



Vamos integrar sistemas

Israel dos Santos Hamdan D'Araujo - 202303592086

Jequié-BA

Vamos integrar sistemas – 2023.1 – 2024.1

Objetivo da Prática

O objetivo desta prática foi construir uma aplicação web usando GlassFish, JPA, JEE, SQL Server, Servlets e JSP

Link do repositório: <https://github.com/IsraelHamdan/CadastroEE.git>

1º Procedimento | Camadas de Persistência e Controle

a) Como é organizado um projeto corporativo no NetBeans?

Resposta: Dentro do NetBeans um projeto JEE, é dividido em 3 partes:

- Modul EAR (Enterprise Archive): Funciona como um container que empacota os módulos EJB e WAR
- Modulo EJB (Enterprise Java Beans): Funciona como o backend da aplicação, onde se encontra as regras de negócio, consumo do banco de dados ele ainda é subdividido em 3 partes
 - Enterprise Beans: Session Beans (Stateless, Stateful)
Message- Driven
 - Stateless: Não mantem o estado entre as chamadas de método, são reutilizáveis, pois, a mesma instancia pode ser chamada por diversos clientes

- **Stateful:** Mantém o estado entre as chamadas do mesmo cliente, específico para cada instancia.

- **Message-Driven Beans:** Usam processamento assíncrono para lidar com mensagens de um sistema com JMS. Não é invocado diretamente pelo cliente, mas sim pelo container

- **Modulo War:** Modulo responsável pelo front-end, contendo servlets que são classes que processam as requisições HTTP chamando o modulo EJB, as páginas JSP que são HTML com Java embutido e a parte de estilização e configuração do modulo

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

Resposta: O JPA é uma API de persistência nela temos o mapeamento de ORM que converte os objetos Java para tabelas e vice-vers, acessamos os dados sem precisar de SQL e conseguimos executar consultas orientadas a objetos. Já o EJB que encapsula toda a lógica em componentes reutilizáveis, facilita a injeção de dependências

c) O que são servlets e como o Netbeans oferece suporte à construção desse tipo de componentes em um projeto web?

Resposta: Servlets são classes Java que consomem o backend da aplicação e criam logica para um JSP poder trabalhar. O Netbeans oferece um suporte no desenvolvimento pois apresentam um modelo básico de Servlets que se necessário o desenvolvedor pode acrescentar métodos adicionais e moldar conforme a necessidade da sua aplicação!

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

Resposta: Feita por meio de injeção de dependências, os servlets injetam as dependências necessárias para acessar a lógica de negócio

com, chamando os métodos necessários e também ocorre pelo pool que é uma coleção das instancias do Session Beans mantidas pelo container EJB

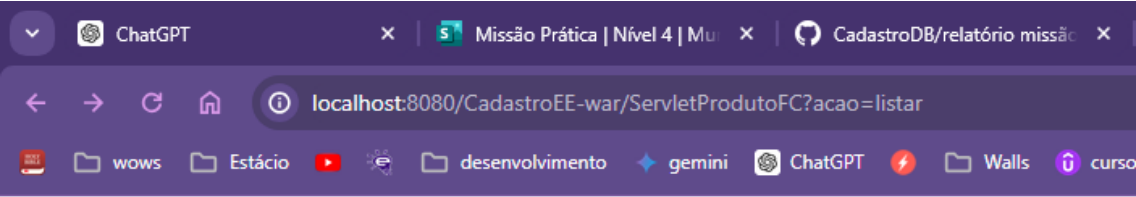
Resultado

Lista de Produtos

- Rum Bacardi Superior 1L
- Tequila Jose Cuervo Especial 750ml
- Água mineral 500ml
- Suco de laranja 1L
- Refrigerante de guaraná 2L
- Cerveja sem álcool lata 350ml
- Energético Red Bull lata 250ml
- Cerveja Original 600 ml
- Cerveja Budwiser 350ml zero alcoool
- Heiniken long neck 350ml zero alcoll
- Sprite limão 2 L
- Amendoim salgado
- Amendoim salgado
- Vinho Perola
- Cerveja Baden-Baden IPA, 350ml

