



Estácio

UNIVERSIDADE ESTÁCIO DE SÁ DE RIBEIRÃO PRETO

POLO PARQUE ANDORINHAS

DESENVOLVIMENTO FULL STACK

2023.3 FULL STACK ALUNO

NIVEL 4: VAMOS INTEGRAR SISTEMAS

ALESSANDRO SENDI SHIGEMATSU

Título da Prática

RPG0017 - Vamos integrar sistemas

Objetivos 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

Todos os códigos solicitados neste roteiro de aula

Servlet Produto java

```
package cadastroee.servlets;

import cadastroee.model.Produto;
import cadastroee.controller.ProdutoFacadeLocal;
import jakarta.ejb.EJB;
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;
import static java.lang.System.out;
import java.util.List;
```

```

public class ServletProduto extends HttpServlet {

    /**
     * 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
     */
    @EJB ProdutoFacadeLocal facade;

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");

        List<Produto> produtos = facade.findAll();

        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>");

```

```
        produtos.stream().forEach( produto -> out.println("<span>" +produto.getNome() +
"</span>") );
```

```
        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>
```

```
}
```

Resultado



Análise

- a) Como é organizado um projeto corporativo no NetBeans?

R:Aplicações corporativas (EAR), projetos de aplicativos da Web (WAR) e projetos de módulo EJB (JAR).

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

R:JPA é responsável pela persistência e EJB encapsula as regras de negócios.

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

R: Existem assistentes chamado de Wizard que facilitam o gerenciamento das tecnologia JPA e EJB, como criação de entidades, criação de servidor web, conexão com o banco.

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

R: O Servlets recebe requisições HTTP, processa-as e responde ao cliente, essa resposta pode ser um HTML, uma imagem etc. O NetBeans tem opção para que a configuração do servlets seja rápido e intuitivo, criando os arquivos relacionado ao Servlets .

2º Procedimento | Interface Cadastral com Servlet e JSPs

Todos os códigos solicitados neste roteiro de aula

ServletProdutoFC

/*

* 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.model.Produto;

import cadatroee.controller.ProdutoFacadeLocal;

import static com.oracle.wls.shaded.org.apache.xalan.xsltc.compiler.util.Type.Int;

```

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 java.util.List;
import org.glassfish.soteria.Utls;

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

    /**
     * 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
     */
    @EJB
    ProdutoFacadeLocal facade;

    @Override

```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}
```

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}
```

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
```

```
    int idProduto, quantidade;
```

```
    long valorUnitario;
```

```
    Produto produto;
```

```
    String nome;
```

```
    List<Produto> listaProdutos;
```

```
    String acao = request.getParameter("acao");
```

```
//    acao = acao == null ? "listar": acao ;
```

```
    String destino = "ProdutoLista.jsp";
```

```
    if (acao == null) acao = "listar";
```

```
    switch (acao) {
```

```
        case "listar":
```

```
            listaProdutos = facade.findAll();
```



```
request.setAttribute("listaProdutos", listaProdutos);

break;

case "incluir":

    nome = request.getParameter("nome");

    valorUnitario = Long.parseLong(request.getParameter("valor"));

    quantidade = Integer.parseInt(request.getParameter("quantidade"));


    produto = new Produto();

    produto.setNome(nome);

    produto.setPrecoVenda(valorUnitario);

    produto.setQuantidade(quantidade);


    facade.create(produto);


    listaProdutos = facade.findAll();

    request.setAttribute("listaProdutos", listaProdutos);

    break;

case "alterar":


    idProduto = Integer.parseInt(request.getParameter("id"));

    nome = request.getParameter("nome");

    valorUnitario = Long.parseLong(request.getParameter("valor"));

    quantidade = Integer.parseInt(request.getParameter("quantidade"));


    produto = facade.find(idProduto);

    produto.setNome(nome);

    produto.setPrecoVenda(valorUnitario);

    produto.setQuantidade(quantidade);


    facade.edit(produto);
```

```
        listaProdutos = facade.findAll();

        request.setAttribute("listaProdutos", listaProdutos);

        break;
    case "excluir":

        idProduto = Integer.parseInt(request.getParameter("id"));

        produto = facade.find(idProduto);

        facade.remove(produto);

        listaProdutos = facade.findAll();

        request.setAttribute("listaProdutos", listaProdutos);

        break;
    case "formIncluir":

        destino = "ProdutoDados.jsp";

        break;
    case "formAlterar":

        destino = "ProdutoDados.jsp";

        idProduto = Integer.parseInt(request.getParameter("id"));

        produto = facade.find(idProduto);

        request.setAttribute("produto", produto);

        break;
    default:

