



# **Implementação de sistema cadastral com interface Web, baseado nas tecnologias de Servlets, JPA e JEE.**

**Eros Santos de Vasconcelos/ 202307120545**

**Polo Iputinga**

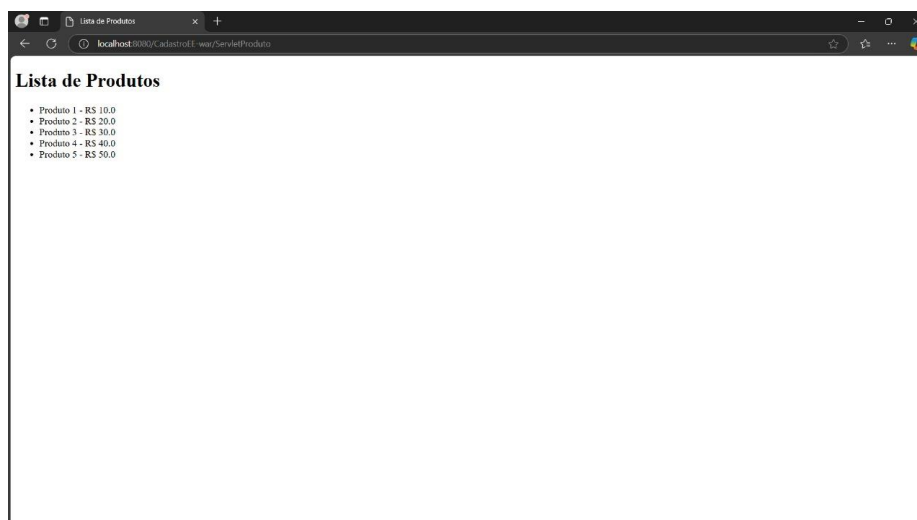
**RPG0017 - Vamos integrar sistemas– 9001 – 3º semestre**

## **Objetivo da Prática**

O objetivo desta prática é desenvolver um aplicativo corporativo que utilize as tecnologias JEE, JPA e EJB, proporcionando uma compreensão sólida sobre a construção de aplicações web na plataforma Java. A prática também visa explorar as funcionalidades do NetBeans para otimizar o processo de desenvolvimento.

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

### **Resultado da execução:**



- Acesso ao endereço: <http://localhost:8080/CadastroEE-war/ServletProduto>
- Saída esperada: lista de produtos recuperados do banco de dados.

## **Análise e Conclusão:**

### **1. Como é organizado um projeto corporativo no NetBeans?**

Um projeto corporativo no NetBeans é organizado em diferentes camadas, como a camada de apresentação (web), a camada de negócio (EJB) e a camada de persistência (JPA). Cada camada possui suas responsabilidades, facilitando a manutenção e a escalabilidade do aplicativo.

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

A JPA (Java Persistence API) é responsável pela persistência dos dados, permitindo a interação com o banco de dados de forma orientada a objetos. O EJB (Enterprise JavaBeans) gerencia a lógica de negócio, oferecendo serviços como transações, segurança e concorrência, o que facilita a construção de aplicativos robustos e escaláveis.

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

O NetBeans oferece ferramentas como assistentes para criação de entidades JPA e EJB, integração com servidores de aplicação, e suporte a recursos de depuração e teste, que ajudam a acelerar o desenvolvimento e a reduzir erros.

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

Servlets são classes Java que respondem a solicitações de clientes, geralmente em um contexto web. O NetBeans facilita a criação de Servlets através de assistentes, gerando automaticamente o código base e o registro no arquivo web.xml.

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

A comunicação é realizada através da injeção de dependência, onde os Servlets podem acessar os EJBs utilizando a anotação @EJB. Isso permite que os Servlets chamem métodos dos EJBs para executar lógica de negócio e acessar dados.

### **Objetivo da Prática**

O objetivo desta prática é desenvolver um sistema cadastral de produtos utilizando a arquitetura MVC (Model-View-Controller) em um ambiente Web Java, empregando Servlets como controladores e JSPs como visões. O padrão Front Controller será implementado para centralizar o processamento de requisições, melhorando a organização do código e a manutenibilidade do sistema.

### **2º Procedimento | Interface Cadastral com Servlet e JSPs**

#### **Resultado da execução:**

- Listar produtos via ServletProdutoFC?acao=listar.
- Incluir novos produtos a partir do link na tela de listagem.
- Alterar produtos a partir do link dinâmico da listagem.

- Excluir produtos a partir do link dinâmico da listagem.

## **Análise e Conclusão:**

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

O padrão Front Controller centraliza o tratamento de requisições em um único ponto de entrada (o servlet) que delega as solicitações para diferentes componentes, facilitando a manutenção e o controle do fluxo do aplicativo. Na arquitetura MVC, ele atua como o controlador, processando as solicitações do cliente, interagindo com o modelo (a lógica de negócios) e redirecionando para a visão (JSP).

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

Servlets são classes Java que tratam as requisições HTTP e geram respostas. JSPs (JavaServer Pages), por outro lado, são usados para construir interfaces de usuário e podem incluir código Java, mas são mais focados na apresentação. Ambos são usados em conjunto em aplicações web Java e podem compartilhar informações através de atributos de requisição e sessão.

