



# Estácio

## Missão Prática Nível 4 – Mundo 3

Fernanda Canto P. da Costa - 202208379788

Campus de Ipanema

Vamos Integrar Sistemas – 22.3 – 3º semestre

Github: <https://github.com/nandacpc/Missao-Nv4-M3-1parte>  
<https://github.com/nandacpc/Missao-Nv4-M3-2parte>

### Objetivo da Prática

- Implementar persistência com base em JPA.
- Implementar regras de negócio na plataforma JEE, através de EJBs.
- Implementar sistema cadastral Web com base em Servlets e JSPs.
- Utilizar a biblioteca Bootstrap para melhoria do design.

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

Arquivo ServletProduto.java:

```
package cadastroee.servlets;

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

public class ServletProduto extends HttpServlet {

    @EJB
    ProdutoFacadeLocal facade;
```

```

        protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
            throws ServletException, IOException {
            response.setContentType("text/html;charset=UTF-8");
            try (PrintWriter out = response.getWriter()) {
                out.println("<!DOCTYPE html>");
                out.println("<html>");
                out.println("<head>");
                out.println("<title>Lista de Produtos</title>");
                out.println("</head>");
                out.println("<body>");
                out.println("<h1>Lista de Produtos</h1>");

                List<Produto> produtos = facade.getProdutos(); // Suponha que
                existe um método getProdutos no EJB

                out.println("<ul>");
                for (Produto produto : produtos) {
                    out.println("<li>" + produto.getNome() + " - " +
                    produto.getPreco() + "</li>");
                }
                out.println("</ul>");

                out.println("</body>");
                out.println("</html>");
            }
        }

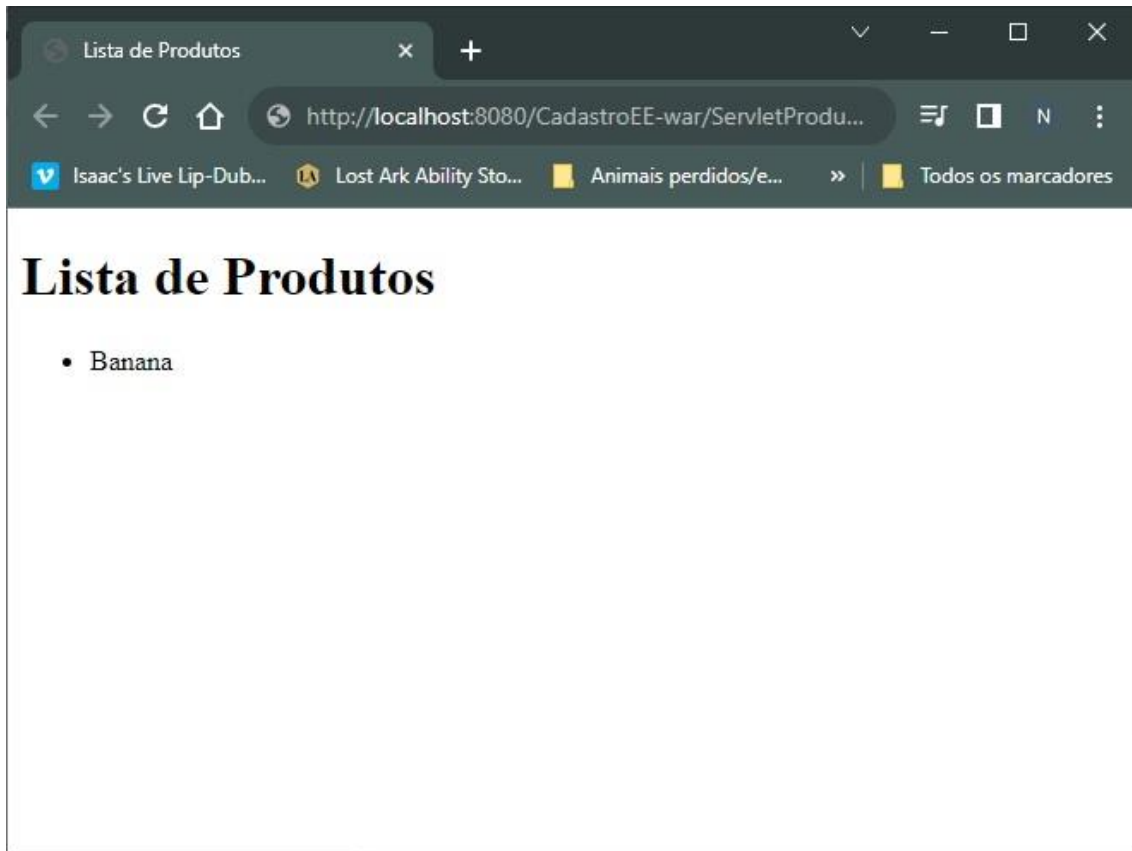
        @Override
        protected void doGet(HttpServletRequest request, HttpServletResponse
        response)
            throws ServletException, IOException {
            processRequest(request, response);
        }

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

        @Override
        public String getServletInfo() {
            return "Servlet de Produtos";
        }
    }

