Missão Prática | Nível 4 | Mundo 3

Eduardo Eugênio de Araujo e Silva Domingues 202302579043

1172 POLO CASA CAIADA – OLINDA – PE
RPG0017 - Vamos integrar sistemas – 2024.1

https://github.com/eduardoduud/Missao-Pratica-Nivel-4-Mundo-3

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

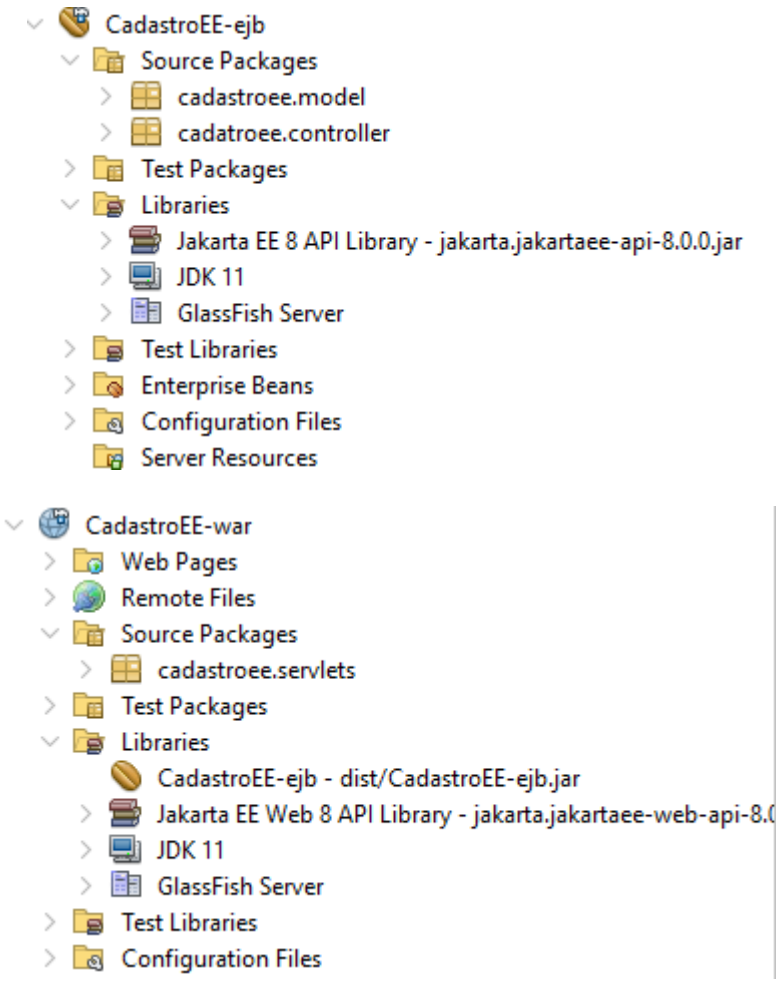
ServletProduto.java

```
package cadastroee.servlets;


import cadatroee.controller.ProdutoFacadeLocal;

import cadastroee.model.Produto;

import jakarta.ejb.EJB;

import java.io.IOException;

import java.util.List;

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

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

```
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name = "ServletProduto", urlPatterns = {"/ServletProduto"})
public class ServletProduto extends HttpServlet {

    @EJB
    private ProdutoFacadeLocal produtoFacade;

    @Override
    protected void doGet(HttpServletRequest requisicao, HttpServletResponse resposta)
throws ServletException, IOException {
        String acaoRequisicao = requisicao.getParameter("acao");
        if (acaoRequisicao == null || acaoRequisicao.isEmpty()) {
            acaoRequisicao = "listar";
        }
        switch (acaoRequisicao) {
            case "excluir":
                excluirProduto(requisicao, resposta);
                break;
            case "formAlterar":
                exibirFormularioAlteracao(requisicao, resposta);
                break;
            case "formIncluir":
```

