



# Estácio

## RPG0017 - Vamos integrar Sistemas

202303832338 - Rafael Rosendo Tagliaferro

**Campos Interlagos**

### Objetivos da prática

Implementar persistência com base em JPA.

1. Implementar regras de negócio na plataforma JEE, através de EJBs.
2. Implementar sistema cadastral Web com base em Servlets e JSPs.
3. Utilizar a biblioteca Bootstrap para melhoria do design.
4. 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.

### Análise e Conclusão:

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

R: Através dos domínios JPA, EJB e WEB, onde no JPA fica as entidades e o gerenciamento de dados, Web contendo os servlets, html e css, e o EJB sendo a camada de negócio.

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

R: JPA seria para o gerenciamento / persistência de dados no Java. Simplificando o processo na conexão e manipulação de dados (banco de dados). O EJB encapsula a lógica do negócio, podendo inclusive ser reutilizado em outras aplicações.

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

R: Por ser uma interface gráfica, ele facilita e muito a compreensão do código e organização do mesmo, além de depender menos de linhas de comando, além de mostrar os logs de erro, execução e compilação para o projeto. Para os JPA e EJB, ele facilita pois você pode importar todas as classes diretamente do banco, e ele já cria as interfaces e entidades automaticamente para você.

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



R: Servlets são basicamente aplicativos que são tratados de forma separada, como se fossem componentes, que podem ser reutilizados, dando como resposta html ou arquivos json.

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

R: Ocorre através da injeção de dependência para injetar diretamente um session bean, para conseguir acessar as lógicas do negócio ou banco de dados.

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

### **Análise e Conclusão**

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

R: O front controller tem como objetivo centralizar todas as chamadas em um único ponto, do ponto de vista do MVC, que já tem como premissa ter um controller para cada página ou recurso, o front controller cai muito bem.

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

R: Acredito que as principais sejam ter o JAVA como linguagem e que ambas suportam requisições HTTP, sendo que JSP está mais focado na apresentação dos dados e os servlets, são componentes que podem ser reutilizados.

**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?**

R: a maior diferença está no estado da página, o forward mantém o estado da requisição e não altera a url do navegador, enquanto o redirecionamento simples cria um novo ciclo de requisição e não mantém os estados dos objetos.

## **3º Procedimento | Melhorando o designer da interface**



# Estácio

## Análise e Conclusão:

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

R: Basta importar sua classe css e/ou js através do cdn na classe header do html

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

R: Porque o bootstrap é o estilo da página, não manipula diretamente objetos html, quando estivermos falando de folhas de estilo (css), sendo assim, o bootstrap apenas manipula os estilos e não a tag.

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

R: A responsabilidade para a responsividade da página é feita através de comando na folha de estilo (css), que visam adicionar várias camadas e tamanhos para diferentes tamanhos de monitores/celular/tablets. Então é possível criar um estilo personalizado para tipo de aparelho e/ou tamanho.

## Print – 1º Procedimento :



## Prints – 2,3 Procedimentos



localhost:8080/CadastroEE-war/ServletProdutoFC?acao=excluir&id=5

## Lista de Produtos

Incluir Novo Produto

ID	Nome	Quantidade	Preço	Ações
1	Bananas	1000	8.0	<a href="#">Alterar</a> <a href="#">Excluir</a>
2	Laranja	500	2.0	<a href="#">Alterar</a> <a href="#">Excluir</a>

localhost:8080/CadastroEE-war/ServletProdutoFC?acao=formAlterar&id=1

## Alterar Produto

Nome:

Bananas

Quantidade:

1000

Preço de Venda

8.0

[alterar](#)

[Voltar à Lista de Produtos](#)

## Códigos

### ServletProdutoFC.java

```
/*
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this
template
```

```
*/
```

```
package cadastroee.servlets;
```

```
import cadastroee.controller.ProdutosFacadeLocal;
```



```
import cadastroee.model.Produtos;

import jakarta.ejb.EJB;

import jakarta.servlet.RequestDispatcher;

import java.io.IOException;

import java.io.PrintWriter;

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.util.List;

/**

 *

 * @author rafae

 */

public class ServletProdutoFC extends HttpServlet {

    @EJB

    private ProdutosFacadeLocal facade;

    /**

     * Processes requests for both HTTP GET and POST

     * methods.

     *
```



- \* @param request servlet request
- \* @param response servlet response
- \* @throws ServletException if a servlet-specific error occurs
- \* @throws IOException if an I/O error occurs
- \*/

protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

```
{  
  
    String acao = request.getParameter("acao");  
  
    String destino = "";  
  
    System.out.println("aki: " + acao);  
  
    if ("listar".equals(acao)) {  
        List<Produtos> produtos = facade.listarProdutos();  
        request.setAttribute("produtos", produtos);  
        destino = "ProdutoLista.jsp";  
    }  
  
    else if ("formIncluir".equals(acao))  
    {  
        destino = "ProdutoDados.jsp";  
    }  
  
    else if ("formAlterar".equals(acao)) {  
        int id = Integer.parseInt(request.getParameter("id"));
```



```
Produtos produto = facade.buscarProduto(id);

request.setAttribute("produto", produto);

destino = "ProdutoDados.jsp";

} else if ("excluir".equals(acao)) {

    Integer id = Integer.parseInt(request.getParameter("id"));

    facade.excluir(id);

    List<Produtos> produtos = facade.listarProdutos();

    request.setAttribute("produtos", produtos);

    destino = "ProdutoLista.jsp";

} else if ("alterar".equals(acao)) {

    Integer id = Integer.parseInt(request.getParameter("id"));

    Produtos produto = facade.buscarProduto(id);

    //System.out.println("aki2: " + produto.getProdutoID());

    produto.setNome(request.getParameter("nome"));

    produto.setPrecoVenda(Float.parseFloat(request.getParameter("precoVenda")));

    produto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));

    produto.setProdutoID(Integer.parseInt(request.getParameter("id")));

    facade.alterar(produto);

    List<Produtos> produtos = facade.listarProdutos();

    request.setAttribute("produtos", produtos);

    destino = "ProdutoLista.jsp";

} else if ("incluir".equals(acao)) {
```



```
Produtos produto = new Produtos();

produto.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));

produto.setPrecoVenda(Float.parseFloat(request.getParameter("precoVenda")));

produto.setNome(request.getParameter("nome"));

facade.incluir(produto);

List<Produtos> produtos = facade.listarProdutos();

request.setAttribute("produtos", produtos);

destino = "ProdutoLista.jsp";

}
```

```
RequestDispatcher dispatcher = request.getRequestDispatcher(destino);

dispatcher.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 para gerenciamento de produtos usando o padrão Front
Controller"; // Informação do Servlet

}

}
```