```

## Execução:



## Análise e Conclusão:

1. Como é organizado um projeto corporativo no NetBeans?

R: Um projeto corporativo no NetBeans é organizado em pastas e pacotes que contêm diferentes partes do aplicativo, como a interface de usuário, a lógica de negócios e o acesso a dados. Cada componente é mantido em locais separados para facilitar a organização e a manutenção.

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

R: JPA (Java Persistence API): é usada para mapear objetos Java para o banco de dados, facilitando o armazenamento e recuperação de dados.

EJB (Enterprise JavaBeans): fornece um framework para criar componentes empresariais que oferecem recursos como transações, segurança e serviços de negócios. Eles são essenciais para o desenvolvimento de aplicativos corporativos em Java.

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

R: O NetBeans oferece recursos e assistentes que simplificam a configuração e o uso de JPA e EJB. Ele gera código automaticamente, facilitando a criação de entidades JPA e componentes EJB, o que economiza tempo e reduz a chance de erros.

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

R: Servlets são componentes Java que lidam com solicitações e respostas HTTP. O NetBeans oferece um assistente para a criação de Servlets, permitindo que os desenvolvedores criem facilmente esses componentes. O NetBeans também ajuda a configurar URLs e mapeamentos para os Servlets.

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

R: A comunicação entre Servlets e Session Beans é realizada por meio da injeção de dependências. Os Servlets podem usar anotações para injetar Session Beans e, assim, acessar métodos de negócios. O NetBeans oferece

suporte a essa injeção, simplificando a comunicação entre esses componentes.

## 2º Procedimento | Interface Cadastral com Servlets e JSPs

### Arquivo ServletProdutoFC.java:

```
package cadastroee.servlets;

import cadastroee.controller.ProdutoFacadeLocal;
import cadastroee.model.Produtos;
import java.io.IOException;
import java.util.List;
import javax.ejb.EJB;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@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 = "";

        if (acao != null) {
            switch (acao) {
                case "listar":
                    List<Produtos> produtos = facade.findAll();
                    request.setAttribute("produtos", produtos);
                    destino = "ProdutoLista.jsp";
                    break;
                case "formIncluir":
                case "formAlterar":
```

```

        destino = "ProdutoDados.jsp";
        break;
    case "excluir":
        int idExcluir =
Integer.parseInt(request.getParameter("id"));
        Produtos produtoExcluir = facade.find(idExcluir);
        if (produtoExcluir != null) {
            facade.remove(produtoExcluir);
        }
        List<Produtos> produtosExcluidos = facade.findAll();
        request.setAttribute("produtos", produtosExcluidos);
        destino = "ProdutoLista.jsp";
        break;
    case "alterar":
        int idAlterar =
Integer.parseInt(request.getParameter("id"));
        Produtos produtoAlterar = facade.find(idAlterar);
        if (produtoAlterar != null) {
        }
        List<Produtos> produtosAlterados = facade.findAll();
        request.setAttribute("produtos", produtosAlterados);
        destino = "ProdutoLista.jsp";
        break;
    case "incluir":
        Produtos novoProduto = new Produtos();

        facade.create(novoProduto);
        List<Produtos> produtosIncluidos = facade.findAll();
        request.setAttribute("produtos", produtosIncluidos);
        destino = "ProdutoLista.jsp";
        break;
    default:
        destino = "Erro.jsp";
    }
} else {
    destino = "Erro.jsp";
}

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);
    }

    @Override
    public String getServletInfo() {
        return "Front Controller de Produtos";
    }
}

