

 Estácio	<p align="center"> UNIVERSIDADE ESTÁCIO DE SÁ POLO INDAIATUBA – INDAIATUBA/SP DESENVOLVIMENTO FULL STACK - 22.3 Relatório da Missão Prática Nível 4 Mundo 3 </p>
Aluno:	Saulo Henrique dos Santos
Professor:	Jhonatan Alves
Repositório:	https://github.com/SauloHenriqueSantos/Mundo03Nivel04

Objetivo:

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
6. exibição e entrada de dados na plataforma Java Web, tornando-se capacitado para
7. lidar com contextos reais de aplicação.

[1º Procedimento | Camadas de Persistência e Controle](#)

Análise e Conclusão:

Como é organizado um projeto corporativo no NetBeans?

Um projeto corporativo no NetBeans é organizado em módulos, onde cada módulo é um projeto separado que pode ser compilado e implantado independentemente. O

NetBeans suporta a criação de projetos corporativos Java EE, que são projetos que usam a plataforma Java Enterprise Edition para desenvolver aplicativos corporativos.

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

A Java Persistence API (JPA) é um framework leve, baseado em POJOS (Plain Old Java Objects) para persistir objetos Java. A JPA oferece diversas funcionalidades essenciais em qualquer aplicação corporativa, além de ser uma solução para o mapeamento objeto-relacional (ORM). Já os Enterprise JavaBeans (EJBs) são responsáveis por facilitar o encapsulamento e compartilhamento da lógica do negócio. Eles permitem que os componentes sejam como Plain Java Objects ou POJOS, o que torna os Enterprise JavaBeans mais fáceis e mais produtivos de usar.

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

O NetBeans oferece suporte integrado para Java Persistence API (JPA) e Enterprise JavaBeans (EJB), simplificando o desenvolvimento e melhorando a produtividade. Ele fornece ferramentas visuais para criar entidades JPA, mapeamentos de banco de dados e consultas, facilitando o desenvolvimento de aplicações de persistência. Para EJB, o NetBeans oferece assistentes para criar facilmente sessões, entidades e beans de mensagem, além de fornecer suporte para implementações EJB. Com recursos como auto-completar, depuração e geração automática de código, o NetBeans ajuda os desenvolvedores a escrever código mais eficientemente ao trabalhar com JPA e EJB.

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 estendem a capacidade de servidores web para gerar conteúdo dinâmico. Eles respondem a solicitações HTTP, processam-nas e podem gerar uma resposta dinâmica, como páginas da web. O NetBeans oferece suporte à construção de Servlets em projetos web por meio de assistentes e modelos que simplificam a criação, implementação e configuração de Servlets. O ambiente integrado do NetBeans facilita o desenvolvimento, teste e depuração desses componentes em projetos web Java.

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

A comunicação entre Servlets e Session Beans do pool de EJBs geralmente é realizada por meio de chamadas de método remoto. Os Servlets podem invocar métodos em Session Beans, que são componentes de lógica de negócios, para realizar operações específicas. Essa comunicação é facilitada pelos serviços de contêiner EJB, que gerenciam a criação, destruição e pooling dos Session Beans. A passagem de parâmetros e a obtenção de resultados são tratadas automaticamente pelo contêiner EJB, proporcionando uma integração eficiente entre a camada de apresentação (Servlets) e a camada de negócios (Session Beans).

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

O padrão Front Controller é um design pattern que centraliza o controle do fluxo de uma aplicação web. Na arquitetura MVC (Model-View-Controller), o Front Controller recebe todas as requisições do cliente, coordena a lógica de processamento, e decide qual controlador (Controller) acionar com base na requisição. Isso ajuda a modularizar o código e centralizar o gerenciamento do fluxo de controle em um único ponto, simplificando a manutenção e melhorando a reusabilidade. Em um aplicativo web Java, o Front Controller pode ser implementado por meio de um servlet que atua como ponto de entrada central, recebendo e distribuindo as requisições para os controladores apropriados na camada de controle (Controller) do padrão MVC.

Quais as diferenças e semelhanças entre Servlets e JSPs?

Servlets e JSPs (JavaServer Pages) são tecnologias Java para desenvolvimento web e frequentemente trabalham juntas em aplicações. Semelhanças: Java: Ambos são baseados em Java, permitindo que os desenvolvedores usem Java para criar conteúdo dinâmico. Server-side: Ambos são executados no lado do servidor, gerando conteúdo dinâmico que é enviado para o cliente. Diferenças: Estilo de Desenvolvimento: Servlets são mais orientados a programação Java, enquanto JSPs permitem a mescla de código Java e marcação HTML, facilitando a integração com designers web. Estrutura: Servlets são classes Java puras que lidam com lógica de controle e processamento, enquanto JSPs são páginas HTML com marcações especiais (tags JSP) que incorporam lógica Java. Manutenção: Geralmente, Servlets são considerados mais poderosos para lógica de controle complexa, mas JSPs são mais convenientes para desenvolvimento web rápido e manutenção, especialmente quando se trata de separar a lógica de apresentação do código Java.

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: Ocorre quando o servidor envia uma resposta de redirecionamento para o navegador, que então solicita a nova página. O URL no navegador muda para a nova página. Isso envolve uma nova solicitação do cliente. **Método Forward:** Envolvendo o RequestDispatcher, o método forward ocorre no lado do servidor sem a participação do navegador. A solicitação é encaminhada internamente para outro recurso (como Servlet ou JSP) no servidor, mantendo o mesmo URL no navegador. **Parâmetros e Atributos no objeto HttpRequest:** **Parâmetros:** São informações incluídas na URL ou no corpo da solicitação HTTP. Em Java, podem ser acessados por `request.getParameter("nomeDoParametro")`. São usados para transmitir dados entre o cliente e o servidor. **Atributos:** São objetos associados ao escopo da solicitação, sessão ou aplicação. Eles podem ser definidos no request, session ou application e são usados para armazenar dados que precisam persistir durante o ciclo de vida da solicitação, sessão ou aplicação. **Finalidade:** **Parâmetros:** São usados para passar informações de uma solicitação para outra, como dados de formulário. **Atributos:** São usados para compartilhar dados entre diferentes partes de uma aplicação no mesmo escopo (solicitação, sessão, aplicação). Eles são úteis quando você precisa que os dados persistam por mais tempo do que apenas uma solicitação.

Como o framework Bootstrap é utilizado?

Para utilizá-lo, você inclui as bibliotecas Bootstrap em seu projeto, referenciando os arquivos CSS e JavaScript. Ao adicionar classes CSS específicas do Bootstrap aos elementos HTML, você aplica estilos prontos e utiliza componentes como botões, formulários, barras de navegação, entre outros.

Por que o Bootstrap garante a independência estrutural do HTML?

O Bootstrap garante a independência estrutural do HTML ao separar a camada de apresentação da camada de conteúdo. Ele fornece estilos e componentes prontos, permitindo que desenvolvedores utilizem classes CSS específicas para aplicar formatação visual sem a necessidade de criar estilos personalizados. Isso simplifica o desenvolvimento, mantendo a estrutura semântica do HTML intacta e facilitando a manutenção e atualização do código.

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

O Bootstrap contribui significativamente para a responsividade das páginas, oferecendo um sistema de grid flexível e componentes adaptáveis. Com sua grade responsiva, é possível criar layouts que se ajustam automaticamente a diferentes tamanhos de tela, proporcionando uma experiência consistente em dispositivos variados, desde desktops até dispositivos móveis. Além disso, os componentes do Bootstrap são projetados para serem sensíveis, facilitando a construção de interfaces que se adaptam de forma eficiente a diferentes resoluções.