



## **Missão Prática do Nível 4 - Vamos integrar sistemas**

**Lorena Rosa Borges Sanches - 202204376067**

**Campus Uberlândia/MG**

**Vamos integrar sistemas – 2023.2 – 3º Semestre**

**Desenvolvimento Full Stack**

**Link do repositório Git: [github.com/LorenaBorgesSanches/Mundo3-Pratica4](https://github.com/LorenaBorgesSanches/Mundo3-Pratica4)**

### **Objetivo da Prática**

Trata-se de material de apoio e complemento para o trabalho prático desenvolvido como requisito de aprovação no curso de Desenvolvimento Full Stack, do 3º semestre da instituição de ensino Estácio.

O trabalho é dividido em três procedimentos, e nos três procedimentos teremos a mesma organização de conteúdo. Sendo a primeira parte do conteúdo referente a prints dos códigos e seus respectivos resultados, e após, teremos perguntas elaboradas pela instituição de ensino seguidas de respostas.

### **1) 1º Procedimento | Camadas de Persistência e Controle**

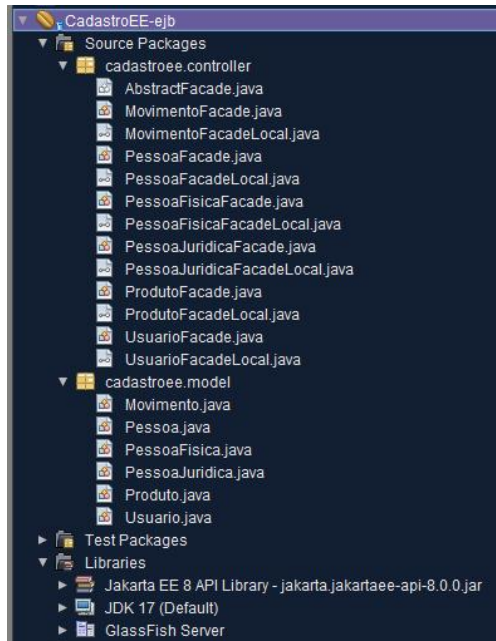
Conexão com o banco:



Ping na SqlServerPool:

```
asadmin> ping-connection-pool SQLServerPool  
Command ping-connection-pool executed successfully.
```

Criação de model e controllers no projeto ejb:



ServletProduto:

```
33 @EJB ProdutoFacadeLocal facade;
34 protected void processRequest(HttpServletRequest request, HttpServletResponse response)
35     throws ServletException, IOException {
36
37     response.setContentType("text/html;charset=UTF-8");
38     try (PrintWriter out = response.getWriter()) {
39         /* TODO output your page here. You may use following sample code. */
40         out.println("<!DOCTYPE html>");
41         out.println("<html>");
42         out.println("<head>");
43         out.println("<title>Servlet ServletProduto</title>");
44         out.println("</head>");
45         out.println("<body>");
46         out.println("<h1>Servlet ServletProduto at " + request.getContextPath() + "</h1>");
47
48         out.println("<h2>Produtos</h2>");
49         for (Produto produto : facade.findAll()) {
50             out.println("<p>" + produto.getNome() + "</p>");
51         }
52         out.println("</body>");
53         out.println("</html>");
54     }
55 }
```

Web.xml:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="4.0" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd">
3     <servlet>
4         <servlet-name>ServletProduto</servlet-name>
5         <servlet-class>cadastroee.servlets.ServletProduto</servlet-class>
6     </servlet>
7     <servlet>
8         <servlet-name>ServletProdutoFC</servlet-name>
9         <servlet-class>cadastroee.servlets.ServletProdutoFC</servlet-class>
10    </servlet>
11    <servlet-mapping>
12        <servlet-name>ServletProduto</servlet-name>
13        <url-pattern>/ServletProduto</url-pattern>
14    </servlet-mapping>
15    <servlet-mapping>
16        <servlet-name>ServletProdutoFC</servlet-name>
17        <url-pattern>/ServletProdutoFC</url-pattern>
18    </servlet-mapping>
19    <session-config>
20        <session-timeout>30</session-timeout>
21    </session-config>
22 </web-app>
23
```

Execução do procedimento1:



## 2) 2º Procedimento | Interface Cadastral com Servlet e JSPs

## 3) 3º Procedimento | Melhorando o Design da Interface

Obs: Prints já com a inclusão do bootstrap

ServletProdutoFC:

```
31 | @EJB ProdutoFacadeLocal facade;
32 | protected void processRequest(HttpServletRequest request, HttpServletResponse response)
33 |     throws ServletException, IOException {
34 |
35 |     String acao = request.getParameter(name:"acao");|
36 |
37 |     if (acao == null) {
38 |         acao = "listar";
39 |     }
40 |
41 |     String destino = "ProdutoLista.jsp";
42 |
43 |     switch (acao) {
44 |         case "listar" ->
45 |             request.setAttribute(name:"lista", o: facade.findAll());
46 |         case "incluir" -> {
47 |             String nome = request.getParameter(name:"nome");
48 |             String quantidade = request.getParameter(name:"quantidade");
49 |             String preco = request.getParameter(name:"preco");
50 |
51 |             Produto produto = new Produto();
52 |             produto.setNome(nome);
53 |             produto.setPrecoVenda(precoVenda: Float.parseFloat(s: preco));
54 |             produto.setQuantidade(quantidade: Integer.parseInt(s: quantidade));
55 |
56 |             facade.create(produto);
57 |
58 |             request.setAttribute(name:"lista", o: facade.findAll());
59 |         }
60 |         case "alterar" -> {
61 |             String id = request.getParameter(name:"id");
62 |             String nome = request.getParameter(name:"nome");
63 |             String quantidade = request.getParameter(name:"quantidade");
64 |             String preco = request.getParameter(name:"preco");
65 |
66 |             Produto produto = facade.find(id: Integer.valueOf(s: id));
67 |
68 |             produto.setNome(nome);
69 |             produto.setPrecoVenda(precoVenda: Float.parseFloat(s: preco));
70 |             produto.setQuantidade(quantidade: Integer.parseInt(s: quantidade));
71 |
72 |             facade.edit(produto);
73 |             request.setAttribute(name:"lista", o: facade.findAll());
74 |         }
75 |         case "excluir" -> {
76 |             String id = request.getParameter(name:"id");
77 |             facade.remove(produto: facade.find(id: Integer.valueOf(s: id)));
78 |             request.setAttribute(name:"lista", o: facade.findAll());
79 |         }
80 |
81 |         case "formIncluir" ->
82 |             destino = "ProdutoDados.jsp";
83 |
84 |         case "formAlterar" -> {
85 |             String id = request.getParameter(name:"id");
86 |             request.setAttribute(name:"produto", o: facade.find(id: Integer.valueOf(s: id)));
87 |             destino = "ProdutoDados.jsp";
88 |         }
89 |     }
90 |     request.getRequestDispatcher(path:destino).forward(request, response);
91 | }
92 |
93 | HttpServlet methods. Click on the + sign on the left to edit the code.
```

## ProdutoLista:

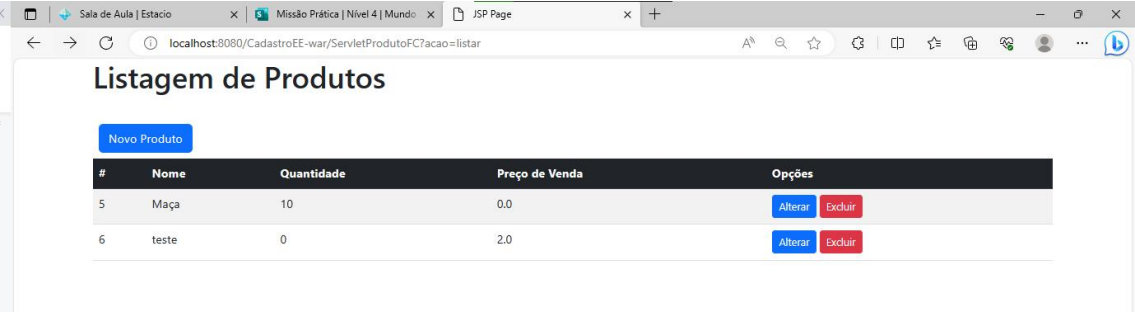
```
7 <%@page import="cadastroee.model.Produto"%>
8 <%@page import="java.util.List"%>
9 <%@page contentType="text/html" pageEncoding="UTF-8"%>
10
11 <!DOCTYPE html>
12 <html>
13 <head>
14 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
15 <title>JSP Page</title>
16 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4bw4/aepP/VC"
17 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-Hmwvtg8No3bZJJLVd8oVXjr"
18 </head>
19 <body class="container">
20 <h1>Listagem de Produtos</h1>
21 <br>