2º Procedimento | Interface Cadastral com Servlet e JSPs

Resultado da lista

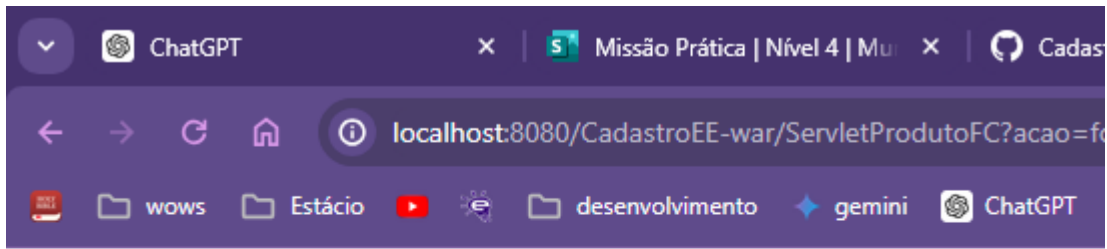


Listagem de Produtos

[Novo Produto](#)

ID	Nome	Quantidade	Preço	Ações
4	Pepsi lata 350ml	90	2.5	Alterar Excluir
5	Guaraná Antarctica lata 350ml	110	2.5	Alterar Excluir
6	Vinho tinto seco 750ml	50	30.0	Alterar Excluir
7	Vodka Smirnoff 1L	30	50.0	Alterar Excluir
8	Whisky Johnnie Walker Red Label 750ml	40	80.0	Alterar Excluir
9	Rum Bacardi Superior 1L	25	40.0	Alterar Excluir
10	Tequila Jose Cuervo Especial 750ml	35	45.0	Alterar Excluir
11	Água mineral 500ml	150	1.5	Alterar Excluir
12	Suco de laranja 1L	70	5.0	Alterar Excluir
13	Refrigerante de guaraná 2L	60	4.0	Alterar Excluir
14	Cerveja sem álcool lata 350ml	75	3.0	Alterar Excluir
15	Energético Red Bull lata 250ml	85	8.0	Alterar Excluir
31	Vinho Perola	100	10.0	Alterar Excluir
34	Cerveja Original 600 ml	100	18.0	Alterar Excluir
35	Cerveja Budwiser 350ml zero alcool	100	4.5	Alterar Excluir
37	Heiniken long neck 350ml zero alcool	100	7.5	Alterar Excluir
38	Heiniken long neck 350ml zero alcool	100	7.5	Alterar Excluir
39	Heiniken long neck 350ml zero alcool	100	7.5	Alterar Excluir
40	Heiniken long neck 350ml zero alcool	100	7.5	Alterar Excluir

Resultado da pagina de cadastro



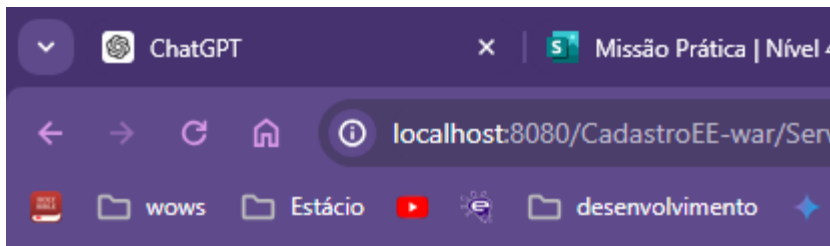
Cadastro de Produto

Nome:

Quantidade:

Preço de Venda:

Resultado da tela alteração



Alteração de Produto

Dados Atuais do Produto

Nome: Pepsi lata 350ml

Quantidade: 90

Preço de Venda: 2.5

Nome:

Quantidade:

Preço de Venda:

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

Resposta: O padrão Front Controller centraliza todo o controle de requisições no aplicativo WEB, ou seja, dentro do modulo War, dentro dos Servlets, eles processam as requisições e fazem as interações com o banco de dados e preparam os dados para a visualização

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

As diferenças são que, o Servlets trabalham com a parte mais lógica, eles que fazem as requisições ao backend da aplicação, por mais que nos dois podemos escrever HTML, o Servlet já focado nas requisições, e também podemos escrever código Java dentro de um JSP, através da sintaxe apropriada, mas nesse caso do JSP fica com uma lógica um pouco mais simples, por exemplo, verificar interações do usuário, exibir um determinado dado

- 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 HttpServletRequest?

- Resposta para pergunta 1: O redirecionamento simples é feito através do método sendRedirect que recebe como parâmetro a página que deve redirecionar o cliente, os atributos da requisição antiga são perdidos, a menos que sejam especificados na url e pode redirecionar para recursos fora do servidor. Já o redirecionamento com o método forward a requisição é feita diretamente no Server-side e serve para encaminhar para outro recurso, como um JSP, dentro do mesmo servidor, nesse caso não é gerado uma nova requisição e os atributos originais da requisição
- Resposta para a pergunta 2: Os parâmetros passados dentro dos métodos do HttpServletRequest servem para buscar dados do cliente (navegador) para o servidor, no caso quando for um parâmetro de Get, e para enviar dados do cliente para o servidor quando for um método do tipo Post, que nesse caso são enviados no corpo da requisição. Já os atributos são objetos que servem para compartilhar informações diferentes

3º Procedimento | Melhorando o Design da Interface

- a) Como o framework Bootstrap é utilizado?

Resposta: Dentro da aplicação proposta o BS é usado dentro do JSP para estilizar a página com suas classes pré-prontas de estilização, sejam tabelas, botões, cards e afins, para utiliza-lo

basta importar o link da folha de estilos e o arquivo JS do BS que ele precisa para funcionar

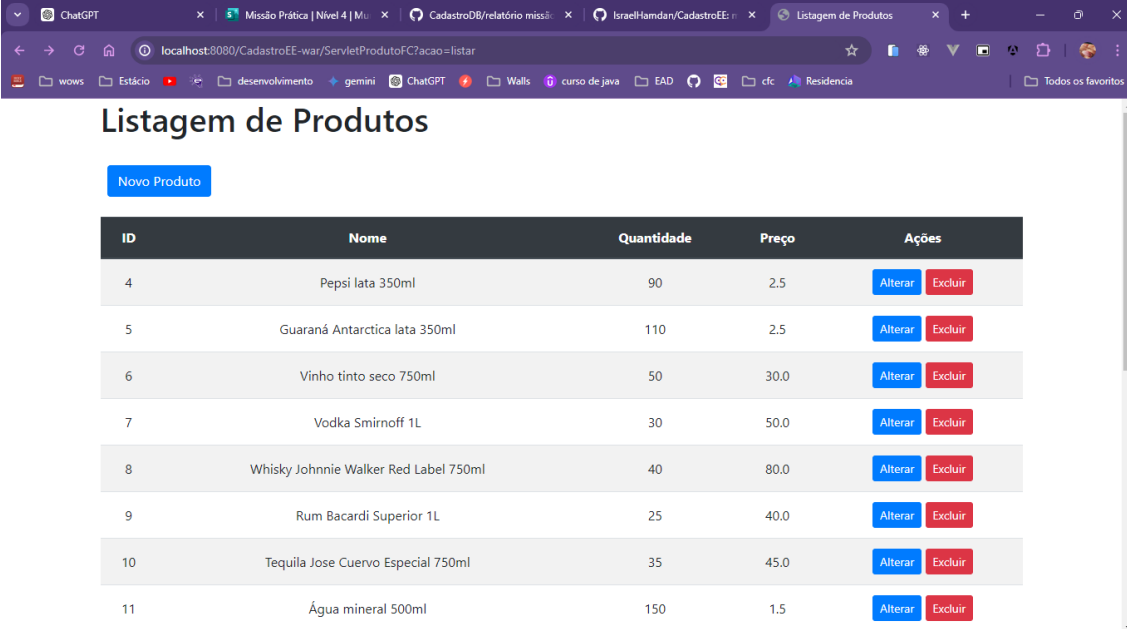
b) Por que o Bootstrap garante a independência estrutural do HTML?

