 Estácio	<p align="center">UNIVERSIDADE ESTÁCIO DE SÁ</p> <p align="center">POLO PARANGABA – FORTALEZA/CE</p> <p align="center">DESENVOLVIMENTO FULL STACK - 22.3</p> <p align="center">Relatório da Missão Prática Nível 4 Mundo 3</p>
Aluno:	Jefferson Ponte Pessoa
Professor:	Jhonatan Alves
Repositório:	https://github.com/jeffersonkako/M3-Nivel4

Título da Prática: 1º Procedimento | Camadas de Persistência e Controle

Objetivos da Prática:

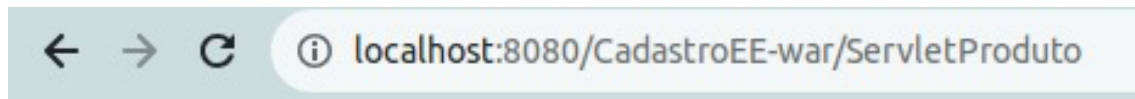
- Implementar persistência com base em JPA.
- Implementar regras de negócio na plataforma JEE, através de EJBS
- Implementar sistema cadastral Web com base em SERVLETS E JSPs
- Utilizar a biblioteca Bootstrap para melhoria do Design
- No final do exercício, o aluno terá criado todos os elementos necessários para exibição e entrada de dados na plataforma java web, tornando-se capacitado para lidar com contextos reais de aplicação

Códigos:

Todos os códigos estão no repositório abaixo:

<https://github.com/jeffersonkako/M3-Nivel4>

Resultados PT1:



Lista de Produtos

Nome do Produto	Preço
uva	R\$ 100,00
banana	R\$ 2,00
tomate	R\$ 3,00
abacaxi	R\$ 2,00

Análise e Conclusão PT1:

1. Como é organizado um projeto corporativo no NetBeans?

A organização de um projeto corporativo no NetBeans geralmente segue uma estrutura modular e hierárquica. No início, você cria um projeto corporativo, que serve como um contêiner para os módulos do projeto. Dentro deste contêiner, podem ser adicionados diferentes tipos de módulos, como projetos EJB para a lógica de negócios, projetos web para a interface do usuário, e projetos de aplicativo cliente para a interação do usuário. Cada módulo é gerenciado separadamente, mas integrado dentro do projeto corporativo para garantir a coesão e a funcionalidade completa do sistema. O ambiente do NetBeans facilita esta organização fornecendo ferramentas e assistentes para configurar, construir e implantar cada parte do projeto de maneira eficiente.

2. Qual o papel das tecnologias JPAe EJB na construção de um aplicativo para plataforma web no ambiente java?

As tecnologias JPA (Java Persistence API) e EJB (Enterprise JavaBeans) desempenham papéis cruciais na construção de aplicativos web em ambiente Java. JPA é responsável pela camada de persistência, facilitando a interação com bancos de dados através de uma abordagem orientada a objetos, permitindo que os desenvolvedores manipulem entidades como objetos Java comuns, sem se preocupar com as complexidades das operações de banco de dados. Por outro lado, EJB é usado para a lógica de negócios, proporcionando um framework robusto para construir componentes de servidor reutilizáveis e escaláveis, gerenciando aspectos como transações, segurança e concorrência. Juntas, essas tecnologias oferecem uma plataforma poderosa para o desenvolvimento de aplicações web empresariais, permitindo uma divisão clara entre lógica de negócios e persistência de dados, além de fornecer um conjunto de serviços que facilitam o desenvolvimento e a manutenção de aplicações complexas.

3. 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 funcionam no lado do servidor e são usados para estender as capacidades de um servidor web. Eles são frequentemente empregados para processar ou armazenar dados que foram submetidos por um formulário HTML, interagir com bancos de dados e outras aplicações de servidor, e produzir conteúdo web dinâmico. O NetBeans, como um ambiente de desenvolvimento integrado (IDE), oferece suporte à criação e gestão de Servlets em projetos web, facilitando o processo com recursos como assistentes de criação, modelos de código, e ferramentas para depurar e testar os Servlets dentro do contexto de um aplicativo web. Isso torna o NetBeans uma escolha popular para desenvolvedores que trabalham com tecnologias Java EE, incluindo Servlets.

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

A comunicação entre Servlets e Session Beans em um pool de Enterprise JavaBeans (EJBs) é realizada através de um mecanismo chamado JNDI (Java Naming and Directory Interface). Os Servlets utilizam o JNDI para localizar e referenciar os Session Beans disponíveis no pool de EJBs. Após a localização, os Servlets podem invocar métodos nos Session Beans como se estivessem chamando métodos em objetos locais. Essa abordagem permite que os Servlets, que geralmente gerenciam solicitações de usuários via HTTP, interajam com lógicas de negócios complexas encapsuladas nos Session Beans, promovendo assim uma separação eficiente entre a camada de apresentação e a lógica de negócios.

