



**UNIVERSIDADE ESTÁCIO DE SÁ**

**DESENVOLVIMENTO FULLSTACK**

## **Mundo 03 - Nível 04**

### **RPG0017 - Vamos integrar sistemas**

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

Ruan Hernandez Finamor Correia

Matrícula 202208175252

Gravataí – RS

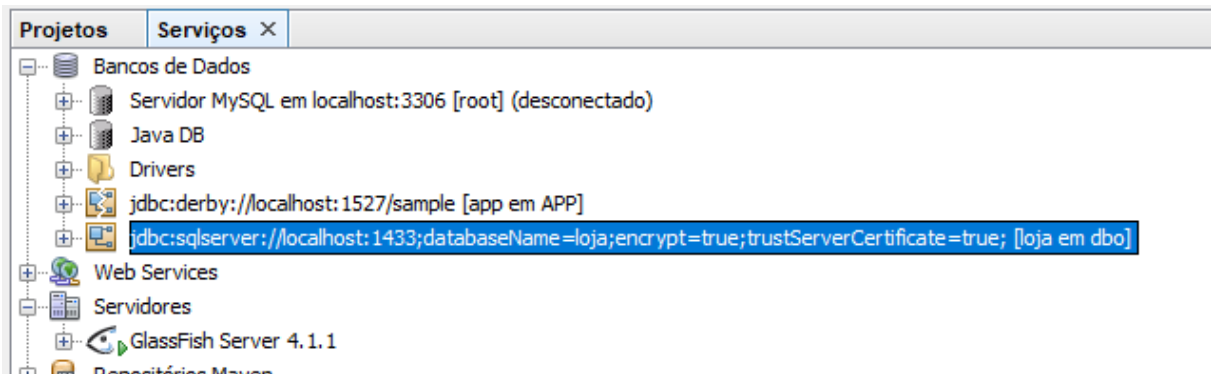
2023

## Objetivo da Prática

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.

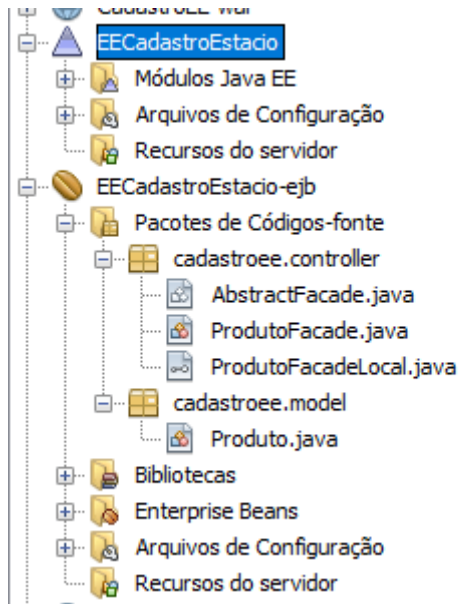
## 1º Procedimento

### Códigos solicitados no roteiro desta aula



Tópico 1, conectando com o banco e o Glassfish.

## Resultado da execução dos códigos



**Exemplos de alguns códigos presentes no trabalho que serão vistos pelo professor:**

```
public class ServletProdutoFC extends Strategy<ProdutoFacadeLocal> {

    public ServletProdutoFC(ProdutoFacadeLocal facade) {
        super(facade);
    }

    @Override
    public String executar(String acao, HttpServletRequest request) {
        String paginaDestino = "ProdutoLista.jsp";

        switch (acao) {
            case "listarOsProdutos":
                listar(request);
                break;
            case "excluirOsProdutos":
                excluir(request);
                listar(request);
                break;
            case "exibirEditar":
                exibirPaginaEditar(request);
                paginaDestino = "EditarProduto.jsp";
                break;
            case "editarOsProdutos":
                alterar(request);
                listar(request);
                break;
            case "incluirOsProdutos":
                incluir(request);
                listar(request);
        }
    }
}
```

```

import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author ruanf
 */
@WebServlet(name = "CadastroFC", urlPatterns = {"/CadastroFC"})
public class CadastroFC extends HttpServlet {

    @EJB
    ProdutoFacadeLocal produtoFacade;

    private final HashMap<String, Strategy> estrategia
        = new HashMap<>();
    private final HashMap<String, String> acoes = new HashMap<>();

    @Override
    public void init() throws ServletException {
        super.init();

        estrategia.put("produto", new ServletProdutoFC(
            produtoFacade));
    }
}

