



Estácio

Universidade Estácio de Sá

-
- Curso: Desenvolvimento Full stack - TURMA 9001
 - Disciplina: RPG0017 - Vamos Integrar Sistemas.
 - Semestre Letivo: 2024.4
 - Aluno: Jonathan Sendi Inowe
 - Matrícula: 202311117502
-

Missão Prática | Nível 4 | Mundo 3

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

Procedimento 1: Camadas de Persistência e Controle

Procedimento 2: Interface Cadastral com Servlet e JSPs

Procedimento 3: Melhorando o Design da Interface



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

Procedimento 1: 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;

@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>
}

```

Procedimento 2: 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;

@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;

```

```

@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

```

<%--
    Document   : ProdutoLista
    Created on : 3 de fev. de 2025, 21:16:09
    Author    : jon
--%>

<%@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

```
<%--
```

```
Document : ProdutoDados
```

```
Created on : 5 de fev. de 2025, 14:26:07
```

```
Author : jon
```

```
--%>
```

```
<%@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>

```

Procedimento 3: Melhorando o Design da Interface

- ProdutoLista.jsp

```

<%--
    Document : ProdutoLista
    Created on : 3 de fev. de 2025, 21:16:09
    Author : jon
--%>

```

```

<%@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/Dwwykc2MPK8M2
HN" 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/o0jlpcV8Qyq46cDf
L" 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>
        <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  : 5 de fev. de 2025, 14:26:07
    Author      : jon
--%>