```

### Arquivo ProdutoLista.jsp:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta 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">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundl
e.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
    <title>Lista de Produtos</title>
</head>
<body class="container">
    <h1>Lista de Produtos</h1>
    <a href="ServletProdutoFC?acao=formIncluir" class="btn btn-primary m-
2">Novo Produto</a>
    <table class="table table-striped" border="1">
        <thead class="table-dark">
            <tr>
                <th>#</th>
                <th>Nome</th>
                <th>Quantidade</th>
                <th>Preço</th>
                <th>Opções</th>

```

```

        </tr>
    </thead>
    <c:forEach var="produto" items="{produtos}">
        <tr>
            <td>${produto.id}</td>
            <td>${produto.nome}</td>
            <td>${produto.quantidade}</td>
            <td>${produto.preco}</td>
            <td>
                <a
href="ServletProdutoFC?acao=formAlterar&id=${produto.id}" class="btn btn-
primary btn-sm">Alterar</a>
                <a
href="ServletProdutoFC?acao=excluir&id=${produto.id}" class="btn btn-
danger btn-sm">Excluir</a>
            </td>
        </tr>
    </c:forEach>
</table>
</body>
</html>

```

### Arquivo ProdutoDados.jsp:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <meta 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">
    <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>
    <title>Produto Dados</title>
</head>
<body class="container">
    <h1>Produto Dados</h1>
    <form action="ServletProdutoFC" method="post" class="form">
        <input type="hidden" name="acao" value="{empty entidade ?
'incluir' : 'alterar'}">

```



```

<c:if test="${entidade != null}">
    <input type="hidden" name="id" value="${entidade.idProduto}">
</c:if>
<div class="mb-3">
    <label for="nome" class="form-label">Nome:</label>
    <input class="form-control" type="text" id="nome" name="nome"
value="${entidade != null ? entidade.nome : ''}" required><br>
</div>
<div class="mb-3">
    <label for="quantidade" class="form-
label">Quantidade:</label>
    <input class="form-control" type="number" id="quantidade"
name="quantidade" value="${entidade != null ? entidade.quantidade : ''}"
required><br>
</div>
<div class="mb-3">
    <label for="precoVenda" class="form-label">Preço de
Venda:</label>
    <input class="form-control" type="number" id="precoVenda"
name="precoVenda" value="${entidade != null ? entidade.precoVenda : ''}"
required><br>
</div>
<input class="btn btn-primary" type="submit" value="${empty
entidade ? 'Incluir Produto' : 'Alterar Produto'}">
</form>
</body>
</html>

```

Execução:

## Lista de Produtos

[Novo Produto](#)

#	Nome	Quantidade	Preço	Opções
1	Abacaxi	70	5.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
2	Banana	200	1.5	<a href="#">Alterar</a> <a href="#">Excluir</a>
4	Laranja	50	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

---

# Produto Dados

Nome:

Quantidade:

Preço de Venda:

## **Análise e Conclusão:**

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

R: O padrão Front Controller é um design comum na arquitetura MVC, onde um servlet central (Front Controller) gerencia todas as solicitações dos clientes, direcionando-as para controladores apropriados. Isso ajuda a organizar um aplicativo web.

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

R: Servlets e JSPs são tecnologias para desenvolvimento web em Java. Servlets são classes Java usadas para lógica de negócios e processamento de solicitações, enquanto JSPs são páginas HTML com código Java embutido, principalmente para apresentação.

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?

R: Um redirecionamento simples envia uma nova solicitação ao navegador do cliente, enquanto o método forward do RequestDispatcher realiza um redirecionamento interno no servidor, mantendo os detalhes da solicitação original. Parâmetros e atributos no HttpServletRequest são usados para compartilhar dados entre componentes do aplicativo. Parâmetros passam informações na URL, e atributos compartilham dados temporários durante o processamento da solicitação.

