getMessage()+"</h1>");}

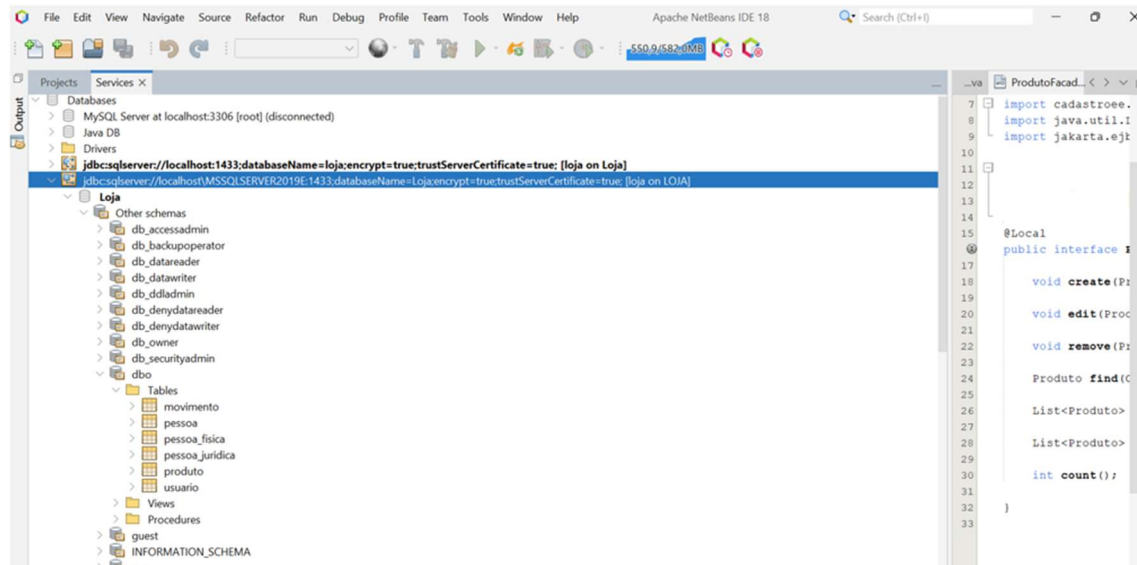
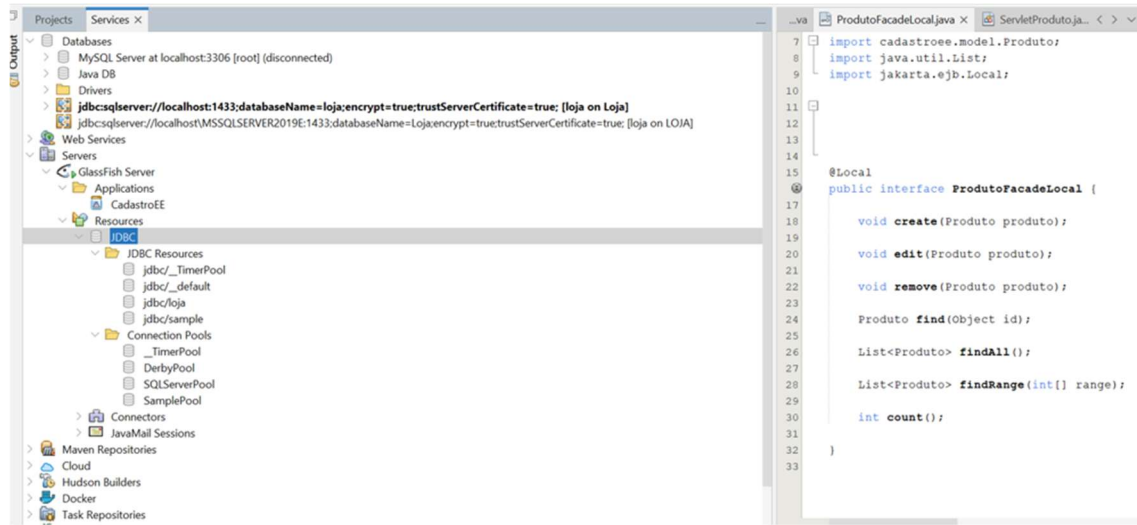
    %>

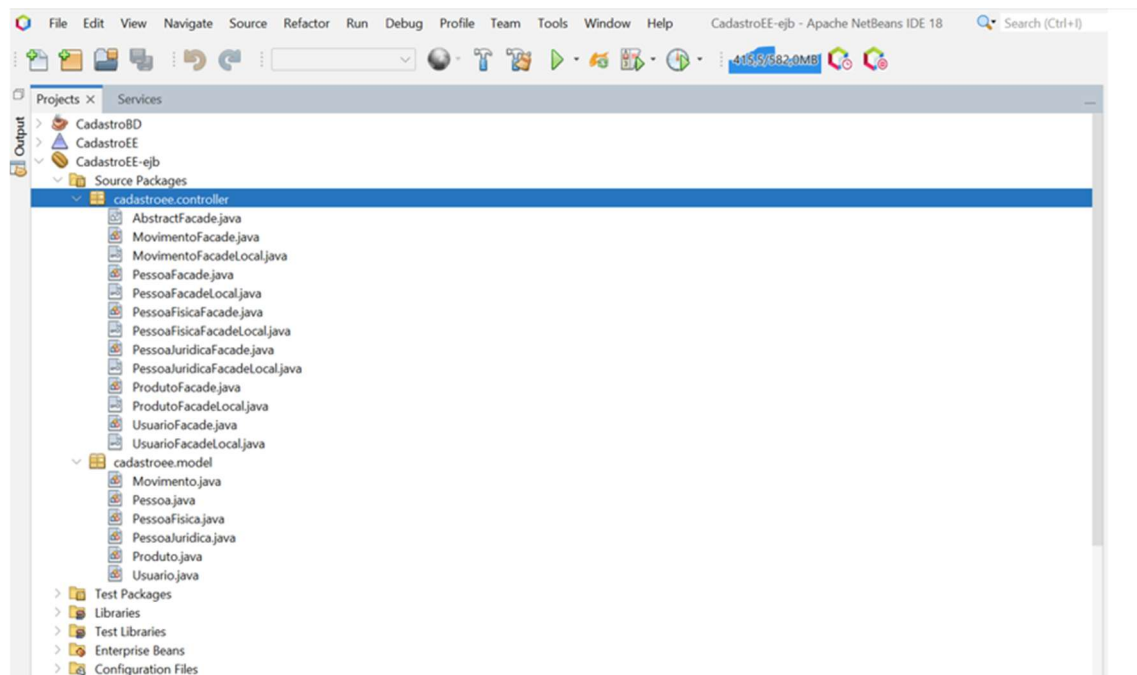
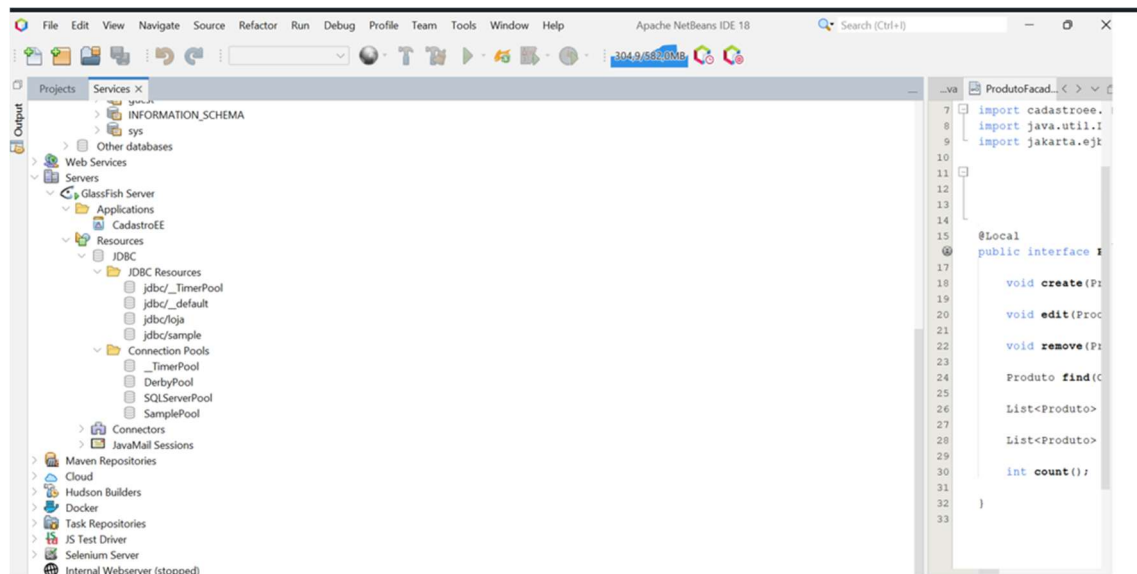
</body>

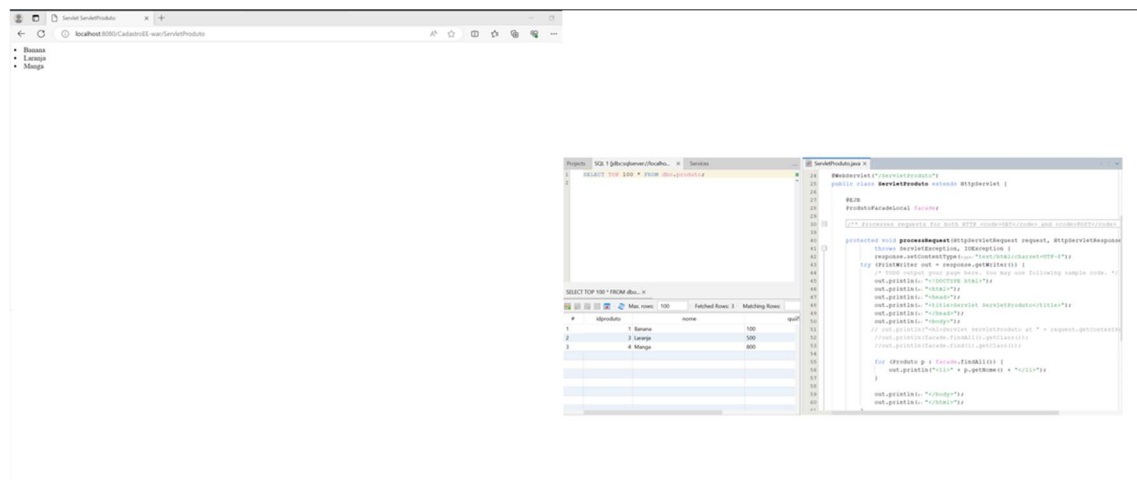
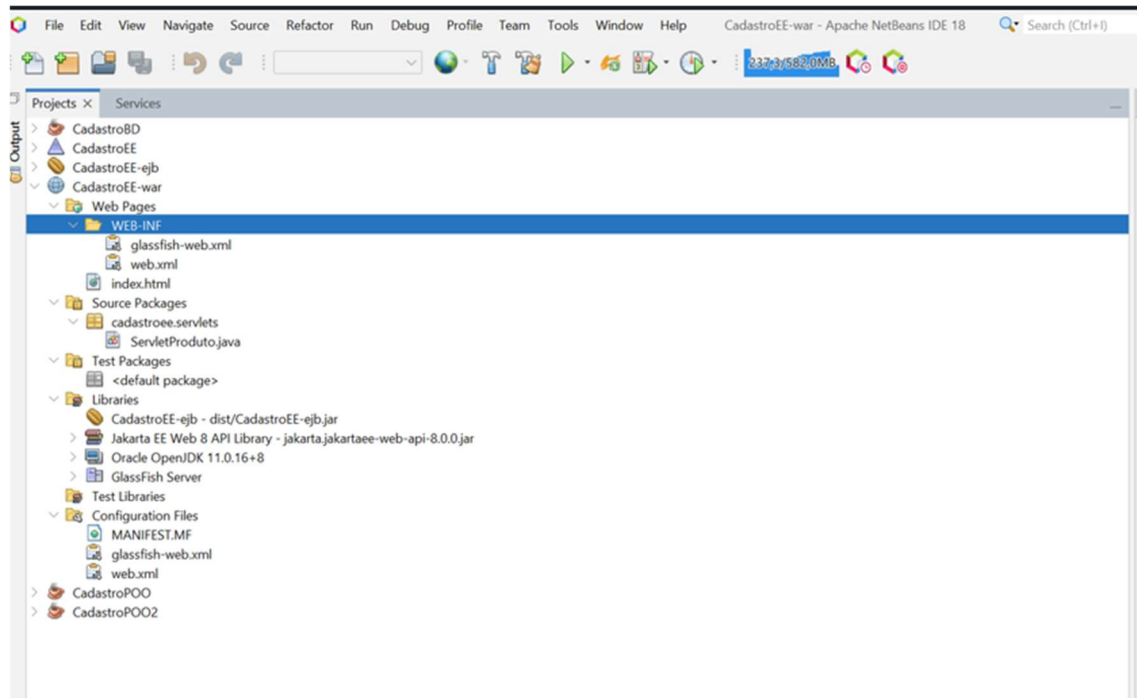
</html>
```

# RESULTADOS NA PRÁTICA

## PROCEDIMENTO 01







# RESULTADOS NA PRÁTICA

## PROCEDIMENTO 02

← → ↻

localhost:8080/CadastroEE-war/ServletProdutoFC

🔍 ☆ 🌟 📄 ⬛ ⬛ ⬛ ⬛ ⬛

Listagem de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana	100	5.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
3	Laranja	500	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Manga	800	4.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
14	Pinha	50	50.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

← → ↻

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

🔍 ☆ 🌟 📄 ⬛ ⬛ ⬛ ⬛ ⬛

Dados do Produto

Nome:  Quantidade:  Preço de Venda:

← → ↻

localhost:8080/CadastroEE-war/ServletProdutoFC

🔍 ☆ 🌟 📄 ⬛ ⬛ ⬛ ⬛ ⬛

Listagem de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana	100	5.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