        // Ação inválida - adicione um tratamento apropriado, se necessário

        break;
}
```

```
request.getRequestDispatcher(destino).forward(request, response);
```

```
}  
}
```

ProdutoLista.jsp

```
<%--
```

Document : ProdutoLista.jsp

Created on : 25 de nov. de 2023, 02:18:30

Author : sendi

```
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<%@page import="cadastroee.model.Produto"%>
```

```
<%@page import="java.util.List"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Lista de Produtos</title>
```

```
</head>
```

```
<body>
```

```
<h1>Lista de Produtos</h1>
```

```
<td>
```

```
<a href="ServletProdutoFC?acao=formIncluir">Incluir</a>
```

```
</td>
```

```
<table border="1">
```

```
<thead>
```

```
<tr>
```

```
<th>ID</th>
```

```
<th>Nome</th>
```

```
<th>Quantidade</th>
```

```
<th>Preço</th>
```

```
<th>Editar</th>
```

```

        <th>Excluir</th>

    </tr>

</thead>

    <%for (Produto produto : (List<Produto>) request.getAttribute("listaProdutos")) {%>

        <tr>

            <!-- Dados do Produto -->

            <td><%=produto.getIdProduto()%></td>

            <td><%=produto.getNome()%></td>

            <td><%=produto.getQuantidade()%></td>

            <td><%=produto.getPrecoVenda()%></td>

            <!-- Links para Alteração e Exclusão -->

            <td>

                <a
href="ServletProdutoFC?acao=formAlterar&id=<%=produto.getIdProduto()%>">Editar</a>

            </td>

            <td>

                <a
href="ServletProdutoFC?acao=excluir&id=<%=produto.getIdProduto()%>">Excluir</a>

            </td>

        </tr>

    <%}%>

</table>

</body>

</html>

```

ProdutoDados.jsp

<%--

Document : ProdutoDados

Created on : 25 de nov. de 2023, 03:08:18

Author : sendi

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="cadastroee.model.Produto"%>

<%@page import="java.util.List"%>

<!DOCTYPE html>

<html>

<head>

<title>Cadastro de Produto</title>

</head>

<body>

<%

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

String acao = (produto == null) ? "incluir" : "alterar";

String idProduto = (produto != null) ? (String) request.getAttribute("id") : null;

%>

<h1><%= acao == "incluir" ? "Cadastro" : "Alteracao" %> de Produto</h1>

<!-- Formulário para cadastro de produto -->

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

<!-- Recuperando a entidade enviada pelo Servlet --%>

<!-- Supondo que a entidade seja um objeto 'produto' --%>

<!-- Campo oculto para o valor de 'acao' --%>

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

<!-- Campo oculto para o ID do produto (apenas quando a ação for 'alterar') --%>

<% if (acao.equals("alterar")) { %>

```
<input type="text" id="id" name="id" hidden  
value="<%=produto.getIdProduto()%>">
```

```
<% } %>
```

```
<%-- Campos para nome, quantidade e preço de venda --%>
```

```
<label for="nome">Nome:</label>
```

```
<input type="text" id="nome" name="nome" value="<%= (produto != null) ?  
produto.getNome() : "" %>"><br><br>
```

```
<label for="quantidade">Quantidade:</label>
```

```
<input type="text" id="quantidade" name="quantidade" value="<%= (produto != null) ?  
produto.getQuantidade() : "" %>"><br><br>
```

```
<label for="precoVenda">Preço de Venda:</label>
```

```
<input type="text" id="precoVenda" name="valor" value="<%= (produto != null) ?  
produto.getPrecoVenda() : "" %>"><br><br>
```

```
<%-- Botão de envio do formulário --%>
```

```
<input type="submit" value="<%= (acao.equals("incluir")) ? "incluir" : "alterar" %>">
```

