



Estácio

Estácio - Mundo 3 - Missão Nível 4

FACULDADE ESTÁCIO-> POLO: SALINÓPOLIS/PA.

CURSO: DESENVOLVIMENTO FULL-STACK.

DISCIPLINA: VAMOS INTEGRAR SISTEMAS.

ALUNO: RUAN PAULO DOS SANTOS SOUZA.

SEMESTRE: 2024.3.

Título da Prática

Vamos Integrar Sistemas

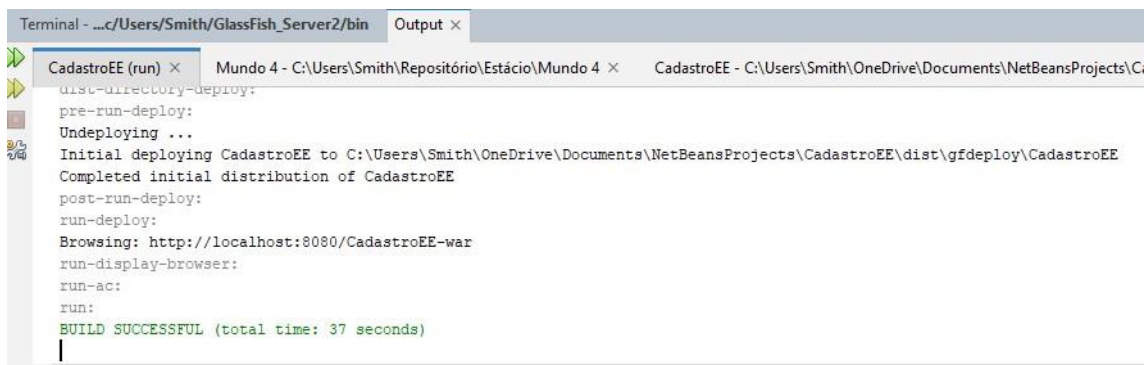
Implementação de sistema cadastral com interface Web, baseado nas tecnologias de Servlets, JPA e JEE.

Objetivos da prática

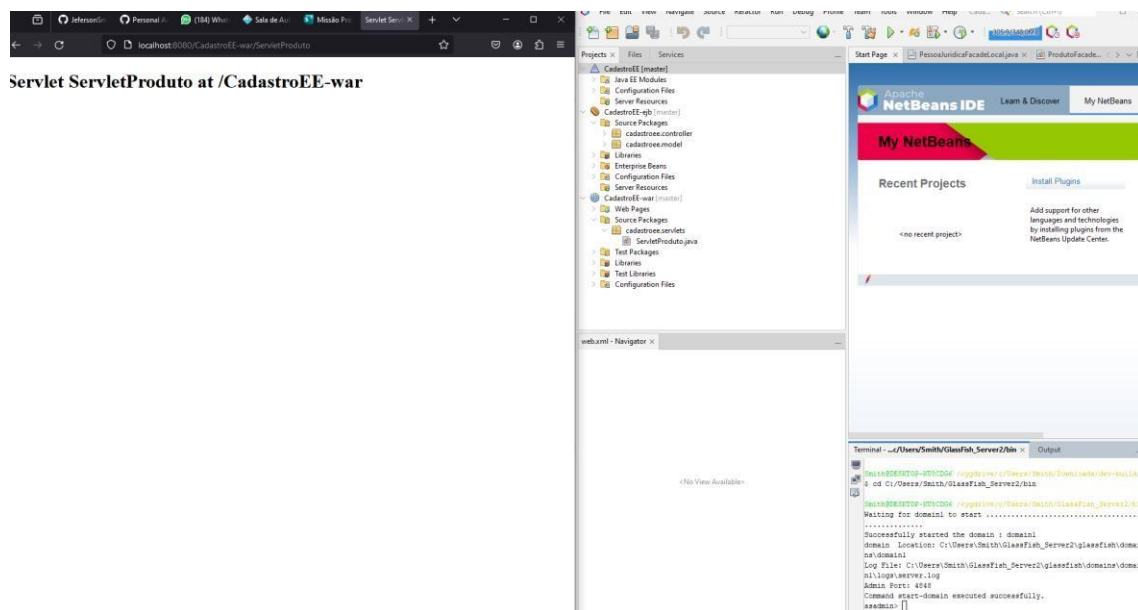
1. Implementar persistência com base em JPA.
2. Implementar regras de negócio na plataforma JEE, através de EJBs.
3. Implementar sistema cadastral Web com base em Servlets e JSPs.
4. Utilizar a biblioteca Bootstrap para melhoria do design.
5. No final do exercício, o aluno terá criado todos os elementos 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