```

## Análise e conclusão:

P – Como é organizado um projeto corporativo no NetBeans?

R – Criar um novo projeto, Selecionar a categoria Java EE, definir o servidor GlassFish, a partir daí criar os pacotes e as classes de entidade.

P – Qual o papel das tecnologias JPA e EJB na construção de um aplicativo para a plataforma Web no ambiente Java

R – O JPA é usado principalmente para lidar com a camada de persistência de dados, já os EJBs gerenciam a lógica.

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

R – Integrando-as diretamente, gerando código automático, edição visual de modelos, assistência de código, entre outros.

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

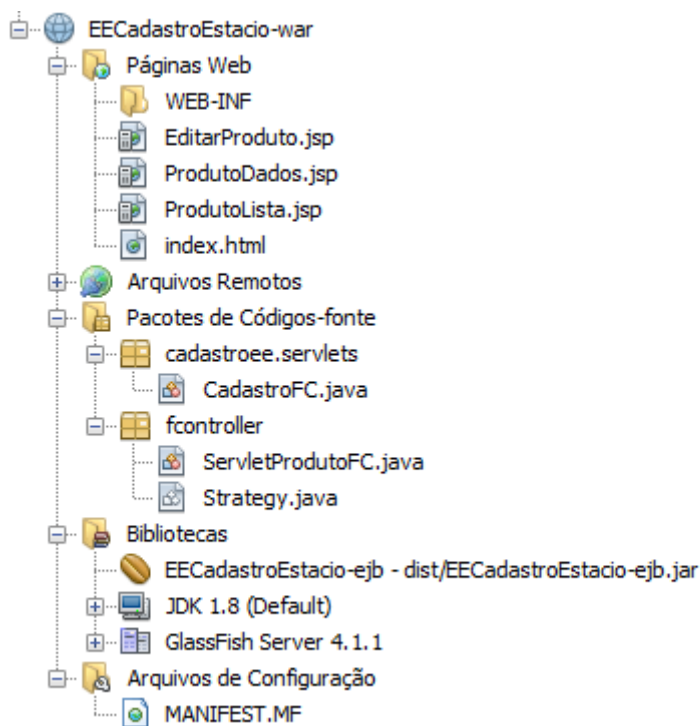
R – Servlets são componentes Java que são usados para criar aplicativos web dinâmicos. Já o NetBeans dá suporte oferecendo assistência desde a criação do projeto até a depuração e implantação em servidores de aplicativos.

P – Como é feita a comunicação entre os Servlets e os Sessions Beans do pool de EJBs

R – Utilizando a injeção de dependência, o que permite o acesso aos métodos e funcionalidades do Session Bean de forma transparente.

## 2º Procedimento

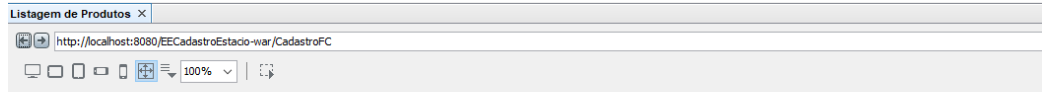
### Códigos solicitados no roteiro desta aula



### Resultado da execução dos códigos

4. Testar as funcionalidades do sistema:

- Listar os produtos com a chamada para o endereço:

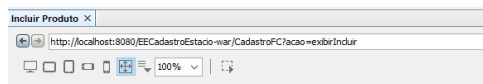


## Listagem de Produtos

[Novo Produto](#)

#	Nome	Quantidade	Preço de Venda	Opções
1008	Melão	3636	4.1241212E7	<a href="#">Alterar</a> <a href="#">Excluir</a>
1010	Melancia	9898	766.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
1011	Pera	56	78.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

- Efetuar uma inclusão:



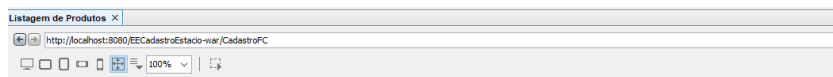
## Dados do Produto

Nome:

Quantidade:

Preço de Venda:

[Incluir Produto](#)



## Listagem de Produtos

[Novo Produto](#)

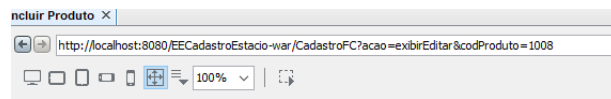
#	Nome	Quantidade	Preço de Venda	Opções
1008	Melão	3636	4.1241212E7	<a href="#">Alterar</a> <a href="#">Excluir</a>
1010	Melancia	9898	766.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
1011	Pera	56	78.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
1012	Jaca	2	300.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

- Efetuar uma alteração

## Listagem de Produtos

[Novo Produto](#)

#	Nome	Quantidade	Preço de Venda	Opções
1008	Melão	3636	4.1241212E7	<a href="#">Alterar</a> <a href="#">Excluir</a> ←
1010	Melancia	9898	766.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
1011	Pera	56	78.0	<a href="#">Alterar</a> <a href="#">Excluir</a>



## Alterar um produto

Nome do Produto:

Quantidade:

Preço Venda:



## Listagem de Produtos

#	Nome	Quantidade	Preço de Venda	Opções
1008	Manga	345	21.0	<input type="button" value="Alterar"/> <input type="button" value="Excluir"/>
1010	Melancia	9898	766.0	<input type="button" value="Alterar"/> <input type="button" value="Excluir"/>

- Excluir um produto:

1010	Melancia	9898	766.0	<input type="button" value="Alterar"/> <input type="button" value="Excluir"/>
1011	Pera	56	78.0	<input type="button" value="Alterar"/> <input type="button" value="Excluir"/>
1012	Jaca	2	300.0	<input type="button" value="Alterar"/> <input type="button" value="Excluir"/>

1010	Melancia	9898	766.0	<input type="button" value="Alterar"/> <input type="button" value="Excluir"/>
1011	Pera	56	78.0	<input type="button" value="Alterar"/> <input type="button" value="Excluir"/>

## Análise e conclusão:

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

R – Ele atua como o ponto central de controle para todas as solicitações de aplicativo web, roteando essas solicitações para os componentes. Ele é implementados por meio de Servlet Front Controller, Controllers, Model, Views, Roteamento e Padrão MVC.

P – Quais as diferenças e semelhanças entre Servlets e JSPs?