```
</form>
```

```
</body>
```

```
</html>
```

Pagina ProdutoLista.jsp

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

Lista de Produtos

[Incluir](#)

| ID | Nome | Quantidade | Preço | Editar | Excluir |
|----|----------|------------|-------|------------------------|-------------------------|
| 1 | Banana | 100 | 5 | Editar | Excluir |
| 2 | Laranja | 520 | 2 | Editar | Excluir |
| 3 | Manga | 805 | 4 | Editar | Excluir |
| 5 | mexirica | 30 | 1 | Editar | Excluir |
| 6 | uva | 50 | 7 | Editar | Excluir |
| 7 | popo | 4 | 1 | Editar | Excluir |
| 8 | abacaxis | 30 | 20 | Editar | Excluir |

Excluindo item

← ↻ ⓘ localhost:8080/CadastroEE-war/ServletProdutoFC?acao=excluir&id=8

Lista de Produtos

[Incluir](#)

| ID | Nome | Quantidade | Preço | Editar | Excluir |
|----|----------|------------|-------|------------------------|-------------------------|
| 1 | Banana | 100 | 5 | Editar | Excluir |
| 2 | Laranja | 520 | 2 | Editar | Excluir |
| 3 | Manga | 805 | 4 | Editar | Excluir |
| 5 | mexirica | 30 | 1 | Editar | Excluir |
| 6 | uva | 50 | 7 | Editar | Excluir |
| 7 | popo | 4 | 1 | Editar | Excluir |

Editando Item

← ↻ ⓘ localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formAlterar&id=7

Alteracao de Produto

Nome:

Quantidade:

Preço de Venda:

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

Lista de Produtos

[Incluir](#)

| ID | Nome | Quantidade | Preço | Editar | Excluir |
|----|----------|------------|-------|------------------------|-------------------------|
| 1 | Banana | 100 | 5 | Editar | Excluir |
| 2 | Laranja | 520 | 2 | Editar | Excluir |
| 3 | Manga | 805 | 4 | Editar | Excluir |
| 5 | mexirica | 30 | 1 | Editar | Excluir |
| 6 | uva | 50 | 7 | Editar | Excluir |
| 7 | nonono | 99 | 29 | Editar | Excluir |

Inserindo item

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

Cadastro de Produto

Nome:

Quantidade:

Preço de Venda:

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

Lista de Produtos

[Incluir](#)

| ID | Nome | Quantidade | Preço | Editar | Excluir |
|----|--------------|------------|-------|------------------------|-------------------------|
| 1 | Banana | 100 | 5 | Editar | Excluir |
| 2 | Laranja | 520 | 2 | Editar | Excluir |
| 3 | Manga | 805 | 4 | Editar | Excluir |
| 5 | mexirica | 30 | 1 | Editar | Excluir |
| 6 | uva | 50 | 7 | Editar | Excluir |
| 7 | nonono | 99 | 29 | Editar | Excluir |
| 9 | yesyesyseyes | 100 | 20 | Editar | Excluir |

Análise

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

R: Um controle que cuida de todas as requisições do WEB SITE, que delega requisições para recursos subsequentes, fica a frente da aplicação WEB. É implementado através do javax.servlet-ap .

b) Quais as diferenças e semelhanças entre Servlets e JSPs?

R: Diferenças:

- 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 JSP geralmente é usado quando há menos envolvimento de processamento de dados.
- Os servlets rodam mais rápido que o JSP, por outro lado, o JSP roda mais devagar 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. Já em JSP, a lógica de negócios é separada da lógica de apresentação usando JavaBeans.

Semelhanças:

- São programas Java executados em um servidor de aplicativos Java e que estendem os recursos do servidor da web.

c) 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`?

R: A diferença está entre *redirecionar o cliente para uma página* (`sendRedirect`) e *encaminhar 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¶m2=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`).

3º Procedimento | Melhorando o Design da Interface

Todos os códigos solicitados neste roteiro de aula

ProdutoLista

<%--

Document : ProdutoLista.jsp

Created on : 25 de nov. de 2023, 02:18:30

Author : sendi

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="cadastroee.model.Produto"%>

<%@page import="java.util.List"%>

<!DOCTYPE html>

<html>

<head>

<title>Lista de Produtos</title>

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

</head>

<body class="container">

<h1>Lista de Produtos</h1>

<td>

Novo Produto

</td>

<table class="table table-striped">

<thead class="table-dark">

<tr>

<th>#</th>

<th>Nome</th>

```

        <th>Quantidade</th>

        <th>Preço de Venda</th>

        <th>Opções</th>

    </tr>

</thead>

<%for (Produto produto : (List<Produto>) request.getAttribute("listaProdutos")) {%>

    <tr>

        <!-- Dados do Produto -->

        <td><%=produto.getIdProduto()%></td>

        <td><%=produto.getNome()%></td>

        <td><%=produto.getQuantidade()%></td>

        <td><%=produto.getPrecoVenda()%></td>

        <!-- Links para Alteração e Exclusão -->

        <td>

            <a class="btn btn-primary m-2"
href="ServletProdutoFC?acao=formAlterar&id=<%=produto.getIdProduto()%>">Alterar</a>