Resultados dos testes:



```
Terminal - ...c/Users/Smith/GlassFish_Server2/bin  Output x
CadastrEE (run) x  Mundo 4 - C:\Users\Smith\Repositório\Estácio\Mundo 4 x  CadastroEE - C:\Users\Smith\OneDrive\Documents\NetBeansProjects\C
dist-ant-repository-deploy:
pre-run-deploy:
Undeploying ...
Initial deploying CadastroEE to C:\Users\Smith\OneDrive\Documents\NetBeansProjects\CadastroEE\dist\gfdeloy\CadastroEE
Completed initial distribution of CadastroEE
post-run-deploy:
run-deploy:
Browsing: http://localhost:8080/CadastroEE-war
run-display-browser:
run-ac:
run:
BUILD SUCCESSFUL (total time: 37 seconds)
```



1. Como é organizado um projeto corporativo no NetBeans?

No NetBeans, um projeto corporativo é tipicamente organizado como um **Enterprise Application Project** (Aplicação Corporativa). Este tipo de projeto pode incluir vários módulos, como módulos de EJB, módulos Web e módulos de aplicação cliente. A estrutura comum inclui:

Enterprise Application Project (EAR): Contém módulos EJB e Web.

Módulo EJB: Contém os Enterprise Java Beans, que incluem lógica de negócios e persistência.

Módulo Web: Contém Servlets, JSPs, e outros componentes da camada de apresentação.

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

JPA (Java Persistence API)

A JPA é utilizada para gerenciar a persistência de dados em aplicativos Java. Ela permite mapear objetos Java para tabelas de banco de dados de forma transparente, utilizando anotações ou XML para definir as entidades e suas relações.

EJB (Enterprise JavaBeans)

Os EJBs são componentes server-side que encapsulam lógica de negócios e são gerenciados pelo container EJB. Existem vários tipos de EJBs, como Session Beans (Stateless e Stateful), que são usados para implementar lógica de negócios, e Message-Driven Beans, que são usados para processar mensagens assíncronas.

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

O NetBeans é uma IDE poderosa que oferece uma série de ferramentas e funcionalidades que aumentam significativamente a produtividade ao trabalhar com tecnologias JPA e EJB. Entre essas ferramentas estão os wizards e assistentes que tornam o processo de criação de entidades JPA, Session Beans e outros componentes EJB mais simples e intuitivo. Esses wizards guiam o desenvolvedor passo a passo, minimizando erros e agilizando o desenvolvimento.

Além disso, o editor de código do NetBeans é uma das suas características mais valiosas. Ele fornece autocompletar para código, o que reduz a quantidade de digitação necessária e ajuda a evitar erros de sintaxe. A verificação de erros em tempo real é outro recurso essencial, permitindo que os desenvolvedores

identifiquem e corrijam problemas à medida que escrevem o código, em vez de descobrir erros apenas durante a compilação ou execução. O editor também é capaz de gerar código boilerplate automaticamente, o que poupa tempo em tarefas repetitivas e permite que os desenvolvedores se concentrem na lógica de negócios mais complexa.