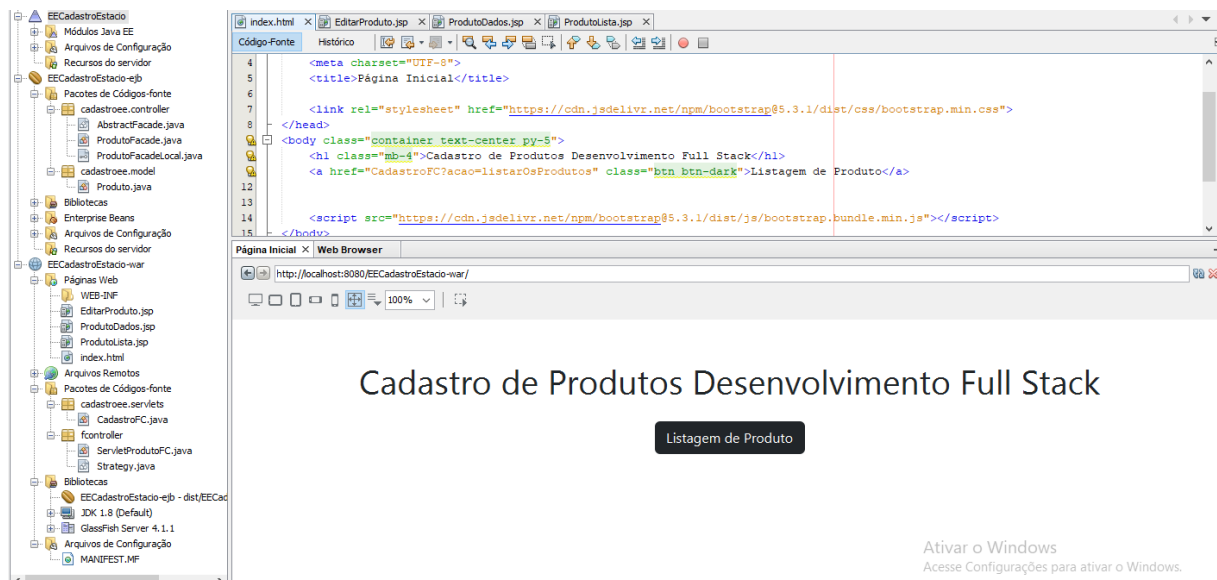
R – Os Servlets são componentes Java que respondem a solicitações HTTP com lógica Java pura, já o JSPs são páginas HTML com código Java embutido. Porém, os dois são Web Java EE, possuem Comunicação com o cliente e bastante flexibilidade.

P – 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 – O redirecionamento simples cria uma nova solicitação independente, já o forward permite compartilhar informações entre recursos no mesmo ciclo de solicitação. Os parâmetros HttpRequest São informações incluídas na URL, já os atributos são objetos armazenados no escopo da solicitação por meio de requests.

### 3º Procedimento

### Códigos e resultados da execução dos códigos solicitados neste roteiro de aula



The image shows a screenshot of an IDE (Eclipse) with two windows. The left window displays the project structure for 'EECadastroEstado', showing a package 'cadastroee' containing 'AbstractFacade.java', 'ProdutoFacade.java', 'ProdutoFacadeLocal.java', 'cadastroee.model', and 'Produto.java'. The right window shows the source code of 'index.html' and its rendered output in a web browser.

