



# Estácio

## **Missão Prática | Nível 4 | Mundo 3**

**POLO UNAMAR - CABO FRIO - RJ**  
**DESENVOLVIMENTO FULL STACK**

**Vamos Integrar Sistemas**

**9001**

**2024.3 flex**

**Gerson Ferreira Cotrim Júnior**

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

#### **Objetivos:**

- 1-Implementar persistência com base em JPA.
- 2-Implementar regras de negócio na plataforma JEE, através de EJBs.
- 3-Implementar sistema cadastral Web com base em Servlets e JSPs.
- 4-Utilizar a biblioteca Bootstrap para melhoria do design.
- 5-No final do exercício, o aluno terá criado todos os elementos necessários para exibição e entrada de dados na plataforma Java Web, tornando-se capacitado para lidar com contextos reais de aplicação.

## Códigos desenvolvidos :

### Resultados da execução dos códigos

1- Criar um Servlet com o nome ServletProdutoFC, no projeto CadastroEE-war

```
package com.estacio.cadastroee;
import java.io.IOException;

import com.estacio.persistence.entities.Produto;
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 jakarta.servlet.RequestDispatcher;
import jakarta.ejb.EJB;

@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 || acao.isEmpty()) {
            acao = "listar";
        }
        switch (acao) {
            case "listar":
                request.setAttribute("produtos", facade.findAll());
                destino = "ProdutoLista.jsp";
                break;

            case "formIncluir":
            case "formAlterar":
                destino = "ProdutoDados.jsp";
                if (acao.equals("formAlterar")) {
                    Integer id =
                        Integer.parseInt(request.getParameter("id"));
                    request.setAttribute("produto", facade.find(id));
```

```

        }
        break;

        case "excluir":
            Integer idExcluir =
Integer.parseInt(request.getParameter("id"));
            facade.remove(facade.find(idExcluir));
            request.setAttribute("produtos", facade.findAll());
            destino = "ProdutoLista.jsp";
            break;

        case "alterar":
            Long idAlterar = Long.parseLong(request.getParameter("id"));
            Produto produtoAlterar = facade.find(idAlterar);
            // Preencha o produtoAlterar com os demais campos do request
            aqui
                facade.edit(produtoAlterar);
            request.setAttribute("produtos", facade.findAll());
            destino = "ProdutoLista.jsp";
            break;

        case "incluir":
            Produto produto = new Produto();
            // Preencha o produto com os campos do request aqui
            facade.create(produto);
            request.setAttribute("produtos", facade.findAll());
            destino = "ProdutoLista.jsp";
            break;

        default:
            // Trate outras ações se necessário ou configure uma mensagem
            de erro
                break;
    }

    RequestDispatcher rd = request.getRequestDispatcher(destino);
    rd.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);
}

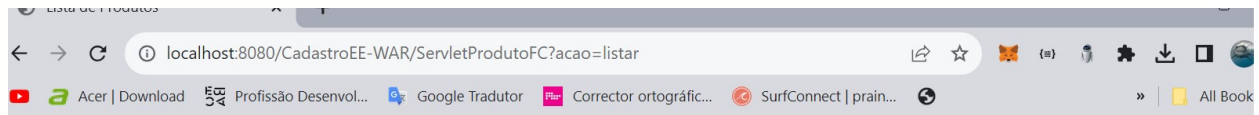
@Override
public String getServletInfo() {
    return "Servlet Produto Front Controller";
}

```

```
}  
}
```

- 2 - Criar a página de consulta, com o nome ProdutoLista.jsp
- 3 - Criar a página de cadastro, com o nome ProdutoDados.jsp

URL : <http://localhost:8080/CadastroEE-WAR/ServletProdutoFC>



## Lista de Produtos

[Novo Produto](#)

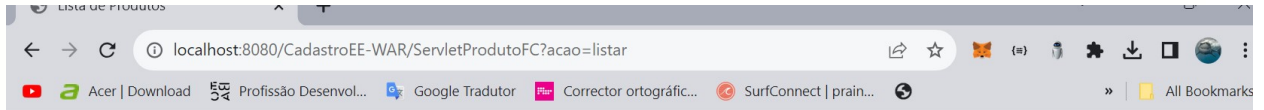
ID	Nome	Quantidade	Preço de Venda	Opções
1	laranjas	30	30.10	<a href="#">Alterar</a>   <a href="#">Excluir</a>
2	maças	20	38.15	<a href="#">Alterar</a>   <a href="#">Excluir</a>
3	pessego	30	22.55	<a href="#">Alterar</a>   <a href="#">Excluir</a>

Message	Summary	Result 1	Profile	Status
	idProduto	nome	quantidade	precoVenda
	1	laranjas	30	30,10
	2	maças	20	38,15
	3	pessego	30	22,55

```
@EJB  
private ProdutoFacadeLocal facade;  
  
2 usages  
protected void processRequest(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
    String acao = request.getParameter(s: "acao");  
    String destino = "";  
  
    if (acao == null || acao.isEmpty()) {  
        acao = "listar";  
    }  
    switch (acao) {  
        case "listar":  
            request.setAttribute(s: "produtos", facade.findAll());  
            destino = "ProdutoLista.jsp";  
            break;  
    }  
}
```

