



RPG0017 - Vamos integrar sistemas

Jéssica Maria de Carvalho Matrícula: 202209187939

POLO JARDIM SÃO BERNARDO - SÃO PAULO - SP

Nível 4: Vamos integrar sistemas – 9003 – 3º

Endereço do Repositório GIT: <https://github.com/Jessicac30/Integrar-Sistemas>

Objetivo 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.

1º Procedimento | 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 jess
 */
@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="HttpServletRequest 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.
         *

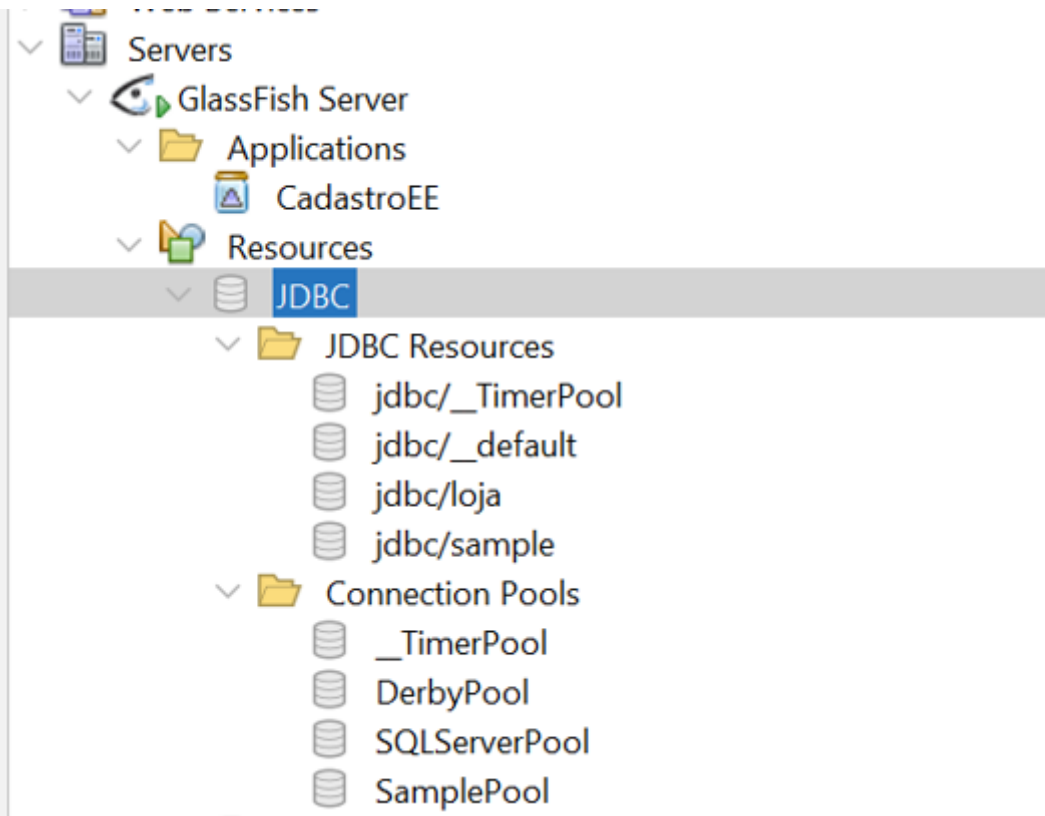
```

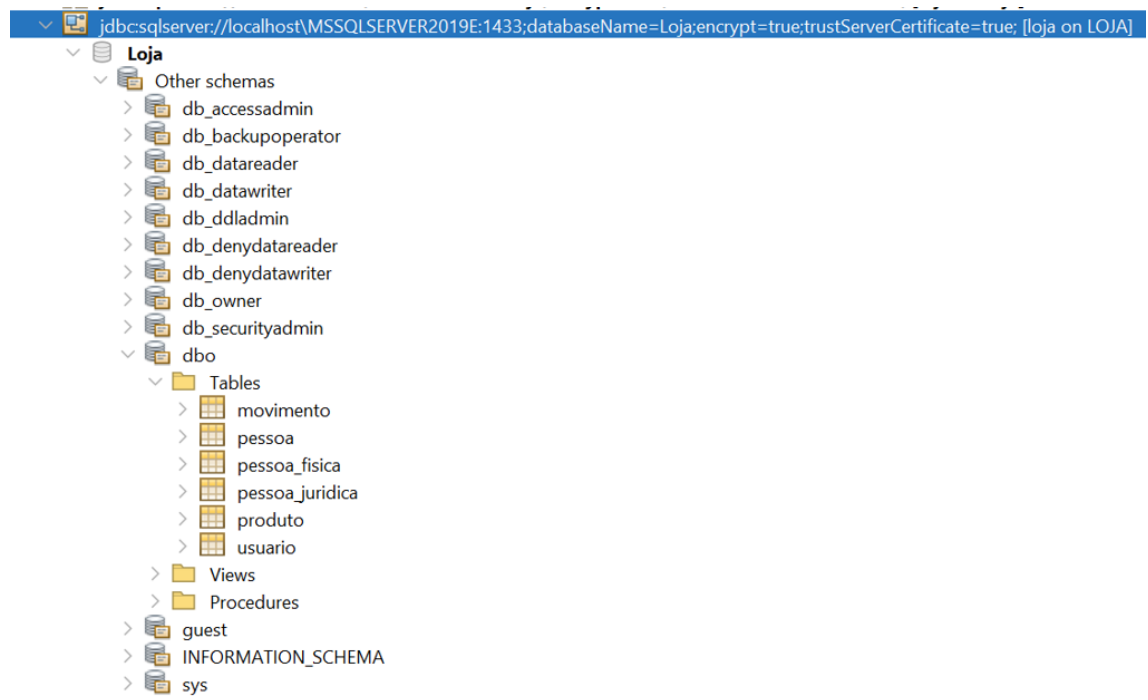
```






