```

```

<%@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/Dwwykc2MPK8M2
HN" 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/o0jlpcV8Qyq46cDf
L" 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"> Preco 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">Preço 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>");}

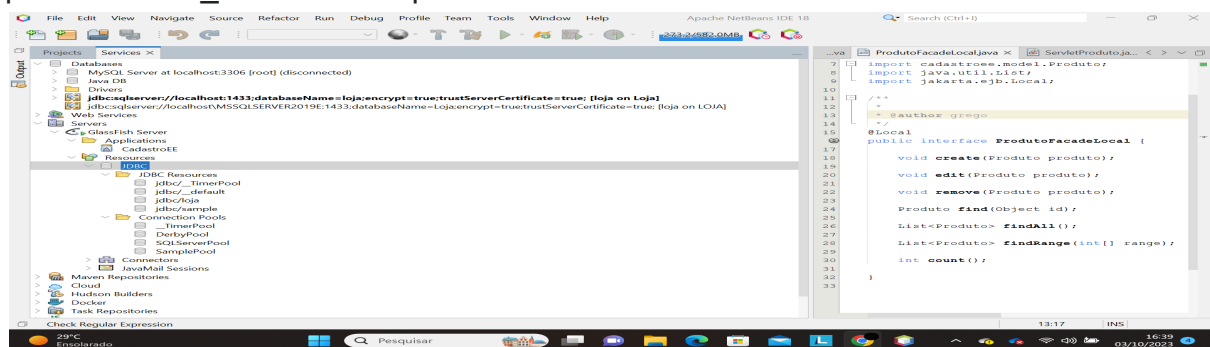
%>
</body>
</html>

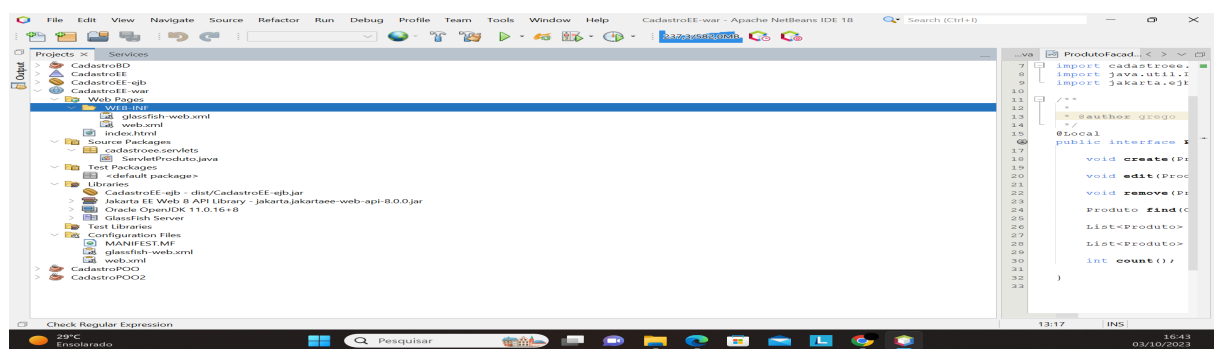
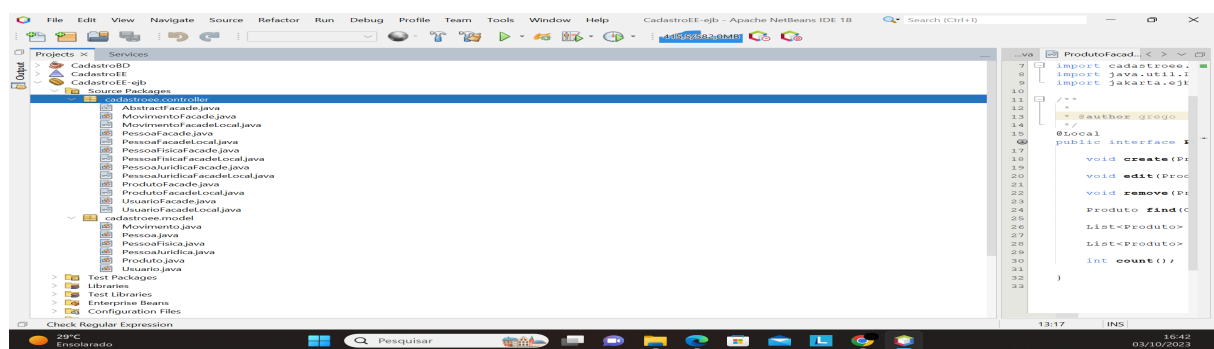
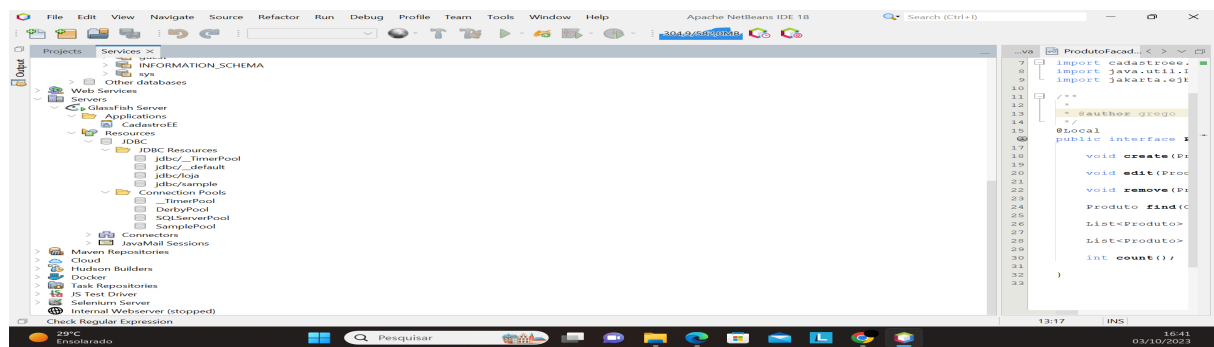
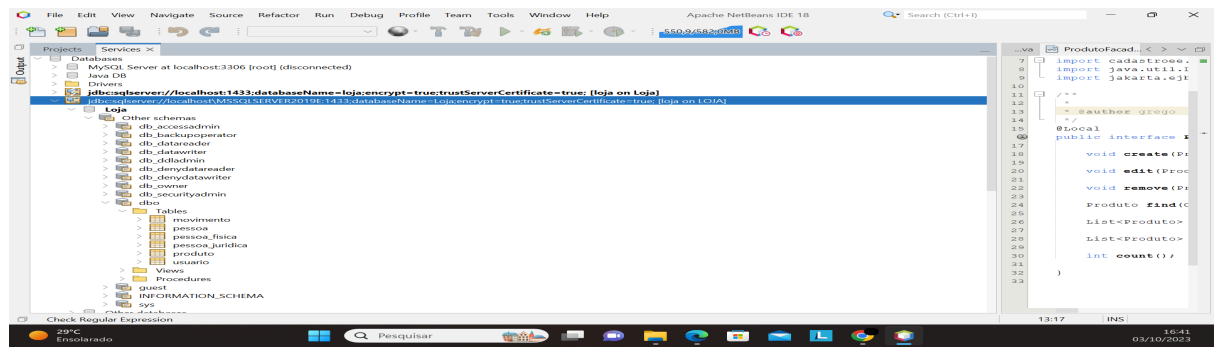
```