22 <a class="btn btn-primary m-2" href="ServletProdutoFC?acao=formIncluir">Novo Produto</a>
23 <br>
24 <table class="table table-striped">
25 <thead class="table-dark">
26 <tr>
27 <th></th>
28 <th>Nome</th>
29 <th>Quantidade</th>
30 <th>Preço de Venda</th>
31 <th>Opções</th>
32 </tr>
33 </thead>
34 <tbody>
35 <%
36 List<Produto> lista = (List<Produto>) request.getAttribute("lista");
37 for (Produto p : lista) {
38 %>
39 <tr>
40 <td><%=p.getIdProduto()%></td>
41 <td><%=p.getNome()%></td>
42 <td><%=p.getQuantidade()%></td>
43 <td><%=p.getPrecoVenda()%></td>
44 <td>
45 <a class="btn btn-primary btn-sm" href="ServletProdutoFC?acao=formAlterar&id=<%=p.getIdProduto()%>">Alterar</a>
46 <a class="btn btn-danger btn-sm" href="ServletProdutoFC?acao=excluir&id=<%=p.getIdProduto()%>">Excluir</a>
47 </td>
48 </tr>
49 <% %>
50 </tbody>
51 </table>
52 </body>
53 </html>
```

## ProdutoDados:

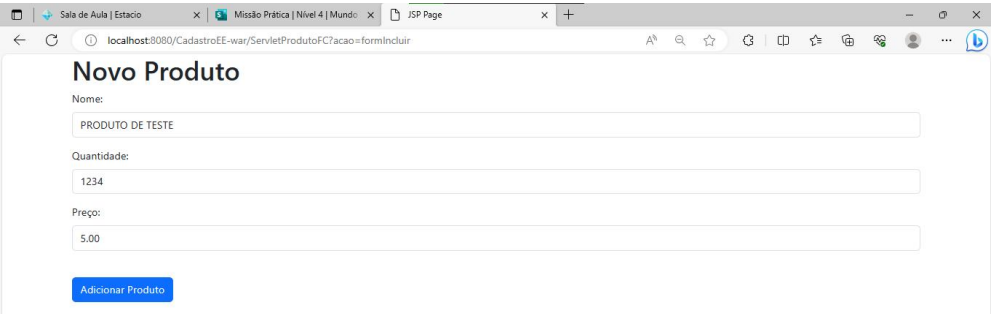
```
7 <%@page import="cadastroee.model.Produto"%>
8 <%@page contentType="text/html" pageEncoding="UTF-8"%>
9 <!DOCTYPE html>
10 <html>
11 <head>
12 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13 <title>JSP Page</title>
14 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-4bw4/aepP/VC"
15 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-Hmwvtg8No3bZJJLVd8oVXjr"
16 </head>
17 <body class="container">
18 <form class="form" action="ServletProdutoFC" method="post">
19 <%
20 Produto produto = (Produto) request.getAttribute("produto");
21 String acao = produto == null ? "incluir" : "alterar";
22 %>
23 <h1><%=(acao == "incluir") ? "Novo" : "Dados do"%> Produto</h1>
24 <br>
25 <input type="hidden" name="acao" value="<%=acao%>"/>
26 <br>
27 <% if (acao == "alterar") {%>
28 <input type="hidden" name="id" value="<%=produto.getIdProduto()%>"/>
29 <%}%>
30 <br>
31 <div class="mb-3">
32 <label class="form-label">Nome:</label>
33 <input class="form-control" type="text" name="nome" value="<%=(acao == "alterar") ? produto.getNome() : ""%>"/>
34 </div>
35 <div class="mb-3">
36 <label class="form-label">Quantidade:</label>
37 <input class="form-control" type="text" name="quantidade" value="<%=(acao == "alterar") ? produto.getQuantidade() : ""%>"/>
38 </div>
39 <div class="mb-3">
40 <label class="form-label">Preço:</label>
41 <input class="form-control" type="text" name="preco" value="<%=(acao == "alterar") ? produto.getPrecoVenda() : ""%>"/>
42 </div>
43 <br>
44 <input class="btn btn-primary" type="submit" value="<%=(acao == "incluir") ? "Adicionar" : "Alterar"%> Produto"/>
45 </form>
46 </body>
47 </html>
```

Execução do procedimento 2 e 3:

Lista:



Novo produto:



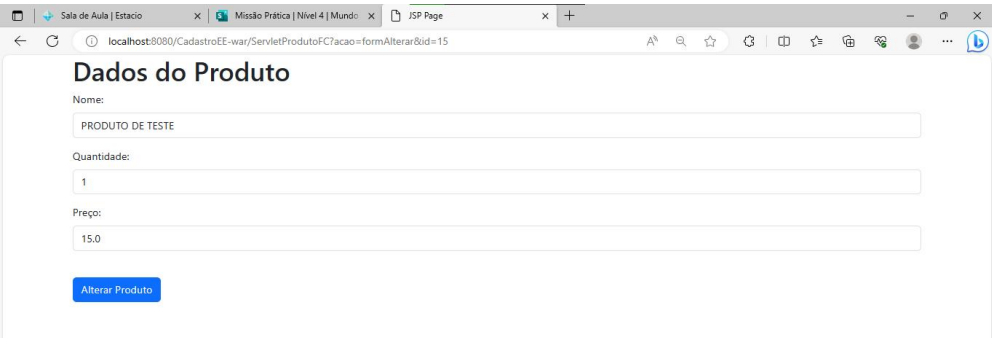
Produto adicionado:

## Listagem de Produtos

[Novo Produto](#)

#	Nome	Quantidade	Preço de Venda	Opções
5	Maça	10	0.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
6	teste	0	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
15	PRODUTO DE TESTE	1234	5.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

Edição de produto:



Produto Alterado:

## Listagem de Produtos

Novo Produto

#	Nome	Quantidade	Preço de Venda	Opções
5	Maça	10	0.0	<div>AlterarExcluir</div>
6	teste	0	2.0	<div>AlterarExcluir</div>
15	PRODUTO DE TESTE	1	15.0	<div>AlterarExcluir</div>

Produto Excluído: (ver url)