## ProdutoDados.jsp

```
<% @page import="cadastroee.model.Produtos"%>

<% @ page contentType="text/html; charset=UTF-8" language="java" %>

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwI
H" crossorigin="anonymous">

    <title>Cadastro de Produto</title>

    <link rel="stylesheet" type="text/css" href="style.css">

</head>

<body class="container">

    <h1>${acao == 'incluir' ? 'Incluir Produto' : 'Alterar Produto'}</h1>
```



```
<form action="ServletProdutoFC" method="post" class="form">

    <input type="hidden" name="acao" value="<%= (request.getAttribute("produto")
== null) ? "incluir" : "alterar" %>"/>

    <input type="hidden" name="id" value="<%= (request.getAttribute("produto") !=
null) ? ((Produtos) request.getAttribute("produto")).getProdutoID(): "" %>"/>

    <div class="mb-3">

        <label class="label">Nome:</label>

        <input type="text" class="form-control" name="nome" value="<%=
(request.getAttribute("produto") != null) ? ((Produtos)
request.getAttribute("produto")).getNome() : "" %>"/>

    </div>

    <div class="mb-3">

        <label class="label">Quantidade:</label>

        <input type="number" class="form-control" name="quantidade" value="<%=
(request.getAttribute("produto") != null) ? ((Produtos)
request.getAttribute("produto")).getQuantidade() : "" %>"/>

    </div>

    <div class="mb-3">

        <label class="label">Preço de Venda</label>

        <input type="text" class="form-control" name="precoVenda" value="<%=
(request.getAttribute("produto") != null) ? ((Produtos)
request.getAttribute("produto")).getPrecoVenda(): "" %>"/>

    </div>

    <input type="submit" class="btn btn-primary" value="<%=
(request.getAttribute("produto") == null) ? "incluir" : "alterar" %>"/>
```



</form>

<a href="ServletProdutoFC?acao=listar">Voltar à Lista de Produtos</a>

</body>

</html>

## ProdutoLista.jsp

<% @ page contentType="text/html; charset=UTF-8" language="java" %>

<% @ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html>

<head>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH" crossorigin="anonymous">

<title>Lista de Produtos</title>

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Link para um arquivo CSS, se necessário -->

</head>

<body class="container">

<h1>Lista de Produtos</h1>

<a href="ServletProdutoFC?acao=formIncluir" class="btn btn-primary m-2">Incluir Novo Produto</a> <!-- Link para inclusão -->

<table border="1" class="table table-striped">



```
<thead class="table-dark">

  <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}"> <!-- Recupera a lista de
produtos -->

    <tr>

      <td>${produto.produtoID}</td> <!-- Exibe o ID do produto -->

      <td>${produto.nome}</td> <!-- Exibe o nome do produto -->

      <td>${produto.quantidade}</td> <!-- Exibe a quantidade -->

      <td>${produto.precoVenda}</td> <!-- Exibe o preço -->

      <td>

        <a href="ServletProdutoFC?acao=formAlterar&id=${
{produto.produtoID}}" class="btn btn-primary btn-sm">Alterar</a> | <!-- Link para
alterar -->

        <a href="ServletProdutoFC?acao=excluir&id=${produto.produtoID}"
class="btn btn-danger btn-sm">Excluir</a> <!-- Link para excluir -->

      </td>

    </tr>

  </tbody>

</table>
```



```
</c:forEach>

</tbody>

</table>

</body>

</html>
```

ServletProduto.java

```
/*

 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license

 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this
template

 */

package cadastroee.servlets;

import jakarta.ejb.EJB;
import cadastroee.controller.ProdutosFacadeLocal;
import cadastroee.model.Produtos;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.util.List;
```



```
/**
 *
 * @author rafae
 */

public class ServletProduto extends HttpServlet {

    @EJB

    private ProdutosFacadeLocal facade; // Declara a variável facade como um EJB

    /**
     * Processes requests for both HTTP GET and POST
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */

    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter())
        {

            List<Produtos> produtos = facade.listarProdutos();
```



```
        out.println("<html><body>");

        out.println("<h1>Lista de Produtos</h1>");

        out.println("<ul>");

        for (Produtos produto : produtos) {

            out.println("<li>" + produto.getNome() + "</li>");

        }

        out.println("</ul>");

        out.println("</body></html>");

    }

}

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    processRequest(request, response);

}

/**
```



\* Handles the HTTP `POST` method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
    throws ServletException, IOException {
```

```
        processRequest(request, response);
```

```
}
```

```
/**
```

```
 * Returns a short description of the servlet.
```

```
 *
```

```
 * @return a String containing servlet description
```

```
*/
```

@Override

```
public String getServletInfo() {
```

```
    return "Short description";
```

```
// </editor-fold>
```

```
}
```