**Source Code (index.html):**

```

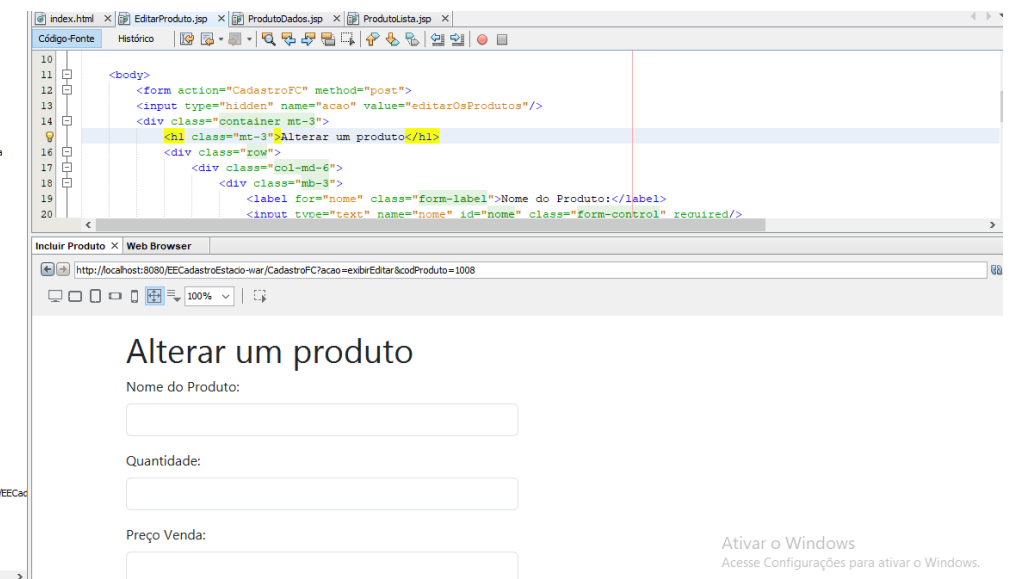
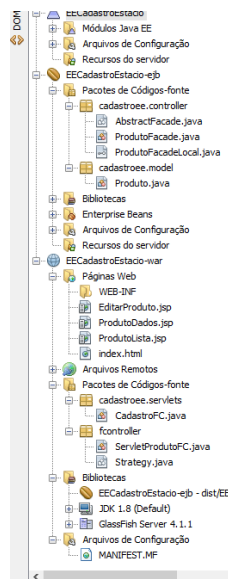
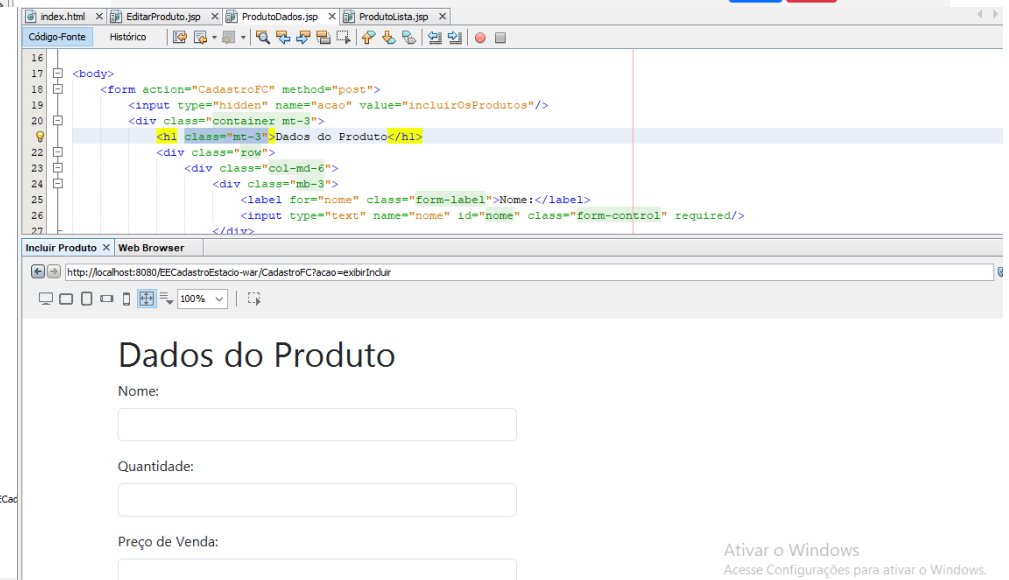
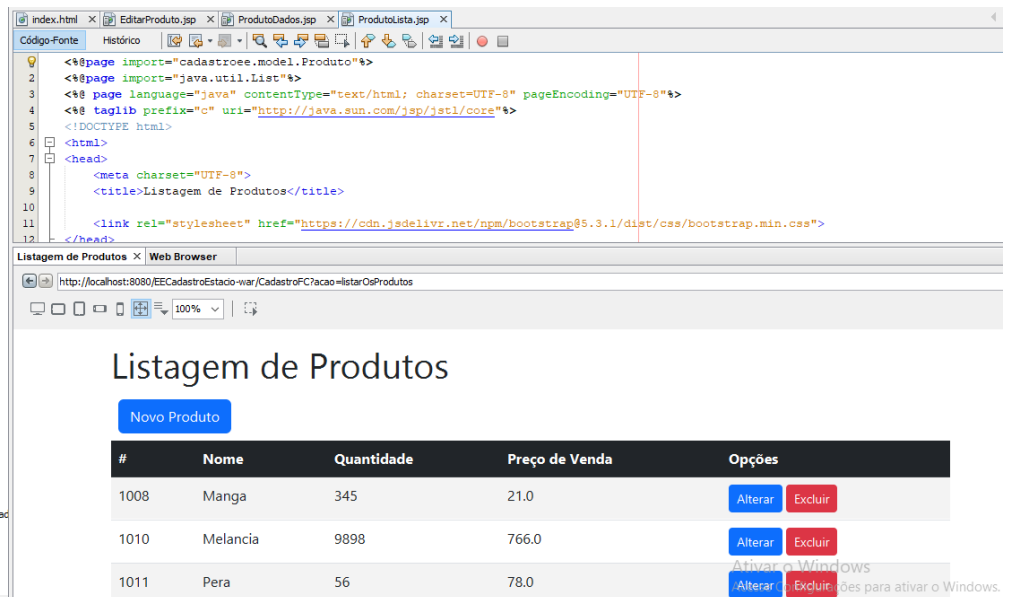
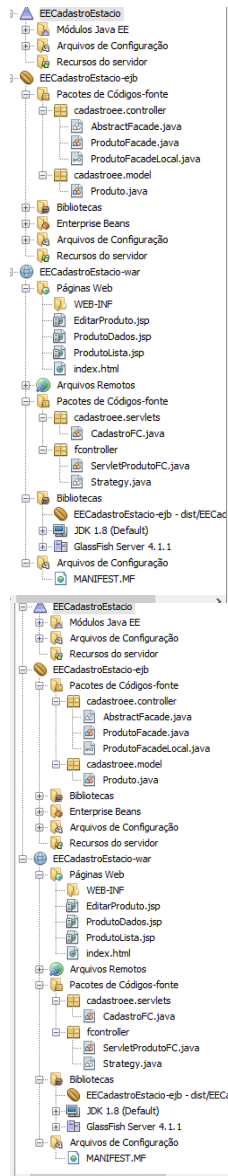
4  <meta charset="UTF-8">
5  <title>Página Inicial</title>
6
7  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css">
8  </head>
9  <body class="container text-center py-5">
10 <h1 class="mb-4">Cadastro de Produtos Desenvolvimento Full Stack</h1>
11 <a href="/CadastroFC?acao=listarOsProdutos" class="btn btn-dark">Listagem de Produto</a>
12
13
14 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"></script>
15 </body>

```

**Rendered Output (Web Browser):**

The browser displays the rendered HTML. The title is 'Página Inicial'. The main heading is 'Cadastro de Produtos Desenvolvimento Full Stack'. Below the heading is a button labeled 'Listagem de Produto'. The browser address bar shows 'http://localhost:8080/EECadastroEstado-war/'.





**Análise e conclusão:**

P – Como o framework Bootstrap é utilizado?

R – O Bootstrap integra em suas bibliotecas o CSS e o JavaScript, criando um design atrativo, ele oferece uma maneira eficiente e rápida de desenvolver interfaces web. Basta fazer a importação dele e utilizar suas classes no projeto para utilizá-lo.

P – Por que o Bootstrap garante a independência estrutural do HTML?

R – Ele fornece classes CSS reutilizáveis e componentes prontos para uso, facilitando a estilização e a adição de funcionalidades sem a necessidade de alterações significativas na estrutura HTML.

P – Qual a relação entre o Bootstrap e a responsividade da página?

R – Ele fornece ferramentas e recursos que facilitam o desenvolvimento de layouts e componentes que se ajustam automaticamente a diferentes dispositivos e tamanhos de tela.