```
            exibirFormularioInclusao(requisicao, resposta);
            break;
        default:
            listarProdutos(requisicao, resposta);
    }
}

@Override
protected void doPost(HttpServletRequest requisicao, HttpServletResponse resposta)
throws ServletException, IOException {
    String acaoRequisicao = requisicao.getParameter("acao");
    if (acaoRequisicao != null && !acaoRequisicao.isEmpty()) {
        switch (acaoRequisicao) {
            case "incluir":
                incluirProduto(requisicao, resposta);
                break;
            case "alterar":
                alterarProduto(requisicao, resposta);
                break;
        }
    } else {
        resposta.sendRedirect("ServletProduto");
    }
}
```

```
    private void listarProdutos(HttpServletRequest requisicao, HttpServletResponse
resposta) throws ServletException, IOException {
        List<Produto> listaProdutos = produtoFacade.findAll();
        requisicao.setAttribute("produtos", listaProdutos);
        RequestDispatcher despachante =
requisicao.getRequestDispatcher("ProdutoLista.jsp");
        despachante.forward(requisicao, resposta);
    }
}
```



Conclusão:

a) É organizado em múltiplos módulos ou subprojetos que separam as diferentes camadas e responsabilidades do aplicativo. Essa organização modular facilita o desenvolvimento, manutenção e escalabilidade do sistema.

b) JPA facilita a persistência de dados em bancos relacionais via ORM. EJB gerencia lógica de negócios, transações e segurança de forma distribuída.

c) Melhora a produtividade com assistentes de criação de entidades e beans, mapeamento visual de JPA, integração com servidores e ferramenta de debugging.

d) São componentes Java para responder a solicitações HTTP. O NetBeans suporta criação, mapeamento automático no web.xml e ferramentas de edição e debug integradas.

e) Servlets comunicam-se com a Session Beans via injeção de dependência usando EJB, permitindo que os servlets usem métodos do beans para lógica de negócio.

Observe que os tópicos acima seguem exatamente o que está na Atividade Prática exigida.

2º Procedimento | Interface Cadastral com Servlet e JSPs

ServletProduto.java

```
package cadastroee.servlets;


import cadatroee.controller.ProdutoFacadeLocal;

import cadastroee.model.Produto;

import jakarta.ejb.EJB;

import java.io.IOException;

import java.util.List;

import jakarta.servlet.RequestDispatcher;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;


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

public class ServletProduto extends HttpServlet {


    @EJB

    private ProdutoFacadeLocal produtoFacade;


    @Override
```

```
    protected void doGet(HttpServletRequest requisicao, HttpServletResponse resposta)
throws ServletException, IOException {
        String acaoRequisicao = requisicao.getParameter("acao");
        if (acaoRequisicao == null || acaoRequisicao.isEmpty()) {
            acaoRequisicao = "listar";
        }
        switch (acaoRequisicao) {
            case "excluir":
                excluirProduto(requisicao, resposta);
                break;
            case "formAlterar":
                exibirFormularioAlteracao(requisicao, resposta);
                break;
            case "formIncluir":
                exibirFormularioInclusao(requisicao, resposta);
                break;
            default:
                listarProdutos(requisicao, resposta);
        }
    }

    @Override
    protected void doPost(HttpServletRequest requisicao, HttpServletResponse resposta)
throws ServletException, IOException {
        String acaoRequisicao = requisicao.getParameter("acao");
```

```
        if (acaoRequisicao != null && !acaoRequisicao.isEmpty()) {
            switch (acaoRequisicao) {
                case "incluir":
                    incluirProduto(requisicao, resposta);
                    break;
                case "alterar":
                    alterarProduto(requisicao, resposta);
                    break;
            }
        } else {
            resposta.sendRedirect("ServletProduto");
        }
    }

    private void listarProdutos(HttpServletRequest requisicao, HttpServletResponse resposta) throws ServletException, IOException {
        List<Produto> listaProdutos = produtoFacade.findAll();
        requisicao.setAttribute("produtos", listaProdutos);
        RequestDispatcher despachante = requisicao.getRequestDispatcher("ProdutoLista.jsp");
        despachante.forward(requisicao, resposta);
    }

    private void excluirProduto(HttpServletRequest requisicao, HttpServletResponse resposta) throws ServletException, IOException {
        try {
```