Resultados:

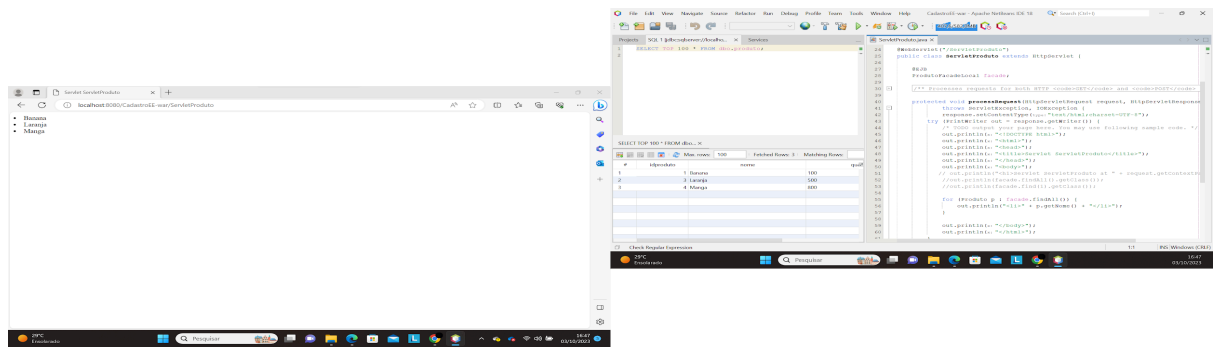
🚩 Procedimento 1:

https://github.com/joninowe/Missao-nivel-4-mundo-3/tree/main/Procedimento%201proc-1-nivel-4_0nPADzdQ.mp4





Teste do Servlet:
<http://localhost:8080/CadastrEE-war/ServletProduto>



Procedimento 2:
https://github.com/joninowe/Missao-nivel-4-mundo-3/tree/main/Procedimento%20proc-2-missao-4_h6F1LkBg.mp4

Listar os produtos com a chamada para o endereço seguinte:
<http://localhost:8080/CadastroEE-war/ServletProdutoFC?acao= listar>



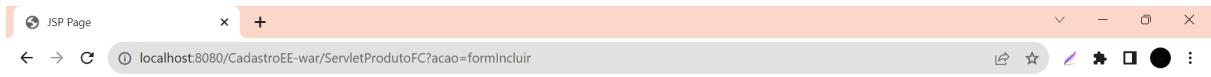
Listagem de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana	100	5.0	Alterar Excluir
3	Laranja	500	2.0	Alterar Excluir
4	Manga	800	4.0	Alterar Excluir
14	Pinha	50	50.0	Alterar Excluir



Efetuar uma inclusão a partir do link da tela de listagem



Dados do Produto

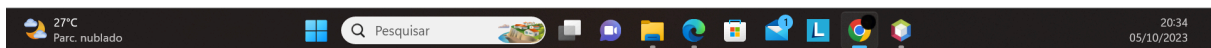
Nome: Quantidade: Preço de Venda:



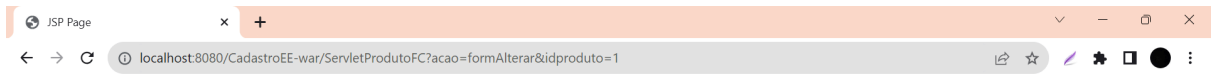
Listagem de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana	100	5.0	Alterar Excluir
3	Laranja	500	2.0	Alterar Excluir
4	Manga	800	4.0	Alterar Excluir
5	Pera	150	0.7	Alterar Excluir
14	Pinha	50	50.0	Alterar Excluir