3	Laranja	500	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Manga	800	4.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
5	Pera	150	0.7	<a href="#">Alterar</a> <a href="#">Excluir</a>
14	Pinha	50	50.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

← → ↻ localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formAlterar&idproduto=1

## Dados do Produto

Nome:  Quantidade:  Preço de Venda:

← → ↻ localhost:8080/CadastroEE-war/ServletProdutoFC

## Listagem de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana Pacovan	100	5.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
3	Laranja	500	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Manga	800	4.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
5	Pera	150	0.7	<a href="#">Alterar</a> <a href="#">Excluir</a>
14	Pinha	50	50.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

← → ↻ localhost:8080/CadastroEE-war/ServletProdutoFC?acao=excluir&idproduto=14

## Listagem de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana Pacovan	100	5.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
3	Laranja	500	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Manga	800	4.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
5	Pera	150	0.7	<a href="#">Alterar</a> <a href="#">Excluir</a>

# RESULTADOS NA PRÁTICA

## PROCEDIMENTO 03

← ↻ ⓘ localhost:8080/CadastroEE-war/ServletProdutoFC?acao=listar 🔍 ⌵ ⭐ 📄 ⚙️

### Listagem de Produtos

Novo Produto

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana Pacovan	100	5.0	<div>Alterar Excluir</div>
3	Laranja	500	2.0	<div>Alterar Excluir</div>
4	Manga	800	4.0	<div>Alterar Excluir</div>
5	Pera	150	0.7	<div>Alterar Excluir</div>

← ↻ ⓘ localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formincluir 🔍 ⌵ ⭐ 📄 ⚙️

### Dados do Produto

Nome:

Quantidade:

Preço de Venda:

Adicionar Produto

← ↻ ⓘ localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formAlterar&idproduto=1 🔍 ⌵ ⭐ 📄 ⚙️

### Dados do Produto

Nome:

Quantidade:

Preço de Venda:

Alterar Produto

## **ANÁLISE E CONCLUSÃO:**

### **1. Como é organizado um projeto corporativo no NetBeans?**

O projeto corporativo é dividido em um módulo EJB. Nele ficam localizadas as classes relativas aos modelos e controles, e um módulo Web, onde ficam os arquivos JSP, HTML e Servlet.

### **2. Qual o papel das tecnologias JPA e EJB na construção de um aplicativo para a plataforma Web no ambiente Java?**

Para que serve: JPA é a tecnologia padrão para gerenciamento de bancos de dados em Java, e permite trabalhar de maneira simples e orientada a objetos com banco de dados. JSF é a tecnologia padrão para geração de interfaces Web. Utilizando EJB para integrá-las, é possível criar aplicações web de maneira simples e poderosa.

### **3. Como o NetBeans viabiliza a melhoria de produtividade ao lidar com as tecnologias JPA e EJB?**

Desenvolvedor pode programar sem se preocupar com conceitos relacionais usando apenas conceitos de orientação a objetos.

### **4. O que são Servlets, e como o NetBeans oferece suporte à construção desse tipo de componentes em um projeto Web?**

Servlets são classes Java, desenvolvidas de acordo com uma estrutura bem definida que quando instaladas e configuradas em um Servidor que implemente um Servlet Container, podem tratar requisições recebidas de clientes Web, como por exemplo os Browsers. Ao receber uma requisição, um Servlet pode capturar os parâmetros desta requisição, efetuar qualquer processamento inerente a uma classe Java, e devolver uma página HTML.