Novo Produto

#	Nome	Quantidade	Preço de Venda	Opções
5	Maça	10	0.0	<div>AlterarExcluir</div>
6	teste	0	2.0	<div>AlterarExcluir</div>

### **1.1) Relatório discente de acompanhamento**

Análise e Conclusão:

a) Como é organizado um projeto corporativo no NetBeans?

Um projeto corporativo Java no NetBeans (Enterprise Application) é organizado em regras de negócios (EJBs), classes que conectam e acessam o banco de dados (estruturado pelo DAO) e a camada de interface web.

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

O papel de ambas as tecnologias visam tornar o desenvolvimento mais ágil, o código reutilizável, de fácil compreensão, modular e seguro. A JPA (Java Persistence API) Java tem como função persistir dados de forma orientada a objetos. Ela realiza o mapeamento de objetos em bancos de dados relacionais, através de anotações, possibilitando a execução de operações CRUD de forma mais simples, pois não precisa desenvolver o código SQL. Além disso, com o mapeamento, são criadas as classes que representam as entidades no banco de dados e informações relevantes das mesmas. Também tem a função de gerenciar essas entidades, atualizando todas as operações que elas vierem a sofrer.

Já o EJB (Enterprise JavaBeans) é responsável pela implementação das regras de negócios e suas operações. Portanto, possui funções de encapsulamento das regras de negócios, o que torna o código organizado e reutilizável; possui autenticação e controle de acesso, tornando-o seguro para ambientes empresariais; gerencia as operações de negócios e mantém a integridade das informações;

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

O NetBeans é uma IDE que suporta o desenvolvimento de aplicativos Java, e possui algumas facilidades para melhorar a produtividade no processo de desenvolvimento e conexão. O que achei de grande ajuda, foi o geramento automático de código para a conexão (a partir do fornecimento de informações), e a automatização das classes e métodos para realizar o CRUD. Além disso, ele realiza a refatoração, depuração e integração com os servidores (ex: GlassFish).



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

O Servlet é um objeto Java, que estende as funções do servidor que esta sendo utilizado. Sua função é lidar com as solicitações HTTP, e devolver uma resposta ao servidor solicitante. Portanto, ele faz parte da camada de apresentação num projeto web.

O NetBeans oferece suporte na construção de Servlets na integração com os servidores web, no desenvolvimento do código fazendo sugestões para melhor orientação, na realização de depuração, entre outros aspectos. Portanto, o NetBeans em tese é um bom ambiente de desenvolvimento de projetos web, que possui ferramentas que configuram e auxilia na jornada de construção do aplicativo.

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

A comunicação entre os Servlets e os Session Beans é baseada na injeção de dependência (acesso de forma direta e automática). Ou seja, o Servlets interagem com o Session Beans injetado e através dessa relação, pode realizar operações de negócios e retornar resultados. Dessa forma, ficam separadas as camadas de apresentação das operações de negócios.