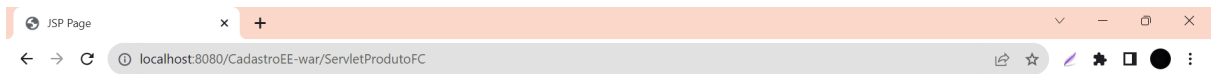


Efetuar uma alteração a partir do link dinâmico da listagem



Dados do Produto

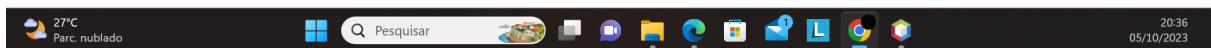
Nome: Quantidade: Preço de Venda:



Listagem de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana Pacovan	100	5.0	Alterar Excluir
3	Laranja	500	2.0	Alterar Excluir
4	Manga	800	4.0	Alterar Excluir
5	Pera	150	0.7	Alterar Excluir
14	Pinha	50	50.0	Alterar Excluir



Efetuar uma exclusão a partir do link dinâmico da listagem



Listagem de Produtos

[Novo Produto](#)

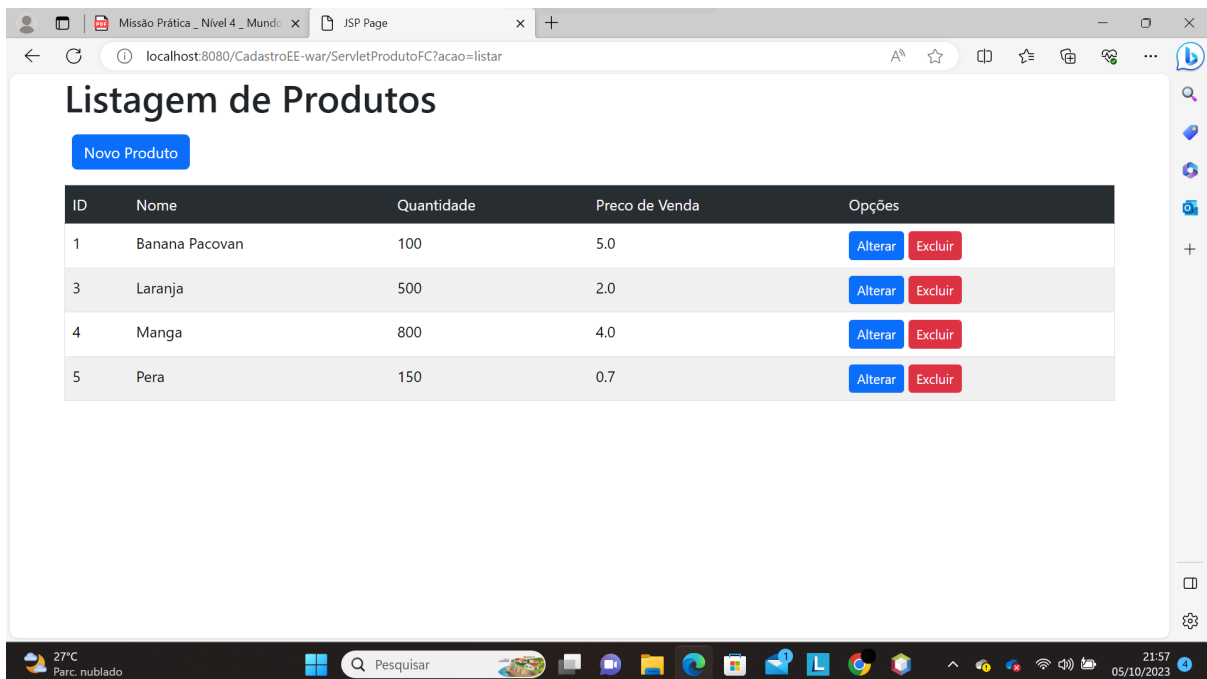
ID	Nome	Quantidade	Preço de Venda	Opções
1	Banana Pacovan	100	5.0	Alterar Excluir
3	Laranja	500	2.0	Alterar Excluir
4	Manga	800	4.0	Alterar Excluir
5	Pera	150	0.7	Alterar Excluir



Procedimento 3:

<https://github.com/joninowe/Missao-nivel-4-mundo-3/tree/main/Procedimento%203>

Modificar as características de ProdutoLista.jsp e ProdutoDados.jsp por meio das bibliotecas do framework Bootstrap