A integração do NetBeans com servidores de aplicação, como GlassFish, WildFly e outros, facilita o deployment e a execução das aplicações. Essa integração permite que os desenvolvedores implantem suas aplicações diretamente a partir do ambiente de desenvolvimento, tornando o processo de teste e debug muito mais eficiente.

4. 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 rodam no servidor e são responsáveis por gerenciar solicitações e respostas HTTP. Eles desempenham um papel crucial na criação de aplicativos web dinâmicos, permitindo a geração de conteúdo dinâmico baseado em solicitações dos usuários. Em essência, um Servlet recebe uma solicitação HTTP do cliente (como um navegador web), processa essa solicitação (por exemplo, consultando um banco de dados ou interagindo com outros serviços), e gera uma resposta HTTP adequada, que é enviada de volta ao cliente.

O NetBeans oferece um suporte abrangente e robusto para o desenvolvimento de Servlets, facilitando várias etapas do processo de criação e implementação desses componentes fundamentais para aplicações web. As principais funcionalidades incluem:

Assistentes de Criação de Servlets: O NetBeans inclui assistentes que simplificam a criação de novas classes Servlet. Esses assistentes guiam o desenvolvedor através do processo de configuração inicial, incluindo a definição do nome do Servlet, o mapeamento de URL e outras configurações essenciais. Isso permite que mesmo desenvolvedores menos experientes possam criar Servlets de maneira rápida e com menos propensão a erros.

Editor de Código: O editor de código do NetBeans é uma ferramenta poderosa que oferece funcionalidades como autocompletar, verificação de sintaxe em tempo real e refatoração. O recurso de autocompletar ajuda a acelerar o desenvolvimento, sugerindo automaticamente métodos e atributos enquanto o desenvolvedor escreve o código. A verificação de sintaxe em tempo real identifica e destaca erros no código à medida que eles são cometidos, permitindo correções imediatas. Além disso, o editor facilita a refatoração do código, permitindo que alterações sejam feitas de maneira segura e eficiente, mantendo a integridade do projeto.

Deployment e Debugging: O NetBeans integra-se diretamente com servidores de aplicação como GlassFish, WildFly, entre outros, o que facilita significativamente

o processo de deployment e debugging das aplicações web. Através dessa integração, os desenvolvedores podem implantar seus Servlets diretamente a partir do ambiente de desenvolvimento, sem necessidade de configurações complexas. Além disso, o NetBeans proporciona ferramentas de debugging que permitem aos desenvolvedores inspecionar o comportamento dos Servlets em tempo real, identificar e corrigir problemas de maneira eficaz.

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

A comunicação entre Servlets e Session Beans em uma aplicação Java EE é um aspecto crucial para a integração da lógica de negócios com a camada de apresentação. Essa comunicação é geralmente realizada através da injeção de dependências, utilizando a anotação `@EJB`.

A anotação `@EJB` permite que um Servlet obtenha uma referência a um Session Bean, que pode ser um Stateless ou Stateful Session Bean. Essa técnica de injeção de dependências simplifica a interação entre o Servlet e o Session Bean, eliminando a necessidade de código boilerplate para realizar lookups no contexto JNDI (Java Naming and Directory Interface). A injeção de dependências garante que o contêiner de aplicação gerencie a criação e o ciclo de vida dos EJBs, facilitando a manutenção e escalabilidade do aplicativo.

2º Procedimento | Interface Cadastral com Servlet e JSPs

1. 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 padrão de design usado para gerenciar todas as requisições para um aplicativo web através de um único ponto de entrada. Ele centraliza a lógica de controle em um único componente, que analisa e processa as requisições, delegando a execução das ações apropriadas. Este padrão é comumente utilizado em arquiteturas Model-View-Controller (MVC) para separar as responsabilidades e promover uma estrutura mais modular e fácil de manter.

No contexto de um aplicativo web Java, o Front Controller é geralmente implementado como um Servlet que atua como o ponto de entrada para todas as requisições do cliente.

