



Missão Prática | Nível 4 | Mundo 3

Italo Augusto Juliano Barbosa - 202303617674

Campus Aparecida de Goiânia

Vamos Integrar Sistemas – Turma: 9001 – 3º Semestre

Link do repositório no GitHub: <https://github.com/Anarquia122/trabalho-facul-CadastroEE>

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.

OBS: devido a ocorrência de vários erros precisei trocar as tecnologias para finalizar o trabalho.

IDE: Eclipse

SGBD: PostgreSQL

Servidor: WildFly 27

1º Procedimento | Camadas de Persistência e Controle

Produto.java:

```
package cadastroee.model;
```

```
import java.io.Serializable;

import jakarta.persistence.*;

import java.math.BigDecimal;

import java.util.List;
```

```
/**
```

```
 * The persistent class for the produto database table.
```

```
 *
```

```
 */
```

```
@Entity
```

```
@NamedQuery(name="Produto.findAll", query="SELECT p FROM Produto p")
```

```
public class Produto implements Serializable {
```

```
    private static final long serialVersionUID = 1L;
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"produto_seq")
```

```
    @SequenceGenerator(name = "produto_seq", sequenceName = "produto_seq",
allocationSize = 1)
```

```
    private Long idproduto;
```

```
    private String nomeproduto;
```

```
    private Float preco;
```

```
private Integer quantidade;

//bi-directional many-to-one association to Movimento
@OneToMany(mappedBy="produto")
private List<Movimento> movimentos;

public Produto() {
}

public Long getIdproduto() {
return this.idproduto;
}

public void setIdproduto(Long idproduto) {
this.idproduto = idproduto;
}

public String getNomeproduto() {
return this.nomeproduto;
}

public void setNomeproduto(String nomeproduto) {
this.nomeproduto = nomeproduto;
}
```

```
public Float getPreco() {  
    return this.preco;  
}
```

```
public void setPreco(Float preco) {  
    this.preco = preco;  
}
```

```
public Integer getQuantidade() {  
    return this.quantidade;  
}
```

```
public void setQuantidade(Integer quantidade) {  
    this.quantidade = quantidade;  
}
```

```
public List<Movimento> getMovimentos() {  
    return this.movimentos;  
}
```

```
public void setMovimentos(List<Movimento> movimentos) {  
    this.movimentos = movimentos;  
}
```

```
public Movimento addMovimento(Movimento movimento) {  
    getMovimentos().add(movimento);  
}
```

```

        movimento.setProduto(this);

        return movimento;
    }

    public Movimento removeMovimento(Movimento movimento) {
        getMovimentos().remove(movimento);
        movimento.setProduto(null);

        return movimento;
    }
}

```

ServletProduto.java:

```

package cadastroee.servlets;

import java.io.IOException;
import java.util.List;
import jakarta.ejb.EJB;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import cadastroee.controller.ProdutoFacadeLocal;
import cadastroee.model.Produto;

```

```

/**
 * Servlet implementation class ServletProduto
 */
public class ServletProduto extends HttpServlet {

    private static final long serialVersionUID = 1L;

    @EJB
    private ProdutoFacadeLocal facade;

    public ServletProduto() {

        // TODO Auto-generated constructor stub

    }

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

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

        response.setContentType("text/html");

        response.getWriter().append("<html><body><h1>Lista          de
Produtos</h1><ul>");

        for (Produto produto : produtos) {

            response.getWriter().append("<li>").append(produto.getNomeproduto() + " - " +
produto.getPreco()).append("</li>");

        }

        response.getWriter().append("</ul></body></html>");
    }
}

```

```
}

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

    // TODO Auto-generated method stub

    doGet(request, response);

}

}
```

Análise e Conclusão:

A. Como é organizado um projeto corporativo no Netbeans?

É estruturado em módulos com pacotes para EJBs, entidades JPA, Servlets e recursos web.

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

Automatizar a criação de classes ligadas ao banco de dados.

C. Como o Netbeans viabiliza a melhoria da produtividade ao lidar com as tecnologias JPA e EJB?

Gerando automaticamente os códigos e configurações.

D. O que são Servlets e como o Netbeans oferece suporte à construção desse tipo de componente em um projeto Web?

Os Servlets são componentes Java que processam requisições HTTP. O Netbeans oferece ferramentas para criar, configurar e depurar os Servlets.

E. Como é feita a comunicação entre os Servlets e o SessionBeans do pool de EJBs?

Através da injeção de dependências, usando a anotação @EJB nos Servlets para conseguir usar métodos EJBs.