### **5. Como é feita a comunicação entre os Servlets e os Session Beans do pool de EJBs?**

A comunicação entre Servlets e Session Beans no contexto de um pool de EJBs é feita de maneira diferente do que em um aplicativo Web. Em um aplicativo EJB, um cliente nunca chama diretamente os métodos de um bean. A comunicação entre clientes e beans é feita através do container EJB, que é o intermediário que permite a interação entre os componentes do aplicativo. Isso é similar a como um cliente browser web precisa usar o container de web para usar um servlet ou uma página JSP. Um aplicativo EJB com um servlet como cliente, onde o servlet atua como um intermediário para chamar os métodos do bean, que é então executado pelo container EJB.



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

É um padrão arquitetural que se comporta como um controlador tratando todas as solicitações para um site Web e então roteia para uma ação (ou comando). o Front Controller trata todas as chamadas vindas de um site web e é organizado em duas partes: através de um Manipulador Web e uma hierarquia de Comandos. O Manipulador Web é o objeto que efetivamente recebe as solicitações HTTP do tipo POST ou GET do servidor web. Ele extrai as informações necessárias da URL e das solicitações e então decide que tipo de ação iniciar e por fim delega a um objeto Comando para executar a ação.

## **7. Quais as diferenças e semelhanças entre Servlets e JSPs?**

- O servlet pode aceitar todas as solicitações de protocolo, incluindo HTTP, enquanto o JSP só pode aceitar solicitações HTTP.
- Na arquitetura MVC, o Servlet funciona como um controlador, enquanto o JSP funciona como uma visualização para exibir a saída.
- O servlet deve ser usado quando há mais processamento de dados envolvido, enquanto o JSP geralmente é usado quando há menos envolvimento no processamento de dados.
- Servlets rodam mais rápido que JSP; por outro lado, o JSP é executado mais lentamente que o Servlet, pois leva tempo para compilar o programa e convertê-lo em servlets.
- Você pode substituir o método `service()` no Servlet, mas em JSP não pode substituir o método `service()`.
- No Servlet, você deve implementar a lógica de negócios e a lógica de apresentação em um único arquivo. Enquanto em JSP, a lógica de negócios é separada da lógica de apresentação usando JavaFeijões.

## **8. 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?**

Redirect redireciona o cliente para uma página (`sendRedirect`) e RequestDispatcher encaminha uma requisição para ser atendida por outro recurso (`forward`). No primeiro caso (`sendRedirect`), o cliente receberá uma resposta http em cujo header haverá a informação de que ele deve requisitar outra página, e o browser fará esta requisição. Ou seja, o redirecionamento ocorre no lado no cliente. No segundo caso (`forward`), no lado do server a requisição do usuário será encaminhada para ser atendida por outro recurso (outro servlet). Este outro servlet eventualmente devolverá outra página para o usuário. Os parâmetros são as informações da página submetidos por um formulário, por exemplo. Nesse caso, cada controle de entrada de dados é um parâmetro, e todos eles pertencem à requisição HTTP. Se você submeter dados através do método GET, os parâmetros ficam visíveis na URL no seguinte formato: `url?param1=valor1&param2=valor2` etc. Já os atributos são objetos associados a nomes - uma espécie de tabela onde a chave é uma string - que ficam guardados no servidor, associados a um determinado escopo que pode ser de página ( `PageContext` ), requisição ( `HttpServletRequest` ), sessão ( `HttpSession` ) ou aplicação ( `ServletContext` ).

## **9. Como o framework Bootstrap é utilizado?**

Bootstrap é um framework front-end que fornece estruturas de CSS para a criação de sites e aplicações responsivas de forma rápida e simples. Ele é livre e de código aberto, baseado em HTML, CSS e Javascript. O Bootstrap é ideal para o desenvolvimento de sites para dispositivos móveis. Ele oferece padrões para HTML, JavaScript e CSS, o que facilita e agiliza o trabalho de desenvolvedores de páginas web.

## **10. Por que o Bootstrap garante a independência estrutural do HTML?**

Com uma vasta biblioteca de componentes e estilos pré-definidos, os desenvolvedores podem evitar a necessidade de criar tudo do zero. Isso não apenas acelera o processo de desenvolvimento, mas também garante uma consistência visual em todo o site.

## **11. Qual a relação entre o Bootstrap e a responsividade da página?**

Permitir que os elementos da página sejam readaptados para o acesso em diferentes dispositivos, como notebooks, tablets, smartphones e, até mesmo, para monitores maiores que os tradicionais.