#### **4) 2º Procedimento | Interface Cadastral com Servlet e JSPs**

##### **2.1) Relatório discente de acompanhamento**

Análise e Conclusão:

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

O padrão Front Controller tem como objetivo concentrar as requisições em um único ponto e a partir daí ele direciona as demandas aos controladores. A implementação do padrão Front Controller no aplicativo Web Java, deve ocorrer na camada view da arquitetura MVC, uma vez que lida somente com o fluxo de chamadas e redirecionamento.

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

A semelhança marcante de ambas tecnologias se dá no fato de que são utilizadas para criar aplicações web dinâmicas. Além disso, ambas fazem processamento de requisições HTTP enviadas pelo navegador (cliente) e retornam as respectivas respostas e também interagem com a lógica de negócios e tem acesso ao banco de dados.

No entanto, o fazem de forma distinta. Enquanto os Servlets processam as requisições e as respondem, ordinariamente através de XML e JSON, e portanto, mesclam os códigos referente a lógica de negócios com apresentação, os JPSs são em sua essência páginas HTML que possuem código Java, e por isso, são mais fortes no quesito de facilidade de desenvolvimento por ser HTML e na manipulação da aparência.

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?

A diferença entre o redirecionamento simples(método sendRedirect) e o método forward se dá porque, o primeiro envia ao navegador uma resposta HTTP com status de redirecionamento e gera uma nova requisição. Enquanto o segundo, executa o redirecionamento internamente e não realiza uma nova requisição HTTP.

## **5) 3º Procedimento | Melhorando o Design da Interface**

### **3.1) Relatório discente de acompanhamento**

Análise e Conclusão:

a) Como o framework Bootstrap é utilizado?

O Bootstrap é um framework de front end que possui interfaces pré construídas(modelos customizáveis). Agilizam o desenvolvimento do front end, e colabora para que o desenvolvedor se concentre na lógica do negócio. Possui documentações e guias para sua utilização, é personalizável, contém inúmeras opções de estilos e tipografias e suas alterações são feitas através de HTML e CSS. Para utilizar basta incluir os arquivos CSS/HTML e JavaScript no projeto desenvolvido. Ele atua através da injeção de dependência e inversão de controle.

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

Sobre essa questão em específico, não encontrei conteúdo relevante a respeito. E se entendi corretamente a pergunta, a resposta no caso é que o Bootstrap não garante a independência estrutural do HTML, uma vez que suas interfaces são criadas a partir da estrutura de HTML. No entanto, para implementação do mesmo no projeto não há necessidade de conhecimento em HTML, uma vez que o arquivo vem pronto e no máximo o desenvolvedor precisa fazer pequenas alterações em seu fonte, e esse arquivo não se mescla com os outros. Fica separado e por isso de fácil manutenção. No site oficial do Bootstrap tem a exigência de que se use o HTML5 para utilizar o framework.

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

O Bootstrap possui como padrão a responsividade em suas interfaces e por isso se adaptam aos diferentes tipos de tela dos usuários.