



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

Missão Prática | Nível 4 | Mundo 3

- **Aluna:** Amanda Duque Kawauchi
- **Matrícula:** 202209170254
- **Campus:** Morumbi
- **Curso:** Desenvolvimento Full-Stack
- **Disciplina:** Nível 4: Vamos Integrar Sistemas
- **Turma:** 2023.3
- **Semestre Letivo:** 3º Semestre
- **Repositório GitHub:** [Link do Repositório GitHub](#)

Objetivo da Prática

O objetivo da prática é dominar a implementação de persistência de dados usando Java Persistence API (JPA), desenvolver regras de negócios com Enterprise JavaBeans (EJBs), e criar um sistema cadastral web utilizando Servlets e JavaServer Pages (JSPs). Além disso, busca-se aprimorar a interface do usuário com o uso da biblioteca Bootstrap, culminando na capacidade de aplicar esses conhecimentos em situações reais de desenvolvimento de aplicações na plataforma Java Web.

Análise e Conclusão

1º Procedimento | Camadas de Persistência e Controle

a. Como é organizado um projeto corporativo no NetBeans?

Um projeto corporativo no NetBeans é organizado em uma estrutura modular que facilita a gestão de diferentes camadas de uma aplicação JEE. No NetBeans, é possível criar um projeto que encapsula tanto a lógica de negócios quanto a interface de usuário em subprojetos distintos, facilitando o gerenciamento do código e a separação de responsabilidades. O ambiente do NetBeans oferece ferramentas integradas que auxiliam desde a configuração do servidor de aplicação até o mapeamento das entidades de banco de dados, resultando em um desenvolvimento mais ágil e organizado.

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

As tecnologias Java Persistence API (JPA) e Enterprise JavaBeans (EJB) são fundamentais na construção de aplicativos para a plataforma web. JPA (Java Persistence API) é usada para gerenciar o acesso e a persistência dos dados em bancos de dados, enquanto EJB (Enterprise JavaBeans) é utilizado para encapsular a lógica de negócios, fornecendo uma camada de serviços transacionais, seguros e escaláveis. Juntos, eles fornecem uma estrutura robusta para construir aplicativos escaláveis e seguros para a web.

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

O NetBeans oferece recursos como assistentes de código, templates, e ferramentas de gestão que simplificam o uso das tecnologias JPA e EJB. Isso melhora a produtividade ao reduzir a quantidade de código manual

necessário e ao facilitar a configuração e o gerenciamento desses componentes. Essas ferramentas reduzem o trabalho manual de codificação e permitem que o desenvolvedor se concentre em aspectos mais importantes do aplicativo.

d. 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 solicitações e geram respostas em aplicações web. O NetBeans suporta o desenvolvimento de Servlets por meio de assistentes e templates que automatizam a criação de Servlets, mapeamento de URLs e configuração do descritor de implantação, além de integrar-se perfeitamente ao servidor de aplicação para facilitar a implantação e o teste.

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

A comunicação entre Servlets e Session Beans é feita por meio de injeção de dependência, onde os Servlets utilizam anotações para acessar os beans de sessão. Isso permite que os Servlets atuem como uma fachada para a camada de apresentação, enquanto os Session Beans gerenciam as transações de negócios, garantindo a separação de conceitos e a modularidade do código. No ambiente NetBeans, essa comunicação é simplificada pelo gerenciamento automático de dependências e pelo controle integrado do ciclo de vida dos componentes JEE.

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?

O padrão Front Controller é uma estratégia de design que centraliza a gestão de requisições em um único ponto de controle para um aplicativo web. Este ponto, comumente um Servlet, age como um mediador inicial que recebe todas as solicitações dos clientes e determina a ação apropriada com base em parâmetros de URL ou dados de entrada. Uma vez decidido, o Front Controller delega a lógica de negócios aos controladores específicos e seleciona a visualização adequada.