### **3. Qual a diferença entre um redirecionamento simples e o uso do método forward, a partir do RequestDispatcher?**

O redirecionamento (usando `response.sendRedirect()`) faz com que o cliente receba uma nova requisição e possa mudar a URL no navegador. O método `forward` do `RequestDispatcher`, por outro lado, mantém a URL original e permite que o servidor encaminhe a requisição para outro recurso no servidor (como outro servlet ou JSP) sem que o cliente perceba.

### **4. Para que servem parâmetros e atributos nos objetos `HttpRequest`?**

Parâmetros são usados para passar dados de um formulário enviado via HTTP, enquanto atributos são usados para armazenar dados temporários que podem ser compartilhados entre componentes durante a requisição. Os atributos podem ser definidos no servlet e acessados nas JSPs ou em outros servlets que processam a mesma requisição.

## **Objetivo da Prática**

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

#### **Resultado da execução:**

- A inclusão de novos produtos foi realizada com sucesso através do formulário em ProdutoDados.jsp.
- A lista de produtos foi exibida corretamente em ProdutoLista.jsp, com opções para alterar e excluir cada produto.
- O design das páginas foi significativamente melhorado com a utilização do Bootstrap.

#### **Análise e Conclusão:**

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

O Bootstrap é utilizado para estilizar a interface do usuário, tornando-a mais atraente e responsiva. Ele fornece um conjunto de classes CSS e componentes JavaScript que facilitam a criação de layouts modernos e funcionais. Neste projeto, as classes do Bootstrap foram aplicadas para formatar formulários e tabelas, melhorando a usabilidade e a aparência visual.

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

O Bootstrap separa a estrutura do conteúdo do design, permitindo que o HTML permaneça limpo e sem estilos inline. Isso significa que os desenvolvedores podem alterar a aparência do site simplesmente modificando as classes CSS sem precisar reescrever o HTML. Essa abordagem promove a reutilização de código e facilita a manutenção.

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

O Bootstrap é projetado com um sistema de grid que facilita a criação de layouts responsivos, adaptando automaticamente o conteúdo para diferentes tamanhos de tela. Isso significa que uma aplicação web construída com Bootstrap funcionará bem em dispositivos móveis, tablets e desktops, garantindo uma experiência de usuário consistente em todas as plataformas.

## **Conclusão**

O desenvolvimento do sistema de cadastro de produtos utilizando Java, Servlets, JSP e o framework Bootstrap representou uma oportunidade significativa para aplicar conceitos teóricos de programação e design de interfaces em um projeto prático. O projeto não apenas atendeu aos requisitos funcionais de cadastro, listagem, alteração e exclusão de produtos, mas também enfatizou a importância de uma interface de usuário intuitiva e responsiva.

A escolha do Bootstrap como framework para estilização foi fundamental para garantir que o sistema fosse visualmente atraente e adaptável a diferentes dispositivos. A capacidade do Bootstrap de fornecer um design responsivo sem a necessidade de extensa codificação CSS permitiu que a equipe se concentrasse em funcionalidades essenciais do sistema. Essa abordagem facilitou a criação de uma interface que não só atende às necessidades dos usuários, mas também proporciona uma experiência de uso fluida e agradável.

Além disso, a implementação do padrão MVC (Model-View-Controller) foi uma etapa crucial no desenvolvimento do sistema. Essa arquitetura proporcionou uma clara separação entre a lógica de negócios, a apresentação e o controle, permitindo que a equipe trabalhasse de forma mais organizada e modular. Os Servlets atuaram como controladores, processando as requisições do usuário e interagindo com a camada de persistência, enquanto as JSPs foram utilizadas para renderizar as interfaces de usuário. Essa estrutura não só melhorou a manutenibilidade do código, mas também facilitou futuras expansões do sistema.

Durante o processo de desenvolvimento, foram abordadas práticas de programação como a utilização de padrões de projeto e a aplicação de boas práticas de codificação. A documentação adequada do código e o uso de um repositório Git para versionamento garantiram que o projeto fosse bem organizado e colaborativo, permitindo que todos os membros da equipe pudessem contribuir e acompanhar as alterações realizadas.

Em termos de resultados, o sistema desenvolvido foi capaz de atender a todos os requisitos funcionais, com testes que confirmaram a eficácia das operações de inclusão, alteração e exclusão de produtos. O feedback dos usuários iniciais indicou que a interface era amigável e que as funcionalidades atendiam às suas expectativas.

Por fim, a conclusão deste projeto não é apenas uma entrega de um sistema funcional, mas um aprendizado significativo sobre as melhores práticas em desenvolvimento web, design de interface e trabalho em equipe. A experiência adquirida ao longo deste processo será valiosa em projetos futuros e contribuirá para o crescimento profissional dos integrantes envolvidos. O projeto não só atendeu aos objetivos propostos, mas também se destacou pela qualidade do produto final, mostrando que a integração de tecnologias e boas práticas de desenvolvimento é essencial para o sucesso em projetos de software.