    * @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);
    }

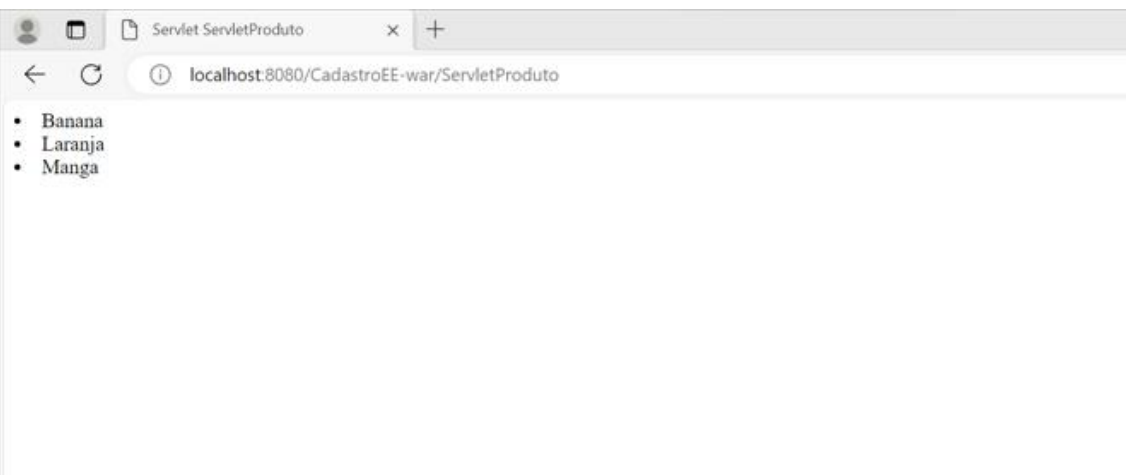
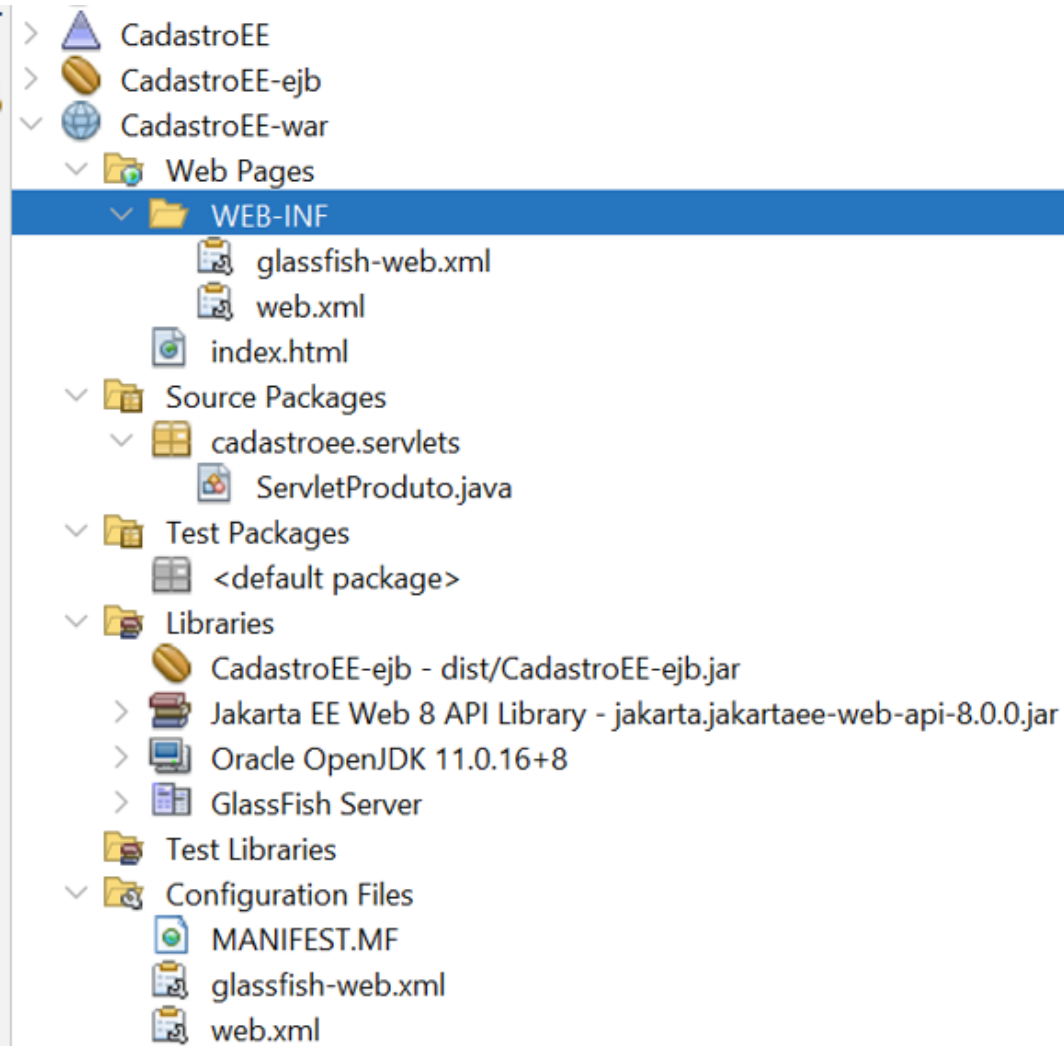
    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    } // </editor-fold>
}

```





- >  CadastroEE
- ✓  CadastroEE-ejb
 - ✓  Source Packages
 - ✓  cadastroee.controller
 -  AbstractFacade.java
 -  MovimentoFacade.java
 -  MovimentoFacadeLocal.java
 -  PessoaFacade.java
 -  PessoaFacadeLocal.java
 -  PessoaFisicaFacade.java
 -  PessoaFisicaFacadeLocal.java
 -  PessoaJuridicaFacade.java
 -  PessoaJuridicaFacadeLocal.java
 -  ProdutoFacade.java
 -  ProdutoFacadeLocal.java
 -  UsuarioFacade.java
 -  UsuarioFacadeLocal.java
 - ✓  cadastroee.model
 -  Movimento.java
 -  Pessoa.java
 -  PessoaFisica.java
 -  PessoaJuridica.java
 -  Produto.java
 -  Usuario.java
 - >  Test Packages
 - >  Libraries
 - >  Test Libraries
 - >  Enterprise Beans
 - >  Configuration Files



2º Procedimento | 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 jess
 */
@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="HttpServletRequest 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 jess
 */
@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(quantidadeProdut
o);
        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>
}