2º Procedimento | Interface Cadastral com Servlet e JSPs

ServletProdutoFC.java:

```
package cadastroee.servlets;
```

```
import jakarta.ejb.EJB;
```

```
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.io.IOException;
```

```
import java.util.List;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
import cadastroee.model.Produto;
```

```
import cadastroee.controller.ProdutoFacadeLocal;
```

```
//http://localhost:8080/CadastroEE-3-war/ServletProdutoFC?acao=listar
```

```
@WebServlet(name = "ServletProdutoFC", urlPatterns = {"/ServletProdutoFC"})
```

```
public class ServletProdutoFC extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```


@EJB

private ProdutoFacadeLocal facade;

private static final Logger logger =
Logger.getLogger(ServletProdutoFC.class.getName());

protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

throws ServletException, IOException {

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

String destino = "";

if (acao != null) {

switch (acao) {

case "listar": {

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

logger.log(Level.INFO, "Número de produtos encontrados: {0}",
produtos.size());

request.setAttribute("produtos", produtos);

destino = "ProdutoLista.jsp";

break;

}

case "formAlterar": {

Long idAlterar = Long.parseLong(request.getParameter("id"));

Produto produtoAlterar = facade.find(idAlterar);

```
        request.setAttribute("produto", produtoAlterar);

        destino = "ProdutoDados.jsp";

        break;
    }

    case "excluir": {

        Long idExcluir = Long.parseLong(request.getParameter("id"));

        Produto produtoExcluir = facade.find(idExcluir);

        facade.remove(produtoExcluir);

        request.setAttribute("produtos", facade.findAll());

        destino = "ProdutoLista.jsp";

        break;
    }

    case "alterar": {

        Long id = Long.parseLong(request.getParameter("id"));

        Produto produto = facade.find(id);

        produto.setNomeproduto(request.getParameter("nome"));

        produto.setPreco(Float.parseFloat(request.getParameter("preco")));

        produto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));

        facade.edit(produto);

        request.setAttribute("produtos", facade.findAll());

        destino = "ProdutoLista.jsp";

        break;
    }

    case "formIncluir": {

        destino = "ProdutoDados.jsp";
```

```

        break;
    }
    case "incluir": {
        Produto novoProduto = new Produto();
        novoProduto.setNomeproduto(request.getParameter("nome"));
        novoProduto.setPreco(Float.parseFloat(request.getParameter("preco")));

        novoProduto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));

        facade.create(novoProduto);

        request.setAttribute("produtos", facade.findAll());

        destino = "ProdutoLista.jsp";

        break;
    }
    default: {
        destino = "erro.jsp";

        break;
    }
}

} else {
    destino = "erro.jsp";
}

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

public ServletProdutoFC() {

```

```

        super();
    }

    @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);
    }
}

```

ProdutoLista.jsp:

```

<%@ page contentType="text/html; charset=UTF-8" language="java"
pageEncoding="UTF-8"%>

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

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

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Lista de Produtos</title>

```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEw
IH" crossorigin="anonymous">
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
```

```
</head>
```

```
<body class="container">
```

```
<h1>Listagem de Produtos</h1>
```

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

```
<br><br>
```

```
<table border="1" class="table table-striped">
```

```
<thead class="table-dark">
```

```
<tr>
```

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

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

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

```
<th>Preço</th>
```

```
<th>Ações</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<%
```

```

List<Produto> produtos = (List<Produto>) request.getAttribute("produtos");

if (produtos != null) {
    if (!produtos.isEmpty()) {
        for (Produto produto : produtos) {
            %>
        <tr>
            <td><%= produto.getIdproduto() %></td>
            <td><%= produto.getNomeproduto() %></td>
            <td><%= produto.getQuantidade() %></td>
            <td><%= produto.getPreco() %></td>
            <td>
                <a
                    class="btn
                        btn-primary
                        btn-sm"
                    href="ServletProdutoFC?acao=formAlterar&id=<%=
                        produto.getIdproduto()
                    %>">Alterar</a>
                <a
                    class="btn
                        btn-danger
                        btn-sm"
                    href="ServletProdutoFC?acao=excluir&id=<%=
                        produto.getIdproduto()
                    %>">Excluir</a>
            </td>
        </tr>
    <%
    }
    } else {
        %>
    <tr>
        <td colspan="5">Nenhum produto encontrado.</td>
    </tr>
    <%

```

```

        }

    } else {

    %>

    <tr>

        <td colspan="5">Nenhum produto encontrado.</td>

    </tr>

    <%

        }

    %>

</tbody>

</table>

</body>

</html>

```

ProdutoDados.jsp:

```

<%@      page      contentType="text/html; charset=UTF-8"      language="java"
pageEncoding="UTF-8"%>

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

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Dados do Produto</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet"      integrity="sha384-

```

QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>

</head>

<body class="container">

<h1>Dados do Produto</h1>

<%

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

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

%>

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

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

<% if (produto != null) { %>

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

<% } %>

<div class="mb-3">

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

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

</div>

<div class="mb-3">

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


```

        <input      class="form-control"      type="number"      name="quantidade"
id="quantidade" value="<%= (produto != null) ? produto.getQuantidade() : 0
%>"><br>

</div>

<div class="mb-3">

    <label class="form-label" for="preco">Preço:</label>

    <input class="form-control" type="text" name="preco" id="preco" value="<%=
(produto != null) ? produto.getPreco() : 0.0 %>"><br>

</div>

<div>

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

</div>

</form>

</body>

</html>

```

Análise e Conclusão:

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

O padrão Front Controller centraliza todas as requisições para um único controlador, que será responsável por decidir qual ação tomar e qual visão renderizar. No MVC um Servlet age como controlador encaminhando as requisições para diferentes componentes, nesse caso, os JSPs.