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

Servlets e JSPs, ambos componentes da especificação Java EE, possuem propósitos distintos mas complementares. Servlets são classes Java que processam requisições HTTP e são ideais para implementar a lógica de controle e negócio. JSPs, por outro lado, são páginas HTML com código Java embutido, projetadas para facilitar a criação de interfaces dinâmicas, servindo principalmente como camada de apresentação. Enquanto Servlets oferecem controle total sobre o fluxo de requisição e resposta, JSPs simplificam a integração de conteúdo dinâmico dentro de páginas web, sendo automaticamente compiladas em Servlets pelo contêiner web.

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?

Um redirecionamento simples (`response.sendRedirect`) envia uma resposta ao cliente instruindo-o a fazer uma nova requisição para uma URL diferente, resultando em uma nova requisição HTTP e atualização da URL no navegador. Já o método `forward` do `RequestDispatcher` encaminha a requisição internamente no servidor para outro recurso, sem que o cliente saiba, mantendo a URL original.

Parâmetros nos objetos `HttpRequest` são dados enviados pelo cliente, como informações de formulários, enquanto atributos são usados para compartilhar dados entre componentes durante o processamento da requisição.

3º Procedimento | Melhorando o Design da Interface

1. Como o framework Bootstrap é utilizado?

Bootstrap é um framework front-end que facilita o desenvolvimento de sites e aplicações web responsivas e modernas. Ele fornece uma coleção de estilos CSS, componentes JavaScript e layouts predefinidos que ajudam a criar interfaces de usuário atraentes e funcionais. Para utilizá-lo, você inclui os arquivos CSS e JS do Bootstrap no seu projeto e aplica as classes e componentes fornecidos pelo framework ao seu HTML.

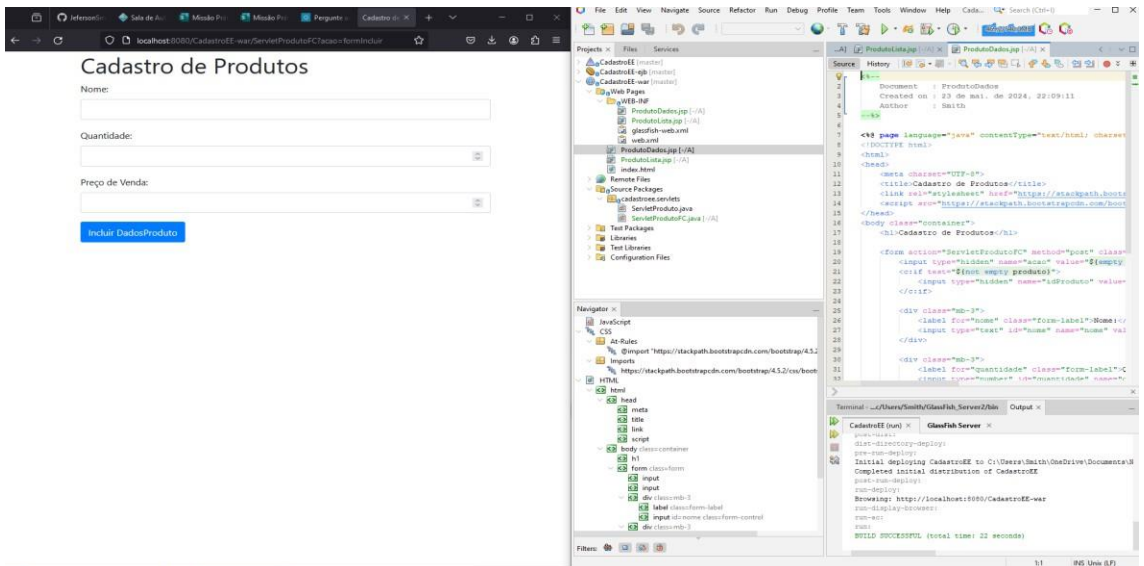
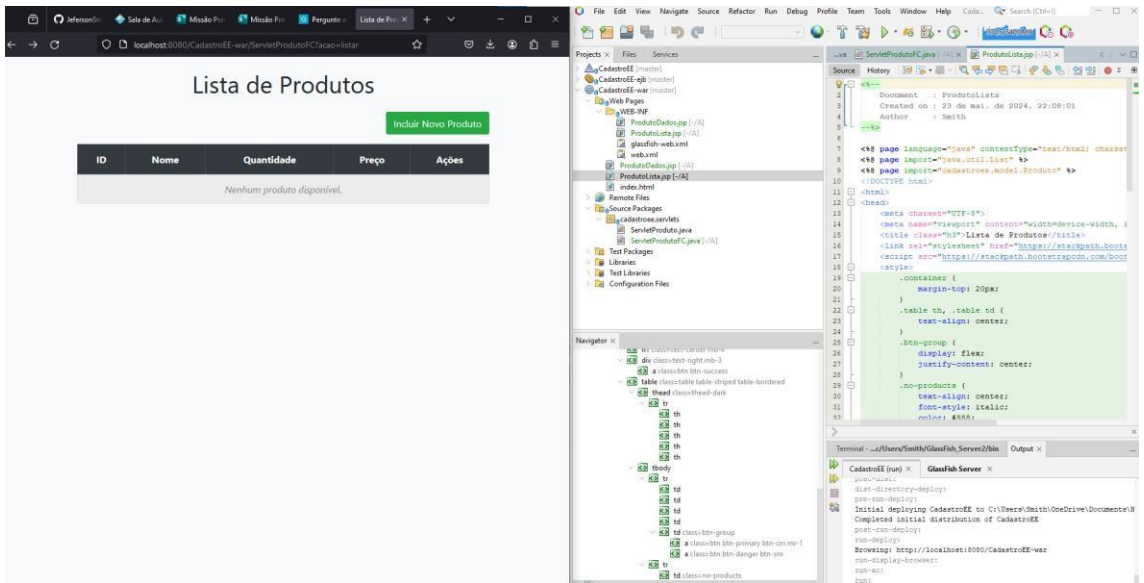
2. Por que o Bootstrap garante a independência estrutural do HTML?