```

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

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

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

3º Procedimento | Melhorando o Design da Interface

ProdutoDados.jsp

```
<%--
    Document      : ProdutoDados
    Created on    : 25 de nov. de 2023, 18:23:12
    Author       : jess
--%>

<%@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-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/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-C6RzsynM9kWDrmNeT87bh950GNyZPhcTNXj1NW7RuBCsyN/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">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>

```

ProdutoLista.jsp

```

<%--
    Document    : ProdutoLista
    Created on   : 24 de nov. de 2023, 19:23:18
    Author      : jess
--%>

<%@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-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
        <title>JSP Page</title>
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundl
e.min.js" integrity="sha384-
C6RzsynM9kWDrmNeT87bh950GNyZPhcTNXj1NW7RuBCsyN/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>
                <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>
```

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

Dados do Produto

Nome:

Quantidade:

Preço de Venda:

Adicionar Produto

Dados do Produto

Nome:

Banana

Quantidade:

100

Preço de Venda:

5.0

Alterar Produto

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
5	Pera	150	0.7	Alterar Excluir

Análise e Conclusão:

Como é organizado um projeto corporativo no NetBeans?

Um projeto corporativo no NetBeans é estruturado de forma modular, abrangendo diferentes aspectos do desenvolvimento de aplicações empresariais. No coração deste modelo, encontram-se o projeto EAR (Enterprise Archive), que serve como um contêiner para os módulos da aplicação, o módulo EJB (Enterprise JavaBeans), responsável pela lógica de negócios e operações backend, e o módulo WAR (Web Application Archive), que contém a parte relacionada à web da aplicação, como Servlets e páginas JSP.

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

As tecnologias JPA (Java Persistence API) e EJB são fundamentais nesse contexto. A JPA facilita o mapeamento de objetos Java para tabelas de banco de dados, permitindo uma gestão de dados mais abstrata e simplificada. Por outro lado, os EJBs fornecem um framework robusto para desenvolver componentes de negócios reutilizáveis e seguros, lidando com aspectos como transações e segurança.

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

O NetBeans melhora significativamente a produtividade ao trabalhar com JPA e EJB. Ele oferece ferramentas integradas que simplificam a criação e gerenciamento de entidades JPA e EJBs. Além disso, assistentes e geradores de código ajudam na geração de código para entidades, sessões beans e outras estruturas, reduzindo a necessidade de codificação manual e acelerando o processo de desenvolvimento.

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

Os Servlets, que são componentes do lado do servidor usados para processar solicitações e gerar respostas na web, também são suportados no NetBeans. Este IDE oferece assistentes de criação para facilitar a criação de Servlets, além de ferramentas de desenvolvimento e depuração que permitem desenvolver e testar Servlets de maneira eficiente.

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

A comunicação entre Servlets e Session Beans é geralmente realizada através da injeção de dependências, uma técnica facilitada no NetBeans pelo uso de anotações como `@EJB`. Isso permite que um Servlet acesse os métodos de um EJB sem necessidade de complexas configurações JNDI, tornando a integração entre a camada web e a lógica de negócios mais direta e menos propensa a erros.

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 padrão de design que centraliza o processamento de todas as solicitações em um único ponto de controle, um "controlador frontal". Em uma aplicação web Java que segue a arquitetura MVC (Model-View-Controller), este padrão é geralmente implementado através de um Servlet. Este Servlet, atuando como o Front Controller, intercepta todas as requisições para a aplicação, gerencia o controle de fluxo, faz o processamento inicial da solicitação e, em seguida, delega responsabilidades específicas para outros componentes.

