



# Estácio

Faculdade Estácio - Polo Centro - Canela - RS

Curso: Desenvolvimento Full Stack

Disciplina: RPG0017 - Vamos Integrar Sistemas

Turma: 9001 - Semestre Letivo: 2025.1 - 3º semestre

Integrante: Pedro Henrique Marques Medeiros Pinho

Matrícula: 202402031831

IDE: Sql Server Management Studio

Repositório Git: <https://github.com/PedroPinho23/Vamos-Integrar-Sistemas.git>

# Vamos Integrar Sistemas

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

## Objetivos da prática

- Implementar persistência com base em JPA.
- Implementar regras de negócio na plataforma JEE, através de EJBs.
- Implementar sistema cadastral Web com base em Servlets e JSPs.
- Utilizar a biblioteca Bootstrap para melhoria do design.
- 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 Utilizados

### 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 cadastroee.controller.ProdutoFacadeLocal;  
import jakarta.ejb.EJB;  
import java.io.IOException;  
import java.io.PrintWriter;  
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 cadastroee.model.Produto;
/**
 *
 * @author Pedro
 */
@WebServlet("/ServletProduto")
public class ServletProduto extends HttpServlet {
    @EJB
    ProdutoFacadeLocal facade;
    /**
     * Processes requests for both HTTP <code>GET</code> and
     * <code>POST</code>
     * 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()) {
            /* TODO output your page here. You may use following sample code.
            */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet ServletProduto</title>");
            out.println("</head>");
            out.println("<body>");
            // out.println("<h1>Servlet ServletProduto at " +
            request.getContextPath() + "</h1>");
            //out.println(facade.findAll().getClass());
            //out.println(facade.find(1).getClass());
            for (Produto p : facade.findAll()) {
                out.println("<li>" + p.getNome() + "</li>");
            }
            out.println("</body>");

```

```

out.println("</html>");
}
}
// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click
on the + sign on the left to edit the code.">
/**
 * 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 <code>POST</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 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>  
}
```

## Resultados:

Foram implementadas e testadas as funcionalidades de persistência com JPA, regras de negócio via EJBs e interface web utilizando Servlets, JSPs e Bootstrap. Os testes confirmaram que os dados foram corretamente armazenados, recuperados e manipulados no banco SQL Server, além de garantir a correta interação entre as camadas da aplicação, comprovando o funcionamento integrado da plataforma Java EE.

## Conclusão:

### Como é organizado um projeto corporativo no NetBeans?

Um projeto corporativo geralmente é organizado em camadas: uma para a interface, como Servlets e JSPs, outra para a lógica de negócio EJBs e outra para a persistência dos dados JPA. Tudo fica separado para facilitar o desenvolvimento e a manutenção da aplicação.

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

A JPA cuida de gerenciar o acesso e a persistência dos dados no banco, facilitando trabalhar com objetos em vez de SQL direto. Já o EJB organiza e executa as regras de negócio, garantindo que a lógica da aplicação seja eficiente, segura e escalável dentro da plataforma Java Web.

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

O NetBeans ajuda na produtividade oferecendo recursos como geração automática de código, assistentes para criação de entidades JPA e EJBs, integração com o servidor de aplicação e suporte a validação em tempo real, o que agiliza o desenvolvimento e reduz erros.

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

Servlets são componentes Java que processam requisições HTTP e geram respostas, geralmente usadas para controlar o fluxo de dados entre o cliente e o servidor em aplicações web. O NetBeans oferece suporte criando a estrutura do servlet automaticamente, com templates prontos, mapeamento no web.xml ou via anotações, além de permitir testes rápidos com integração ao servidor de aplicação.

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

A comunicação é feita por meio de injeção de dependência ou lookup via JNDI. O servlet chama métodos dos Session Beans, que estão no pool de EJBs, para executar regras de negócio. Essa integração permite que a lógica fique centralizada nos EJBs, enquanto o servlet cuida das requisições do usuário.