Dados do Produto

Nome:

Quantidade:

Preço de Venda:

Adicionar Produto

Dados do Produto

Nome:

Quantidade:

Preço de Venda:

Alterar Produto

Análise e Conclusão

- Como é organizado um projeto corporativo no NetBeans?
O projeto corporativo é estruturado em dois módulos: o módulo EJB, que contém as classes responsáveis pelos modelos e controles, e o módulo Web, onde estão os arquivos JSP, HTML e Servlet.
- Qual o papel das tecnologias JPA e EJB na construção de um aplicativo para a plataforma Web no ambiente Java?

JPA (Java Persistence API) é responsável por gerenciar a interação com o banco de dados de forma abstrata, permitindo a persistência de dados de maneira orientada a objetos. EJB (Enterprise JavaBeans) encapsula a lógica de negócio, processando e manipulando os dados conforme as regras definidas no sistema.

- Como o NetBeans viabiliza a melhoria de produtividade ao lidar com as tecnologias JPA e EJB?

O desenvolvedor pode focar exclusivamente em conceitos de orientação a objetos, sem a necessidade de lidar com estruturas relacionais, como tabelas, ou se preocupar com os detalhes do banco de dados.

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

Um Servlet é um objeto Java que atua como um mini-servidor, processando requisições do cliente (request) e gerando respostas (response), como uma página HTML dinâmica.

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

O Front Controller é um padrão arquitetural que atua como um controlador central, gerenciando todas as solicitações de um site e encaminhando-as para a ação apropriada. Ele é composto por duas partes principais: o Manipulador Web e a hierarquia de Comandos. O Manipulador Web é responsável por receber as requisições HTTP (POST ou GET) do servidor, extrair as informações necessárias da URL e da solicitação, determinar a ação adequada e, por fim, delegar a execução dessa ação a um objeto Comando.

- Quais as diferenças e semelhanças entre Servlets e JSPs?

Ambos oferecem alta portabilidade, facilidade de desenvolvimento, flexibilidade e escalabilidade. Em uma página JSP, a formatação HTML fica separada do código de programação, permitindo alterações sem impactar significativamente a aplicação. Já no Servlet, a lógica da aplicação e a geração da resposta estão mais integradas, tornando o código menos separado da apresentação.

- 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?

O Redirect (sendRedirect) direciona o cliente para outra página, enquanto o RequestDispatcher (forward) encaminha a requisição para outro recurso dentro do servidor. No caso do sendRedirect, o servidor responde ao cliente com um cabeçalho HTTP informando que ele deve solicitar uma nova página, fazendo com que o navegador realize essa nova requisição. Esse processo ocorre no lado do cliente. Já com o forward, a requisição do usuário é processada internamente no servidor e repassada a outro recurso, como um Servlet, que gerará a resposta final. Esse encaminhamento acontece no lado do servidor, sem que o cliente perceba a mudança. Parâmetros são dados

enviados em uma requisição HTTP, como os valores submetidos em um formulário. No método GET, esses parâmetros ficam visíveis na URL no formato: `url?param1=valor1¶m2=valor2`. Atributos, por outro lado, são objetos associados a um nome e armazenados no servidor. Eles funcionam como uma estrutura chave-valor e podem pertencer a diferentes escopos: página (PageContext), requisição (HttpServletRequest), sessão (HttpSession) ou aplicação (ServletContext).

- Como o framework Bootstrap é utilizado?
O Bootstrap é um framework CSS voltado para o desenvolvimento do front-end de aplicações web. Ele combina CSS e JavaScript para estilizar páginas e adicionar funcionalidades, utilizando classes específicas que são aplicadas diretamente às tags em um documento HTML.
- Por que o Bootstrap garante a independência estrutural do HTML?
O Bootstrap modifica a aparência da página e incorpora funcionalidades adicionais.
- Qual a relação entre o Bootstrap e a responsividade da página?
O Bootstrap tem como objetivo garantir que o site seja acessível em qualquer dispositivo, já que a responsividade é uma das suas principais características. Ele utiliza um sistema de grid que divide a tela em 12 colunas, facilitando o ajuste dos elementos de acordo com o tamanho da tela do dispositivo.