

Implementação de Sistema Cadastral Web com Arquitetura Multicamadas (JPA, EJB, Servlet e JSP).

1. Objetivo da Prática

- Implementar a camada de persistência utilizando **JPA**.
- Desenvolver a lógica de negócio através de **EJBs** (Session Beans).
- Criar uma interface Web dinâmica com **Servlets** e **JSPs**.
- Aplicar o padrão **Front Controller** para gerenciamento de requisições.
- Utilizar o framework **Bootstrap** para garantir design responsivo e moderno.

2. Códigos Solicitados

persistence.xml

XML

```
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" ...>
  <persistence-unit name="CadastroEE-ejbPU" transaction-type="JTA">
    <jta-data-source>jdbc/loja</jta-data-source>
    <exclude-unlisted-classes>>false</exclude-unlisted-classes>
  </persistence-unit>
</persistence>
```

ServletProdutoFC (Trecho Principal)

Java

```
String acao = request.getParameter("acao");
String destino = "ProdutoLista.jsp";

if (acao.equals("listar")) {
    request.setAttribute("listaProdutos", facade.findAll());
} else if (acao.equals("formAlterar")) {
    int id = Integer.parseInt(request.getParameter("id"));
```

```
request.setAttribute("produto", facade.find(id));  
  
destino = "ProdutoDados.jsp";  
  
}  
  
// ... outros casos (incluir, excluir, alterar)  
  
request.getRequestDispatcher(destino).forward(request, response);
```

3. Resultados da Execução

O sistema permite a navegação completa entre a listagem de produtos e o formulário de cadastro. Ao acessar `ServletProdutoFC?acao=listar`, os dados persistidos no SQL Server são exibidos em uma tabela Bootstrap, com links funcionais para edição e exclusão que refletem as alterações em tempo real no banco de dados.

4. Análise e Conclusão

Sobre o Frontend (Bootstrap)

- **Utilização:** O Bootstrap é integrado ao projeto via CDN ou arquivos locais, utilizando classes pré-definidas (como table, btn, container) diretamente nas tags HTML.
- **Independência Estrutural:** Ele permite que o HTML foque na estrutura do conteúdo enquanto as classes CSS gerenciam a estética, evitando que o desenvolvedor precise escrever CSS customizado complexo para layouts padrão.
- **Responsividade:** Através de um sistema de **Grid (Grid System)** baseado em colunas e *breakpoints*, o Bootstrap ajusta automaticamente o tamanho e a disposição dos elementos conforme a tela do usuário (mobile, tablet ou desktop).

Sobre o Padrão Front Controller e MVC

- **Funcionamento:** O **Front Controller** concentra todas as requisições em um único ponto (o Servlet ServletProdutoFC). Ele interpreta o parâmetro *acao*, executa a lógica necessária via EJB e decide qual JSP deve exibir o resultado.
- **Arquitetura MVC:** * **M (Model):** Entidades JPA e Banco de Dados.
 - **V (View):** Páginas JSP.
 - **C (Controller):** O Servlet (Front Controller).

Servlets vs JSPs

Característica	Servlet	JSP
Natureza	Classe Java (.java)	Documento de texto/HTML (.jsp)
Foco	Lógica de controle e processamento	Apresentação e interface visual
Vantagem	Melhor para manipular dados e APIs	Mais fácil para desenhar o HTML

Manipulação de Requisições

- **Forward vs Redirect:** O forward (RequestDispatcher) ocorre internamente no servidor; a URL no navegador não muda e os dados do request são preservados. O redirect instrui o navegador a fazer uma nova requisição, limpando os dados anteriores e alterando a URL.
- **Parâmetros vs Atributos:** **Parâmetros** são dados enviados pelo usuário (via URL ou formulário, sempre Strings). **Atributos** são objetos Java adicionados pelo servidor ao request para passar dados entre o Servlet e o JSP.

Integração no NetBeans (Enterprise)

- **Organização:** O NetBeans organiza o projeto em um arquivo **EAR**, separando o **EJB** (lógica/persistência) do **WAR** (interface web).
- **Produtividade:** A IDE automatiza a geração de entidades JPA a partir do banco e cria os *Session Beans* (Facades) com todos os métodos de CRUD prontos, economizando horas de codificação manual.
- **Comunicação EJB/Servlet:** É feita via **Injeção de Dependência** (@EJB). O Servlet não precisa saber como o objeto foi criado; o servidor GlassFish gerencia o ciclo de vida e fornece a referência pronta para uso.