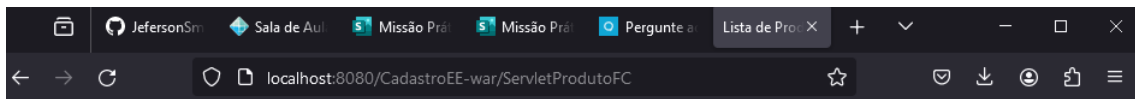
Bootstrap garante a independência estrutural do HTML ao usar classes CSS padronizadas que podem ser aplicadas a qualquer estrutura HTML sem a necessidade de alterar a marcação base. Isso significa que você pode aplicar estilos e comportamentos consistentes a diferentes elementos HTML, mantendo a semântica e a organização do código. As classes do Bootstrap são projetadas para serem reutilizáveis e modulares, permitindo a personalização e a extensão sem modificar o HTML subjacente.

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

Bootstrap é fundamental para a criação de páginas responsivas, pois inclui um sistema de grid flexível e classes utilitárias que se ajustam automaticamente a diferentes tamanhos de tela. Utilizando o sistema de grid, você pode definir layouts que se reorganizam conforme a resolução do dispositivo, garantindo uma experiência de usuário consistente em desktops, tablets e smartphones. Além disso, Bootstrap oferece componentes e utilitários responsivos, como imagens fluidas e classes de visibilidade, que ajudam a adaptar o conteúdo e o design para diferentes contextos de visualização.

Resultados dos testes:





Lista de Produtos

Incluir Novo Produto

ID	Nome	Quantidade	Preço	Ações
1	Macarujã	1	20.0	Alterar Excluir
2	Pera	2	25.0	Alterar Excluir
3	Melancia	5	10.0	Alterar Excluir