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

#### Arquivo ProdutoLista.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta 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">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundl
e.min.js" integrity="sha384-
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
    <title>Lista de Produtos</title>
</head>
<body class="container">
    <h1>Lista de Produtos</h1>
    <a href="ServletProdutoFC?acao=formIncluir" class="btn btn-primary m-
2">Novo Produto</a>
    <table class="table table-striped" border="1">
        <thead class="table-dark">
            <tr>
                <th>#</th>
                <th>Nome</th>
                <th>Quantidade</th>
```

```

        <th>Preço</th>
        <th>Opções</th>
    </tr>
</thead>
<c:forEach var="produto" items="${produtos}">
    <tr>
        <td>${produto.id}</td>
        <td>${produto.nome}</td>
        <td>${produto.quantidade}</td>
        <td>${produto.preco}</td>
        <td>
            <a
href="ServletProdutoFC?acao=formAlterar&id=${produto.id}" class="btn btn-
primary btn-sm">Alterar</a>
            <a
href="ServletProdutoFC?acao=excluir&id=${produto.id}" class="btn btn-
danger btn-sm">Excluir</a>
        </td>
    </tr>
</c:forEach>
</table>
</body>
</html>

```

### Arquivo ProdutoDados.jsp:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <meta 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">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundl
e.min.js" integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>
    <title>Produto Dados</title>
</head>
<body class="container">

```

```

<h1>Produto Dados</h1>
<form action="ServletProdutoFC" method="post" class="form">
  <input type="hidden" name="acao" value="{empty entidade ?
'incluir' : 'alterar'}">
  <c:if test="{entidade != null}">
    <input type="hidden" name="id" value="{entidade.idProduto}">
  </c:if>
  <div class="mb-3">
    <label for="nome" class="form-label">Nome:</label>
    <input class="form-control" type="text" id="nome" name="nome"
value="{entidade != null ? entidade.nome : ''}" required><br>
  </div>
  <div class="mb-3">
    <label for="quantidade" class="form-
label">Quantidade:</label>
    <input class="form-control" type="number" id="quantidade"
name="quantidade" value="{entidade != null ? entidade.quantidade : ''}"
required><br>
  </div>
  <div class="mb-3">
    <label for="precoVenda" class="form-label">Preço de
Venda:</label>
    <input class="form-control" type="number" id="precoVenda"
name="precoVenda" value="{entidade != null ? entidade.precoVenda : ''}"
required><br>
  </div>
  <input class="btn btn-primary" type="submit" value="{empty
entidade ? 'Incluir Produto' : 'Alterar Produto'}">
</form>
</body>
</html>

```

## Execução:

# Produto Dados

Nome:

Quantidade:

Preço de Venda:

Alterar Produto

## Lista de Produtos

Novo Produto

#	Nome	Quantidade	Preço	Opções	
1	Abacaxi	70	5.0	<button>Alterar</button>	<button>Excluir</button>
2	Banana	200	1.5	<button>Alterar</button>	<button>Excluir</button>
4	Laranja	50	2.0	<button>Alterar</button>	<button>Excluir</button>

## Análise e Conclusão:

1.Como o framework Bootstrap é utilizado?

R: O Bootstrap é um conjunto de ferramentas que facilita o design de páginas web. Você o incorpora ao seu site e usa as classes CSS e os componentes fornecidos para estilizar e estruturar seu conteúdo. No seu código, isso é feito com os links para os arquivos CSS e JavaScript do Bootstrap.

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

R: Ele permite adicionar estilo e funcionalidade sem mexer muito no HTML. Usando as classes do Bootstrap, você mantém uma separação legal entre a estrutura (HTML) e o visual (CSS), facilitando a manutenção.

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

R: O Bootstrap é ótimo para fazer sites que funcionam bem em diferentes tamanhos de tela. Ele tem uma grade flexível que se ajusta automaticamente, tornando seu site bonito tanto no celular quanto no computador.