            <a class="btn btn-danger m-2"
href="ServletProdutoFC?acao=excluir&id=<%=produto.getIdProduto()%>">Excluir</a>

        </td>

    </tr>

<%}%>

</table>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrmNEt87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>

```

</body>

</html>

Produto Dados

<%--

Document : ProdutoDados

Created on : 25 de nov. de 2023, 03:08:18

Author : sendi

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="cadastroee.model.Produto"%>

<%@page import="java.util.List"%>

<!DOCTYPE html>

<html>

<head>

<title>Cadastro de Produto</title>

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

</head>

<body class="container">

<%

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

String acao = (produto == null) ? "incluir" : "alterar";

String idProduto = (produto != null) ? (String) request.getAttribute("id") : null;

%>

<h1><%= acao == "incluir" ? "Cadastro" : "Alteracao" %> de Produto</h1>

<!-- Formulário para cadastro de produto -->

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

```
<input type="hidden" id="acao" name="acao" value="<%=acao%>">
```

```
<% if (acao.equals("alterar")) { %>
```

```
    <input type="text" id="id" name="id" hidden value="<%=produto.getIdProduto()%>">
```

```
<% } %>
```

```
<div class="mb-3">
```

```
    <label class="form-label" for="nome">Nome:</label>
```

```
    <input class="form-control" type="text" id="nome" name="nome" value="<%= (produto  
!= null) ? produto.getNome() : "" %>">
```

```
</div>
```

```
<div class="mb-3">
```

```
    <label class="form-label" for="quantidade">Quantidade:</label>
```

```
    <input class="form-control" type="text" id="quantidade" name="quantidade"  
value="<%= (produto != null) ? produto.getQuantidade() : "" %>">
```

```
</div>
```

```
<div class="mb-3">
```

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

```
    <input class="form-control" type="text" id="precoVenda" name="valor" value="<%=  
(produto != null) ? produto.getPrecoVenda() : "" %>">
```

```
</div>
```

```
    <input class="btn btn-primary" type="submit" value="<%= (acao.equals("incluir")) ?  
"Incluir Produto" : "Alterar Produto" %>">
```

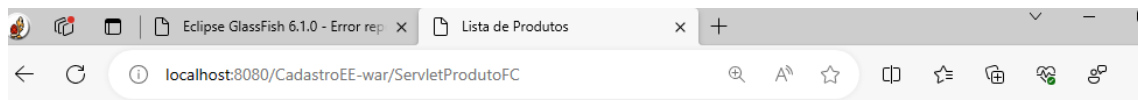
```
</form>
```

```
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"  
integrity="sha384-  
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0JlpcV8Qyq46cDfL"  
crossorigin="anonymous"></script>
```

```
</body>
```

```
</html>
```

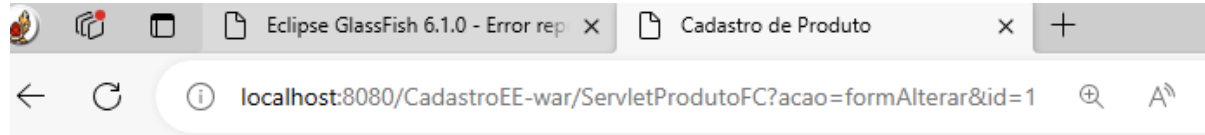
Página ProdutoLista



Lista de Produtos

Novo Produto

#	Nome	Quantidade	Preço de Venda	Opções	
1	Banana	100	5	Alterar	Excluir
2	Laranja	520	2	Alterar	Excluir
3	Manga	805	4	Alterar	Excluir
5	mexirica	30	1	Alterar	Excluir
6	uva	50	7	Alterar	Excluir
7	nonono	99	29	Alterar	Excluir



Alteracao de Produto

Nome:

Banana

Quantidade:

100

Preço de Venda:

5

Alterar Produto

Análise

a) Como o framework Bootstrap é utilizado?

R: Ele é utilizado no jsp e inserido na página como um CSS qualquer através de cdn ou localmente.

b) Por que o Bootstrap garante a independência estrutural do HTML?

R: O Bootstrap trabalha com a estilização do elemento <div> e com a utilização da classe container.

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

R: O objetivo do framework é oferecer uma experiência mais agradável ao usuário durante a utilização do site e permitir a construção de sites responsivos. Isso significa que as páginas são desenvolvidas para funcionar em desktops, tablets e smartphones, de uma forma simples e bem organizada.