## 4 - Testar as funcionalidades do sistema:

### 4.1 inclusão



#### Lista de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	laranjas	30	30.10	<a href="#">Alterar</a>   <a href="#">Excluir</a>
2	maças	20	38.15	<a href="#">Alterar</a>   <a href="#">Excluir</a>
3	pessego	30	22.55	<a href="#">Alterar</a>   <a href="#">Excluir</a>
4	abacates	10	50.88	<a href="#">Alterar</a>   <a href="#">Excluir</a>



#### Incluir Produto

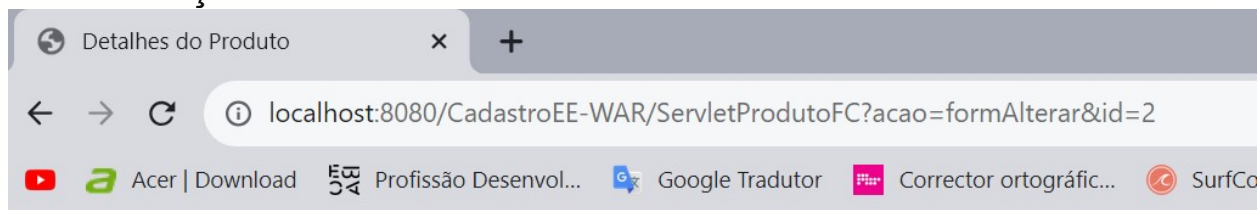
Nome:

Quantidade:

Preço de Venda:

Message	Summary	Result 1	Profile	Status
idProduto	nome	quantidade	precoVe	
1	laranjas	30		
2	maças	20		
3	pessego	30		

## 4.2 alteração



## Alterar Produto

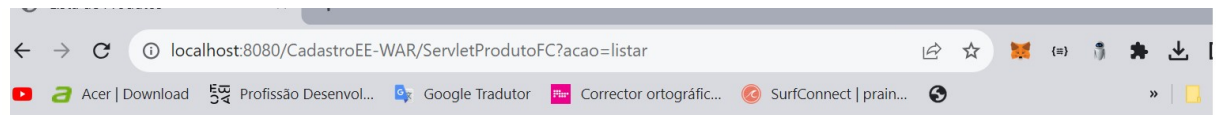
Nome:

Quantidade:

Preço de Venda:

Message	Summary	Result 1	Profile	Status
	idProduto	nome	quantidade	precoVen
▶	1	laranjas	30	
	2	maças argei	20	
	3	pessego	30	
	4	abacates	10	

## 4.3 exclusão



## Lista de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço de Venda	Opções
1	laranjas	30	30.10	<a href="#">Alterar</a>   <a href="#">Excluir</a>
2	maças argentina	20	38.15	<a href="#">Alterar</a>   <a href="#">Excluir</a>
3	pessego	30	22.55	<a href="#">Alterar</a>   <a href="#">Excluir</a>
4	abacates	10	50.88	<a href="#">Alterar</a>   <a href="#">Excluir</a>

Objects \* Untitled - Query

Save Query Builder Beautify SQL Code Snippet

user loja loja Run Stop

```
1 select * from produto
```

Message	Summary	Result 1	Profile	Status
	idProduto	nome	quantidade	precoVenda
▶	1	laranjas	30	30,10

Obs: listagem foi apresentado acima

## Análise e Conclusão:

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

r) Funciona para permitir que cada recurso ou página da web tenha sua própria lógica de controle de entrada, o Front Controller centraliza essa responsabilidade, proporcionando um único ponto de entrada para todas as requisições e são implementados através de JavaServer Faces (JSF) e o Spring MVC que são os frameworks mais populares

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

r) Servlets são geralmente usados para controlar a lógica de negócios, processar dados e gerenciar o fluxo da aplicação enquanto que os JSPs são usados principalmente para a camada de apresentação, renderizando a interface do usuário e exibindo conteúdo dinâmico.

## **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?**

### **Redirecionamento Simples (Response Redirect):**

Quando usamos um redirecionamento, o servidor responde ao navegador com um código de status HTTP 302 Found (redirecionamento temporário) junto com um novo local (Location header) para onde o navegador deve ir. O navegador, então, faz uma nova requisição para esse novo local.

### **Método forward do RequestDispatcher:**

O encaminhamento é realizado internamente no servidor, de um servlet (ou JSP) para outro recurso (servlet, JSP, etc.). O cliente não vê essa transferência e não é ciente dela.

### **Para que servem**



## **parâmetros e atributos nos objetos HttpRequest? (cont)**

r) Fornecem maneiras de passar informações ao longo de uma requisição, mas têm origens e usos diferentes. Parâmetros são tipicamente entradas do cliente, enquanto atributos são usados para comunicação e compartilhamento de dados entre componentes no lado do servidor.