Título da Prática: 2º Procedimento | Interface Cadastral com Servlet e JSPs

Dados do Produto

Nome: Quantidade: Preço de Venda:

Análise e Conclusão PT2:

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

O padrão Front Controller em aplicações web Java, especialmente na arquitetura MVC (Model-View-Controller), funciona como um controlador centralizado que gerencia todas as solicitações do cliente. Em vez de ter vários controladores para diferentes tipos de solicitações, o Front Controller atua como um ponto único de entrada, que interpreta as solicitações, delega as tarefas apropriadas para modelos específicos (Model) e, em seguida, seleciona a visualização correta (View) para responder ao cliente. Isso é comumente implementado através de um servlet, que intercepta todas as solicitações, realiza o processamento necessário ou a lógica de negócios e, finalmente, encaminha a resposta para a página JSP ou outra tecnologia de visualização para renderização. Este padrão ajuda na manutenção e gerenciamento centralizado das solicitações, promovendo a reutilização de código e uma estrutura mais organizada.

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

Servlets e JSPs (Java Server Pages) são tecnologias usadas para criar aplicações web em Java, mas têm abordagens distintas. Servlets são classes Java que permitem escrever HTML em código Java, sendo mais adequados para processamento de dados e lógica de negócios. JSPs, por outro lado, são páginas HTML com capacidade de incorporar código Java, sendo mais utilizados para a apresentação e interface com o usuário. Ambos rodam no servidor e podem interagir com bancos de dados e outras tecnologias de servidor. Enquanto Servlets são mais voltados para o controle, JSPs são melhores para a visualização, e frequentemente os dois são usados em conjunto para separar a lógica de negócios da interface de usuário.

3. Como 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 HttpServletRequest?

A principal diferença entre um redirecionamento simples (usando `response.sendRedirect(url)`) e o uso do método `forward` do `RequestDispatcher` está no tratamento da requisição. O redirecionamento simples envia uma resposta ao navegador, indicando que ele deve fazer uma nova requisição para outra URL. Já o `forward` encaminha a requisição atual para outro recurso no servidor sem informar o cliente, mantendo a URL original. Os parâmetros e atributos em objetos `HttpServletRequest` são fundamentais para passar informações entre o cliente e o servidor ou entre diferentes partes do servidor. Parâmetros são tipicamente usados para enviar dados de formulários ou de solicitações de URL, enquanto atributos são usados para manter dados durante a vida útil de uma requisição ou sessão, permitindo a comunicação entre diferentes componentes do servidor.

Título da Prática: 3º Procedimento | Melhorando o Design da Interface

Desenvolvimento Full Stack

Aluno: Jefferson Ponte Pessoa

[Ver Lista de Produtos](#)

Listagem de Produtos

ID	Produto	Quantidade	Preço	Ações
2	azeitona	12	R\$ 1,00	Alterar Excluir

[Cadastrar Produto](#)

Cadastro de Produto

[Voltar](#)

Nome

uva

Quantidade

20

Preço de Venda

100

[Cadastrar](#)

Listagem de Produtos

ID	Produto	Quantidade	Preço	Ações
2	azeitona	12	R\$ 1,00	Alterar Excluir
3	uva	20	R\$ 100,00	Alterar Excluir

[Cadastrar Produto](#)

Alteração de Produto

[Voltar](#)

Nome

uva

Quantidade

20

Preço de Venda

100.0

[Alterar](#)

Listagem de Produtos				
ID	Produto	Quantidade	Preço	Ações
3	uva	20	R\$ 100,00	<div> <div>Alterar</div> <div>Excluir</div> </div>
				<div>Cadastrar Produto</div>

Análise e Conclusão PT3:

1. Como o framework Bootstrap é utilizado?

O framework Bootstrap é amplamente utilizado para desenvolver interfaces de usuário responsivas e mobile-first para websites e aplicações web. Ele oferece um conjunto robusto de ferramentas baseadas em HTML, CSS e JavaScript, que incluem templates pré-desenvolvidos para botões, formulários, navegação e outros elementos de interface, além de um sistema de grid flexível para layout. Isso permite que desenvolvedores construam rapidamente sites esteticamente agradáveis e funcionais, que se adaptam automaticamente a diferentes tamanhos de tela e dispositivos, sem a necessidade de escrever muito código do zero.

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

O Bootstrap assegura a independência estrutural do HTML ao fornecer um conjunto de classes CSS pré-definidas e componentes de interface que podem ser facilmente integrados ao HTML. Isso significa que, em vez de escrever e ajustar estilos CSS personalizados para cada elemento, os desenvolvedores podem simplesmente usar as classes do Bootstrap para alcançar um design consistente e responsivo. Assim, a estrutura do HTML permanece limpa e desacoplada de estilos específicos, facilitando a manutenção e a escalabilidade do código.

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

O Bootstrap é um framework de desenvolvimento web que facilita a criação de páginas responsivas. Ele oferece um sistema de grid flexível, componentes pré-desenvolvidos e classes CSS que se ajustam automaticamente ao tamanho da tela do dispositivo, garantindo que o layout da página seja otimizado para desktops, tablets e smartphones. Isso simplifica o processo de design responsivo, permitindo que desenvolvedores criem sites que proporcionam uma experiência de usuário consistente em diversos dispositivos com menos esforço.