```
            int idProdutoExcluir = Integer.parseInt(requisicao.getParameter("id"));
            produtoFacade.remove(produtoFacade.find(idProdutoExcluir));
            resposta.sendRedirect("ServletProduto");
        } catch (NumberFormatException | NullPointerException e) {
            resposta.sendError(HttpServletResponse.SC_BAD_REQUEST);
        }
    }

    private void exibirFormularioAlteracao(HttpServletRequest requisicao,
HttpServletResponse resposta) throws ServletException, IOException {
        try {
            int idProdutoAlterar = Integer.parseInt(requisicao.getParameter("id"));
            Produto produto = produtoFacade.find(idProdutoAlterar);
            requisicao.setAttribute("produto", produto);
            RequestDispatcher despachante =
requisicao.getRequestDispatcher("ProdutoDados.jsp");
            despachante.forward(requisicao, resposta);
        } catch (NumberFormatException | NullPointerException e) {
            resposta.sendError(HttpServletResponse.SC_BAD_REQUEST);
        }
    }

    private void incluirProduto(HttpServletRequest requisicao, HttpServletResponse
resposta) throws ServletException, IOException {
        try {
            String nomeProduto = requisicao.getParameter("nome");
```

```
            int quantidadeProduto =
Integer.parseInt(requisicao.getParameter("quantidade"));
            Float precoProduto = Float.valueOf(requisicao.getParameter("precoVenda"));
            Produto novoProduto = new Produto();
            novoProduto.setNome(nomeProduto);
            novoProduto.setQuantidade(quantidadeProduto);
            novoProduto.setPrecoVenda(precoProduto);
            produtoFacade.create(novoProduto);
            resposta.sendRedirect("ServletProduto");
        } catch (NumberFormatException | NullPointerException e) {
            resposta.sendError(HttpServletResponse.SC_BAD_REQUEST);
        }
    }

    private void alterarProduto(HttpServletRequest requisicao, HttpServletResponse
resposta) throws ServletException, IOException {
        try {
            int idProduto = Integer.parseInt(requisicao.getParameter("id"));
            String nomeProduto = requisicao.getParameter("nome");
            int quantidadeProduto =
Integer.parseInt(requisicao.getParameter("quantidade"));
            Float precoProduto = Float.valueOf(requisicao.getParameter("precoVenda"));
            Produto produtoAlterado = produtoFacade.find(idProduto);
            produtoAlterado.setNome(nomeProduto);
            produtoAlterado.setQuantidade(quantidadeProduto);
            produtoAlterado.setPrecoVenda(precoProduto);
```

```
            produtoFacade.edit(produtoAlterado);
            resposta.sendRedirect("ServletProduto");
        } catch (NumberFormatException | NullPointerException e) {
            resposta.sendError(HttpServletResponse.SC_BAD_REQUEST);
        }
    }

    private void exibirFormularioInclusao(HttpServletRequest requisicao,
HttpServletResponse resposta) throws ServletException, IOException {
        RequestDispatcher despachante =
requisicao.getRequestDispatcher("ProdutoDados.jsp");
        despachante.forward(requisicao, resposta);
    }
}
```

ProdutoLista.jsp

```
<%@page import="java.util.List"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page import="cadastroee.model.Produto"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Lista de Produtos</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
rel="stylesheet">
```