No contexto do MVC, o Front Controller é responsável por selecionar qual ação (ou controlador específico) deve ser executada com base na solicitação do usuário, e depois direciona a solicitação para a visualização apropriada (geralmente uma página JSP). Ele também pode realizar operações comuns, como autenticação, logging, e tratamento de erros.

Quais as diferenças e semelhanças entre Servlets e JSPs?

Quanto às diferenças e semelhanças entre Servlets e JSPs, ambos são tecnologias do lado do servidor usadas para desenvolver aplicações web Java, mas têm enfoques diferentes. Servlets são classes Java que permitem escrever código do lado do servidor de forma mais controlada e programática. Eles são mais adequados para lidar com lógica de negócios complexa e processamento de dados. JSPs (JavaServer Pages), por outro lado, são mais voltados para a apresentação e são mais adequados para gerar conteúdo dinâmico na web, como HTML. Eles permitem a escrita de código Java dentro do HTML, facilitando a criação de páginas web dinâmicas sem a necessidade de escrever muito código Java puro. Apesar das diferenças, Servlets e JSPs muitas vezes trabalham juntos, onde Servlets lidam com o processamento de solicitações e JSPs com a apresentação dos resultados.

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?

A diferença entre redirecionamento simples e o uso do método **forward** do **RequestDispatcher** reside em como a solicitação é tratada. Um redirecionamento (usando **response.sendRedirect(url)**) informa ao navegador do cliente para fazer uma nova solicitação para um URL diferente, resultando em duas solicitações distintas no servidor. Já o método **forward** é usado para encaminhar a solicitação atual e a resposta para outro recurso dentro do servidor, sem a necessidade de uma nova solicitação pelo navegador. É uma maneira eficiente de transferir o controle dentro do servidor.

Os parâmetros e atributos nos objetos **HttpRequest** têm papéis distintos. Os parâmetros são geralmente usados para passar dados do cliente para o servidor, como entradas de formulário. Eles são acessados usando **request.getParameter()**. Os atributos, acessados via **request.getAttribute()**, são usados para passar dados dentro do servidor, como entre Servlets e JSPs, ou para manter dados durante a vida útil de uma solicitação. Eles são úteis para compartilhar informações dentro do mesmo contexto de solicitação.

Como o framework Bootstrap é utilizado?

O framework Bootstrap é amplamente utilizado no desenvolvimento de interfaces de usuário para aplicações web. É um conjunto de ferramentas de código aberto que fornece templates baseados em HTML, CSS e JavaScript para tipografia, formulários, botões, navegação e outros componentes de interface, além de extensões de JavaScript opcionais.

Por que o Bootstrap garante a independência estrutural do HTML?

Bootstrap é projetado para facilitar o desenvolvimento de páginas web responsivas e mobile-first. Ele utiliza um sistema de grid flexível que adapta o layout da página para diferentes tamanhos de tela e dispositivos. Isso é feito através de classes CSS predefinidas que controlam o comportamento de redimensionamento e posicionamento dos elementos da página.

A independência estrutural do HTML com Bootstrap é garantida pelo uso de classes CSS que são aplicadas aos elementos HTML. Em vez de escrever estilos CSS personalizados para cada elemento, os desenvolvedores podem usar as classes do Bootstrap para estilizar rapidamente os componentes. Isso não só economiza tempo, mas também ajuda a manter a consistência em todo o projeto, uma vez que as classes do Bootstrap seguem um padrão de design uniforme.

Qual a relação entre o Bootstrap e a responsividade da página?

A responsividade é uma característica fundamental do Bootstrap. Ele emprega um layout fluido que usa unidades relativas (como percentuais) em vez de unidades fixas (como pixels). Isso, combinado com um sistema de grid responsivo e classes de utilidade para visibilidade, margem, preenchimento, etc., permite que os elementos da página se ajustem automaticamente para se adaptar a diferentes tamanhos de tela. O framework inclui pontos de interrupção CSS para diferentes larguras de tela, permitindo que os desenvolvedores especifiquem diferentes estilos para dispositivos móveis, tablets e desktops. Isso facilita a criação de sites que se parecem bem e funcionam bem em uma ampla gama de dispositivos e tamanhos de tela.