



## **RPG0017 - Vamos integrar sistemas**

**Paulo Wuéliton Horacio Fernandes (202302025919)**

**507 Polo Tucuruvi – São Paulo/SP**

**Nível 4: Vamos Integrar Sistemas– 2024.1 FLEX - 3º Semestre Letivo**

### **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 | Camadas de Persistência e Controle**

- a) Como é organizado um projeto corporativo no NetBeans?
- Um projeto corporativo é dividido em três camadas, utilizando o padrão MVC. Ao iniciar um projeto corporativo no NetBeans ele irá gerar três projetos, um projeto responsável pelo consumo e persistência de dados em um banco de dados, que seria a camada Model da aplicação, essa camada fornece todos os métodos para comunicação com o banco de dados; o segundo projeto EJB possui a responsabilidade de implementar as regras de negócios da aplicação, atuando como a camada Controller, além de fornecer e receber as solicitações da camada de visualização e por último a camada de View, que é responsável por fornecer toda a interface da aplicação para o cliente Web.

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

As duas tecnologias são cruciais para o desenvolvimento de aplicativos web no Java, o JPA permite abstrair a consulta e persistência dos dados em um banco de dados, permitindo o mapeamento objeto-relacional (ORM) em tabelas de bancos de dados relacionais, especificação de relacionamentos entre as tabelas com anotações Java e o consumo de diversos fornecedores de banco de dados com a mesma interface, e, o EJB fornece um conjunto pronto de componentes que agiliza o desenvolvimento de soluções, seu objetivo é fornecer todo o código de infraestrutura para que o desenvolvedor se concentre apenas no desenvolvimento das regras de negócios da aplicação.

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

O NetBeans fornece diversas ferramentas que agilizam e auxiliam na criação de projetos com tecnologias JPS e EJB, desde a criação do projeto, que possui uma estrutura que abstrai a infraestrutura necessária para o funcionamento dos componentes, além de fornecer ferramentas que permitem o mapeamento das entidades do banco de dados de forma automática, integração com diversos servidores que facilita o desenvolvimento, execução e depuração das aplicações. Todas essas ferramentas permitem que o desenvolvedor tenha foco apenas nas regras de negócios.

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

Os Servlets são componentes que estendem as funcionalidades de servidores Web, ficam responsáveis por processar e responder as requisições HTTP. O NetBeans oferece suporte robusto para construção dos Servlets através de suas ferramentas. Na criação de um novo projeto é possível, através de seu

assistente, criar e personalizar um servidor integrado ao projeto a partir de um modelo pré-definido, durante o processo de desenvolvimento ele fornece ferramentas de execução e depuração do servidor.

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

A comunicação é realizada através da injeção de dependências, no Java EE os Session Beans podem ser injetados nos Servlets utilizando a anotação @EJB, desta forma é injetada uma referência ao Session Bean no momento da inicialização, que permite que o Servlet acesse os métodos do Session Bean diretamente, como se fossem métodos locais.

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

- a) Como funciona o padrão Front Controller, e como ele é implementado em um aplicativo Web Java, na arquitetura MVC?
- No padrão Front Controller, todas as requisições do cliente são direcionadas para um único ponto de entrada, que é o componente Front Controller, ele fica responsável por analisar cada requisição, determinar qual componente deve responder a solicitação e despachar a requisição para o componente, após o processamento, o Front Controller coordena a resposta à solicitação do cliente. Em um aplicativo Web Java, o Front Controller fica na camada View da aplicação, geralmente implementado como um Servlet que é mapeado para uma URL específica que atua como ponto de entrada central para todas as requisições do cliente, o Servlet fica responsável por analisar a requisição, determinar qual controller (EJB) deve processar a requisição e, após o processamento, deve responder a solicitação do cliente, seja com o

redirecionamento para uma View específica ou enviando uma resposta em HTML/JSON/XML.

b) Quais as diferenças e semelhanças entre Servlets e JSPs?

Os Servlets possuem um objetivo diferente dos JSPs, eles são responsáveis por ser o ponto central de recebimento e processamento das requisições de um servidor web, diferente dos JSPs, que são páginas que contém código HTML e Java responsável pela camada de apresentação (View) do conteúdo. Os dois possuem em comum a possibilidade de retornar código HTML como resposta a solicitação, assim como são executados do lado servidor, onde possuem acesso aos mesmos objetos de solicitação (`HttpServletRequest`) e de resposta (`HttpServletResponse`).

c) 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`?

No redirecionamento simples, o Servlet responde a requisição com um redirecionamento que deverá ser executado pelo navegador ao receber a resposta, neste caso a nova solicitação não tem vínculo com a requisição original, no caso do método `forward` o redirecionamento é feito internamente pelo servidor, onde a solicitação pode ser enviada para um outro Servlet, uma página HTML ou JSP, esse segundo método permite o compartilhamento de informações por meio de parâmetros e/ou atributos. Os parâmetros e atributos nos objetos `HttpRequest` permitem que o Servlet adicione informações adicionais necessárias para o processamento e criação da View solicitada pelo usuário.

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

a) Como o framework Bootstrap é utilizado?

O framework Bootstrap é utilizado por meio de classes que possuem predefinições, essas classes podem ser aplicadas aos elementos HTML que irão definir a interface que será apresentada para o usuário que independem da estrutura do HTML, pois a sua maioria não depende de elementos específicos ou estruturas específicas para serem aplicadas. No projeto ela foi utilizada para tornar a interface mais amigável para o usuário, adicionando por meio de classes a responsividade da aplicação e caracterização dos elementos, como tabela, formulários e botões de ação de forma que os tornam mais fáceis de reconhecimento e entendimento das ações para o usuário da interface.

b) Por que o Bootstrap garante a independência estrutural do HTML?

Suas classes de personalização, em sua maioria, não dependem de uma estrutura ou elemento HTML específico, onde uma classe de botão pode ser aplicada a qualquer elemento HTML, além de possuir uma grande quantidade de predefinições visuais responsivas, assim como o Grid responsivo, que permite desenvolver a interface com a reutilização de classes.

c) Qual a relação entre o Bootstrap e a responsividade da página?

O Bootstrap é construído considerando o conceito de mobile first, que possui o foco em desenvolver interfaces primeiro para dispositivos móveis, e só então, a partir deles adaptar os elementos para dispositivos maiores, visto que hoje em dia mais de 60% dos dispositivos em uso são celulares. A maioria de seus componentes predefinidos possuem um comportamento responsivo, que se adaptam ao tamanho disponível, mas também é possível usar o sistema de Grid e outras classes auxiliares do Bootstrap para definir como serão os comportamentos em cada tamanho de tela, chamados de Breakpoints; na maioria de suas classes é possível utilizar sufixos como sm, md, lg, etc... que irão determinar quando aquela personalização deverá ser aplicada de acordo com o tamanho de tela do dispositivo do usuário.