B. Quais as diferenças e semelhanças entre Servlets e JSPs?

Ambos podem implementar uma interface web, mas os JSPs usam a linguagem HTML como predominante, enquanto que no Servlet usa o Java.

C. Qual a diferença entre um redirecionamento simples e o uso do método forward, a partir do RequestDispatcher? Para que servem os parâmetro e atributos nos objetos HttpServletRequest?

O redirecionamento simples envia uma nova requisição HTTP ao cliente, alterando a URL. O método forward encaminha internamente as requisições a outro recurso, sem alterar a URL. Nos objetos HttpServletRequest os parâmetros são dados enviados pelo cliente, os atributos são dados armazenados no servidor durante a requisição.

3º Procedimento | Melhorando o Design da Interface

ProdutoLista.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java"
pageEncoding="UTF-8"%>
<%@ page import="java.util.List" %>
<%@ page import="cadastroee.model.Produto" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Lista de Produtos</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEw
IH" crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
</head>
<body class="container">
    <h1>Listagem de Produtos</h1>
    <a href="ServletProdutoFC?acao=formIncluir" class="btn btn-primary m-2">Incluir
Novo Produto</a>
    <br><br>
    <table border="1" class="table table-striped">
        <thead class="table-dark">
```

```

<tr>
    <th>ID</th>
    <th>Nome</th>
    <th>Quantidade</th>
    <th>Preço</th>
    <th>Ações</th>
</tr>
</thead>
<tbody>
    <%
        List<Produto> produtos = (List<Produto>) request.getAttribute("produtos");
        if (produtos != null) {
            if (!produtos.isEmpty()) {
                for (Produto produto : produtos) {
    %>
    <tr>
        <td><%= produto.getIdproduto() %></td>
        <td><%= produto.getNomeproduto() %></td>
        <td><%= produto.getQuantidade() %></td>
        <td><%= produto.getPreco() %></td>
        <td>
            <a
                class="btn
                btn-primary
                btn-sm"
                href="ServletProdutoFC?acao=formAlterar&id=<%=
                produto.getIdproduto()
                %>">Alterar</a>
            <a
                class="btn
                btn-danger
                btn-sm"
                href="ServletProdutoFC?acao=excluir&id=<%=
                produto.getIdproduto()
                %>">Excluir</a>
        </td>
    </tr>
    <%
        }
    } else {
    %>
    <tr>
        <td colspan="5">Nenhum produto encontrado.</td>
    </tr>

```

```

        <%
            }
        } else {
    %>

    <tr>

        <td colspan="5">Nenhum produto encontrado.</td>

    </tr>

    <%
        }
    %>

</tbody>

</table>

</body>

</html>

```

ProdutoDados.jsp

```

<%@      page      contentType="text/html; charset=UTF-8"      language="java"
pageEncoding="UTF-8"%>

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

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Dados do Produto</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet"      integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEw
IH" crossorigin="anonymous">

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>

</head>

<body class="container">

    <h1>Dados do Produto</h1>

    <%

```

```

        Produto produto = (Produto) request.getAttribute("produto");
        String acao = (produto == null) ? "incluir" : "alterar";
    %>
    <form action="ServletProdutoFC" method="post" class="form">
        <input type="hidden" name="acao" value="<%= acao %>">
        <% if (produto != null) { %>
            <input type="hidden" name="id" value="<%= produto.getIdproduto() %>">
        <% } %>
        <div class="mb-3">
            <label class="form-label" for="nome">Nome:</label>
            <input class="form-control" type="text" name="nome" id="nome" value="<%=
(produtor != null) ? produto.getNomeproduto() : "" %>"><br>
        </div>
        <div class="mb-3">
            <label class="form-label" for="quantidade">Quantidade:</label>
            <input class="form-control" type="number" name="quantidade"
id="quantidade" value="<%= (produto != null) ? produto.getQuantidade() : 0
%>"><br>
        </div>
        <div class="mb-3">
            <label class="form-label" for="preco">Preço:</label>
            <input class="form-control" type="text" name="preco" id="preco" value="<%=
(produtor != null) ? produto.getPreco() : 0.0 %>"><br>
        </div>
        <div>
            <button class="btn btn-primary" type="submit"><%= (acao.equals("incluir")) ?
"Incluir Produto" : "Alterar Produto" %></button>
        </div>
    </form>
</body>
</html>

```

Análise e Conclusão:

A. Como o framework Bootstrap é utilizado?

O Bootstrap é usado a partir de um link de conexão no <head> da estrutura HTML, permitindo o acesso a sua biblioteca e suas classes de estilo.

B. Por que o Bootstrap garante a independência estrutural do HTML?

Porque fornece uma coleção de classes CSS e componentes prontos permitindo aplicar estilos e funcionalidade sem alterar a estrutura básica do HTML.

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

O Bootstrap fica responsável pela estilização da página, deixando ela mais atraente e acessível para diferentes plataformas que acessarem.