```
</head>
<body>
    <div class="container">
        <h1>Lista de Produtos</h1>
        <a href="ServletProduto?acao=formIncluir" class="btn btn-primary mb-3">Incluir Produto</a>
        <c:if test="${not empty produtos}">
            <table class="table table-striped">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Nome</th>
                        <th>Quantidade</th>
                        <th>Preço</th>
                        <th>Ações</th>
                    </tr>
                </thead>
                <tbody>
                    <c:forEach var="produto" items="${produtos}">
                        <tr>
                            <td>${produto.idProduto}</td>
                            <td>${produto.nome}</td>
                            <td>${produto.quantidade}</td>
                            <td>${produto.precoVenda}</td>
                            <td>
```


```
                        <a
href="ServletProduto?acao=formAlterar&id=${produto.idProduto}"  class="btn  btn-
warning">Alterar</a>
                        <a  href="ServletProduto?acao=excluir&id=${produto.idProduto}"
class="btn btn-danger">Excluir</a>
                    </td>
                </tr>
            </c:forEach>
        </tbody>
    </table>
</c:if>
<c:if test="${empty produtos}">
    <p>Nenhum produto encontrado.</p>
</c:if>
</div>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></scrip
t>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
</body>
</html>
```

ProdutoDados.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>Cadastro de Produto</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        <c:if test="${produto != null}">
            <h1>Alteração de Produto</h1>
        </c:if>
        <c:if test="${produto == null}">
            <h1>Cadastro de Produto</h1>
        </c:if>
        <form action="ServletProduto" method="post">
```

```
        <input type="hidden" name="acao" value="${produto != null ? 'alterar' :
'incluir'}">
        <c:if test="${produto != null}">
            <input type="hidden" name="id" value="${produto.idProduto}">
        </c:if>
        <div class="form-group">
            <label for="nome">Nome</label>
            <input type="text" class="form-control" id="nome" name="nome"
value="${produto != null ? produto.nome : ''}" required>
        </div>
        <div class="form-group">
            <label for="quantidade">Quantidade</label>
            <input type="number" class="form-control" id="quantidade"
name="quantidade" value="${produto != null ? produto.quantidade : ''}" required>
        </div>
        <div class="form-group">
            <label for="precoVenda">Preço de Venda</label>
            <input type="number" step="0.01" class="form-control" id="precoVenda"
name="precoVenda" value="${produto != null ? produto.precoVenda : ''}" required>
        </div>
        <button type="submit" class="btn btn-primary">${produto != null ? 'Alterar' :
'Incluir'}</button>
    </form>
</div>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
```

```
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

</body>

</html>
```

Estácio

Alteração de Produto

Nome

Samsung

Quantidade

55

Preço de Venda

22,0

Alterar

Conclusão:

a) O padrão Front Controller centraliza o controle de todas as solicitações em um único servlet, que encaminha as solicitações para diferentes manipuladores. Em MVC, o servlet processa as ações e decide qual view mostrar.

b) São classes Java que processam solicitações HTTP, JSPs são páginas que combinam HTML com Java para gerar conteúdo dinâmico. Ambos podem gerar HTML, mas servlets são mais orientados a lógica e JSPs a apresentação.

c) O redirecionamento simples ‘response.sendRedirect’ faz o navegador fazer uma nova solicitação, e o ‘RequestDispatcher.forward’ encaminha a solicitação internamente sem mudar a URL. Parâmetros transmitem dados na URL ou corpo da solicitação, atributos armazenam dados no request ou session.

Observe que os tópicos acima seguem exatamente o que está na Atividade Prática exigida.

3º Procedimento | Melhorando o Design da Interface

Eu fui fazendo já com o bootstrap em mente desde o 1º procedimento, o Código já está nos procedimentos anteriores e as prints também.

Conclusão:

a) Ao incluir seus arquivos CSS e JavaScript em uma página HTML, fornecendo estilos e componentes pré-desenvolvidos
b) Separando o CSS da estrutura, permitindo que os desenvolvedores apliquem estilos sem alterar o HTML.
c) Facilita a responsividade da página através de um grid system flexível e classes utilitárias que adaptam o layout a diferentes tamanhos de tela, garantindo uma experiência consistente em dispositivos variados