Na arquitetura MVC de um aplicativo Web Java, o Front Controller é implementado como um Servlet que atua como o controlador central, interceptando todas as requisições. Ele coordena as interações com os modelos, que realizam operações de negócios e atualizam as informações necessárias, e com as visões, sejam elas JSPs ou páginas HTML, para apresentar a resposta adequada ao cliente. Este método unificado de tratamento de solicitações simplifica o mapeamento entre requisições e seus manipuladores, promove a reutilização do código e mantém a organização do fluxo dentro do aplicativo.

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

Servlets e JSPs são dois pilares fundamentais do desenvolvimento web com Java, cada um com suas particularidades e casos de uso. Servlets são classes Java que permitem o controle detalhado do processo de requisições HTTP. Eles são particularmente eficazes no processamento de dados e na implementação de lógica de negócios. Por outro lado, JSPs são páginas que mesclam HTML com Java, facilitando a criação de interfaces de usuário dinâmicas. Enquanto os Servlets são mais versáteis e podem ser utilizados em qualquer lógica de negócios, JSPs são mais direcionados para a apresentação e exibição de conteúdo.

Apesar de suas diferenças, Servlets e JSPs compartilham o objetivo comum de dinamizar páginas web e podem ser usados conjuntamente: Servlets geralmente lidam com o processamento de solicitações, enquanto JSPs são usados para apresentar os resultados. Ambos têm acesso a objetos de solicitação HTTP, podendo receber parâmetros e compartilhar

atributos, o que permite a passagem de informações entre a lógica de controle e a camada de apresentação.

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?

Entre as estratégias de navegação web, o redirecionamento simples e o método forward do RequestDispatcher se distinguem na maneira como tratam as solicitações do usuário. O redirecionamento simples, usando o método sendRedirect, causa uma nova solicitação do cliente ao servidor, geralmente resultando em uma mudança visível na URL na barra de endereços do navegador. Em contraste, o método forward mantém a solicitação original e a encaminha internamente para outro recurso no servidor, como outro Servlet ou uma página JSP, mantendo a URL original intacta para o usuário.

Parâmetros em objetos HttpRequest são utilizados para transmitir informações de entrada, como dados de formulários ou seleções de usuário, do cliente para o servidor. Atributos, por sua vez, são usados para armazenar dados que precisam ser acessados por diferentes componentes da aplicação durante o mesmo ciclo de requisição e resposta, facilitando a comunicação interna e o compartilhamento de estados entre Servlets, JSPs e EJBs.

3º Procedimento | Melhorando o Design da Interface

a. Como o framework Bootstrap é utilizado?

O Bootstrap é amplamente reconhecido por sua capacidade de acelerar o desenvolvimento de interfaces web responsivas. Este framework oferece uma vasta gama de estilos CSS prontos para uso e componentes JavaScript, facilitando a implementação de um design coerente e adaptável a variados dispositivos e tamanhos de tela. Comumente, os

desenvolvedores integram o Bootstrap às suas páginas HTML ou JSP adicionando links para os arquivos CSS e JavaScript nos cabeçalhos das páginas. Isso habilita o uso das classes do Bootstrap para estilizar e organizar elementos HTML com eficiência, seguindo um conjunto padronizado de convenções de design.

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

Bootstrap promove uma nítida separação entre a estrutura do conteúdo HTML e o design visual, através do uso de classes CSS que são aplicadas diretamente nos elementos HTML. Esta abordagem desvincula a aparência da página de sua estrutura de marcação, permitindo que alterações de estilo sejam feitas sem necessidade de editar o HTML subjacente. Tal independência potencializa a reutilização de componentes e facilita a manutenção do site, uma vez que ajustes de design podem ser realizados apenas com mudanças nas classes CSS, sem afetar a estrutura do conteúdo.

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

Bootstrap é fundamentalmente projetado para suportar a criação de páginas web responsivas. O framework incorpora um sistema de grid flexível, que se adapta automaticamente aos diferentes tamanhos de tela, bem como uma variedade de componentes e utilitários que mantêm a responsividade independente do dispositivo. Através desses recursos, Bootstrap assegura que a página web se ajuste e apresente de maneira apropriada tanto em dispositivos móveis quanto em desktops, oferecendo uma experiência de usuário consistente e de alta qualidade independentemente da plataforma utilizada.