



Nome do campus: 3366 Polo Centro – Garopaba - SC.

Nome do curso: Desenvolvimento Full Stack.

Nome da disciplina: Vamos integrar sistemas.

Número da turma: 9001.

Nome: Daniel dos Santos Pereira.

Endereço: [DaniielDev/JPAeJEE: Nível 4](#)

Objetivo da Prática: Exibição e entrada de dados na plataforma Java Web.

1 - Como é organizado um projeto corporativo no NetBeans?

É dividido em **módulos**:

- EAR (principal): empacota o sistema.
- EJB (backend): lógica de negócios e persistência.
- WAR (frontend): interface Web (Servlets, JSPs etc.).

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

☐ **JPA**: faz o **mapeamento objeto-relacional** (ORM) entre classes Java e tabelas do banco.

☐ **EJB**: fornece **componentes de negócio reutilizáveis** (Session Beans) para processar regras e lógica.

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

☐ Geração automática de:

- **Entidades JPA a partir do banco.**
- **Session Beans (EJBs) para entidades.**

☐ Facilita criação, edição e persistência com poucos cliques.

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

☐ Servlets são **componentes Java que respondem a requisições HTTP**.

☐ NetBeans oferece:

- **Assistentes visuais** para criação.
- **Geração automática de configuração** no web.xml.

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

☐ É feita por **injeção automática** via anotação @EJB.

☐ O Servlet usa o Session Bean como se fosse um objeto local, sem se preocupar com a criação ou gerenciamento.

Códigos:

```
package cadastroee.servlets;
```

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

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

```
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.io.PrintWriter;
```

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

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

```
public class ServletProduto extends HttpServlet {
```

```
    @EJB
```

```

private ProdutoFacadeLocal facade;

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

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

    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        out.println("<html><head><title>Lista de Produtos</title></head><body>");
        out.println("<h1>Produtos Cadastrados</h1><ul>");
        for (Produto p : produtos) {
            out.println("<li>" + p.getId() + " - " + p.getDescricao() + " - R$ " + p.getPrecoVenda() +
"</li>");
        }
        out.println("</ul></body></html>");
    }
}
}

```

```

CREATE DATABASE loja;

```

```

GO

```

```

USE loja;

```

```

GO

```

```

CREATE TABLE Produto (
    id INT PRIMARY KEY IDENTITY(1,1),
    descricao VARCHAR(100) NOT NULL,
    precoVenda FLOAT
);

```

GO

```
INSERT INTO Produto (descricao, precoVenda) VALUES
```

```
('Teclado Gamer', 199.99),
```

```
('Mouse Óptico', 89.90),
```

```
('Monitor LED 24"', 799.00),
```

```
('Notebook i5 10ª Geração', 3599.99);
```

GO

## 2 – Procedimento:

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

☐ **Função:** Centraliza o controle das requisições em um único ponto (geralmente um Servlet).

☐ **Responsabilidade:** Interpretar a ação solicitada, encaminhar para o modelo e escolher a view correta.

☐ **Implementação:** Um único Servlet (ex: ServletProdutoFC) recebe todas as requisições e decide o fluxo com base no parâmetro acao.

2 - Quais as diferenças e semelhanças entre Servlets e JSPs?

Aspecto	Servlet	JSP
Código Java	Escrito diretamente em Java	Escrito em HTML com inserções Java
Foco principal	Lógica de controle	Apresentação (interface)
Execução	Compilado como classe Java	Convertido para Servlet internamente
Uso ideal	Controle, acesso a dados, lógica	Exibir resultados ao usuário

3 - 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?

### ❓ **Parâmetros (request.getParameter)**

- Vêm da URL ou do formulário (GET ou POST).
- Sempre são String.
- Ex: ?id=5 → request.getParameter("id").

### ❓ **Atributos (request.setAttribute)**

- Usados para enviar objetos (como entidades) entre Servlets e JSPs.
- Existentes apenas durante a requisição atual.
- Ex: request.setAttribute("produto", produto);

Códigos:

```
package cadastroee.servlets;
```

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

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

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

```
import jakarta.servlet.RequestDispatcher;
```

```
import jakarta.servlet.ServletException;
```

```
import jakarta.servlet.annotation.WebServlet;
```

```
import jakarta.servlet.http.*;
```

```
import java.io.IOException;
```

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

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

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

```
    @EJB
```

```
    private ProdutoFacadeLocal facade;
```

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException {
```

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

String destino = "ProdutoLista.jsp";

if ("formIncluir".equals(acao) || "formAlterar".equals(acao)) {
    destino = "ProdutoDados.jsp";
    if ("formAlterar".equals(acao)) {
        int id = Integer.parseInt(request.getParameter("id"));
        Produto p = facade.find(id);
        request.setAttribute("produto", p);
    }
} else if ("listar".equals(acao)) {
    List<Produto> produtos = facade.findAll();
    request.setAttribute("lista", produtos);
} else if ("excluir".equals(acao)) {
    int id = Integer.parseInt(request.getParameter("id"));
    Produto p = facade.find(id);
    facade.remove(p);
    request.setAttribute("lista", facade.findAll());
} else if ("alterar".equals(acao)) {
    int id = Integer.parseInt(request.getParameter("id"));
    Produto p = facade.find(id);
    p.setDescricao(request.getParameter("descricao"));
    p.setPrecoVenda(Float.parseFloat(request.getParameter("precoVenda")));
    facade.edit(p);
    request.setAttribute("lista", facade.findAll());
} else if ("incluir".equals(acao)) {
    Produto p = new Produto();
    p.setDescricao(request.getParameter("descricao"));
    p.setPrecoVenda(Float.parseFloat(request.getParameter("precoVenda")));
    facade.create(p);
    request.setAttribute("lista", facade.findAll());
}
```

```

    }

    RequestDispatcher dispatcher = request.getRequestDispatcher(destino);
    dispatcher.forward(request, response);
}

@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);
}
}

```

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.util.List"%>
<%@page import="cadastroee.model.Produto"%>
<html>
<head><title>Lista de Produtos</title></head>
<body>
    <h1>Produtos Cadastrados</h1>
    <a href="ServletProdutoFC?acao=formIncluir">Incluir Novo Produto</a>
    <table border="1">
        <tr><th>ID</th><th>Descrição</th><th>Preço</th><th>Ações</th></tr>
        <%
            List<Produto> lista = (List<Produto>) request.getAttribute("lista");

```

```

        if (lista != null) {
            for (Produto p : lista) {
                %>
                <tr>
                    <td><%= p.getId() %></td>
                    <td><%= p.getDescricao() %></td>
                    <td>R$ <%= p.getPrecoVenda() %></td>
                    <td>
                        <a href="ServletProdutoFC?acao=formAlterar&id=<%= p.getId() %>">Alterar</a> |
                        <a href="ServletProdutoFC?acao=excluir&id=<%= p.getId() %>">Excluir</a>
                    </td>
                </tr>
            }
        }
    %>
</table>
</body>
</html>

```

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="cadastroee.model.Produto"%>
<%
    Produto produto = (Produto) request.getAttribute("produto");
    boolean alterando = (produto != null);
    String acao = alterando ? "alterar" : "incluir";
%>
<html>
<head><title><%= alterando ? "Alterar" : "Incluir" %> Produto</title></head>
<body>
    <h1><%= alterando ? "Alterar" : "Incluir" %> Produto</h1>

```



```

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

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

    <% if (alterando) { %>

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

    <% } %>

    Descrição: <input type="text" name="descricao" value="<%= alterando ?
produto.getDescricao() : "" %>"><br>

    Preço Venda: <input type="text" name="precoVenda" value="<%= alterando ?
produto.getPrecoVenda() : "" %>"><br>

    <input type="submit" value="<%= alterando ? "Salvar Alterações" : "Cadastrar Produto"
%>">

</form>

</body>

</html>

```

### 3 – Procedimento:

#### 1. Como o framework Bootstrap é utilizado?

É incluído no HTML via **CDN** ou arquivos locais (CSS + JS).

Usa **classes CSS prontas** para estilizar botões, tabelas, formulários, etc.

Permite montar **layouts modernos e responsivos** com menos código.

#### 2 -Por que o Bootstrap garante a independência estrutural do HTML?

Separa **estrutura (HTML)** da **estilização (CSS)**.

O HTML continua funcional mesmo sem os estilos Bootstrap.

Facilita a troca de aparência sem alterar a estrutura da página.

#### 3 - Qual a relação entre o Bootstrap e a responsividade da página?

- Bootstrap usa o conceito de **grid flexível (sistema de colunas)**.

- Adapta o conteúdo automaticamente a **tamanhos de tela diferentes** (celular, tablet, desktop).
- Inclui **classes utilitárias** para esconder, alinhar ou redimensionar elementos conforme o dispositivo.

Códigos:

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Cadastro de Produtos</title>
```

```
  <!-- Bootstrap CSS -->
```

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

```
  <!-- Bootstrap JS Bundle (com Popper) -->
```

```
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></scr
ipt>
```

```
</head>
```

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

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

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

```
<%
```

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

```
%>
```

```
<!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">

<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></scr  
ipt>

</head>

<body class="container">

<h1>Lista de Produtos</h1>

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

<table 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>

<%

```

        if (lista != null) {

            for (Produto p : lista) {

                %>

                <tr>

                    <td><%= p.getId() %></td>

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

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

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

                    <td>

                        <a href="ServletProdutoFC?acao=formAlterar&id=<%= p.getId() %>"
class="btn btn-primary btn-sm">Alterar</a>

                        <a href="ServletProdutoFC?acao=excluir&id=<%= p.getId() %>"
class="btn btn-danger btn-sm">Excluir</a>

                    </td>

                </tr>

                <%

                    }

                }

            %>

        </tbody>

    </table>

</body>

</html>

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

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

<%

```

```

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

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

    %>

<!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">

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></scr
ipt>

</head>

<body class="container">

    <h1><%= (acao.equals("incluir") ? "Incluir" : "Alterar") %> Produto</h1>

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

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

        <%

            if (acao.equals("alterar")) {

        %>

        <input type="hidden" name="id" value="<%= prod.getId() %>" />

        <%

            }

        %>

```

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

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

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

</div>


<div class="mb-3">

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

    <input type="number" name="quantidade" class="form-control" value="<%=
(prod != null) ? prod.getQuantidade() : "" %>"/>

</div>


<div class="mb-3">

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

    <input type="text" name="precoVenda" class="form-control" value="<%=
(prod != null) ? prod.getPrecoVenda() : "" %>"/>

</div>


<button type="submit" class="btn btn-primary">

    <%= (acao.equals("incluir") ? "Incluir" : "Alterar") %>

</button>

</form>

</body>

</html>
```