Resposta: O BS garante a independência do HTML fazendo com que o desenvolvedor não precise se preocupar tanto com a estilização da página pois o BS tem classes pré-prontas para navbars, para modais, para botões, posicionamento de itens com CSS

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

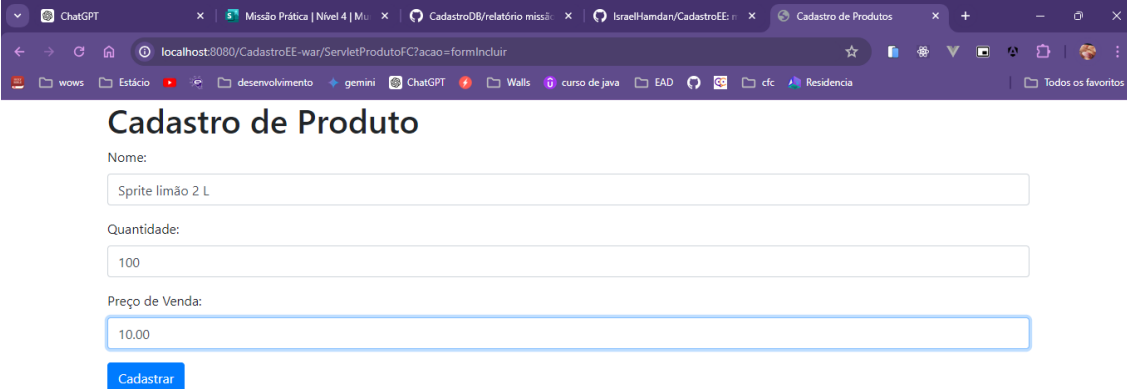
Resposta: o bootstrap garante com que com suas classes grid, com componentes responsivos e outras demais ferramentas, o layout da aplicação se adapte facilmente ao dispositivo no qual a aplicação está sendo acessada

Resultado da listagem de produtos, agora com bootstrap



ID	Nome	Quantidade	Preço	Ações
4	Pepsi lata 350ml	90	2.5	Alterar Excluir
5	Guaraná Antarctica lata 350ml	110	2.5	Alterar Excluir
6	Vinho tinto seco 750ml	50	30.0	Alterar Excluir
7	Vodka Smirnoff 1L	30	50.0	Alterar Excluir
8	Whisky Johnnie Walker Red Label 750ml	40	80.0	Alterar Excluir
9	Rum Bacardi Superior 1L	25	40.0	Alterar Excluir
10	Tequila Jose Cuervo Especial 750ml	35	45.0	Alterar Excluir
11	Água mineral 500ml	150	1.5	Alterar Excluir

Resultado da tela de criação de um novo produto



Nome:

Quantidade:

Preço de Venda:

[Cadastrar](#)

40	Heiniken long neck 350ml zero alcool	100	7.5	Alterar	Excluir
41	Sprite limão 2 L	100	10.0	Alterar	Excluir

Mostrando o produto inserido no banco

Resultado da tela de alteração. Agora com bootstrap

Alteração de Produto

Dados Atuais do Produto

Nome: Guaraná Antarctica lata 350ml

Quantidade: 110

Preço de Venda: 2.5

Nome:

Quantidade:

Preço de Venda:

[Alterar](#)

Dados alterados: o preço que agora é 3.00

Alteração de Produto

Dados Atuais do Produto

Nome: Guaraná Antarctica lata 350ml

Quantidade: 110

Preço de Venda: 2.5

Nome:

Quantidade:

Preço de Venda:

[Alterar](#)

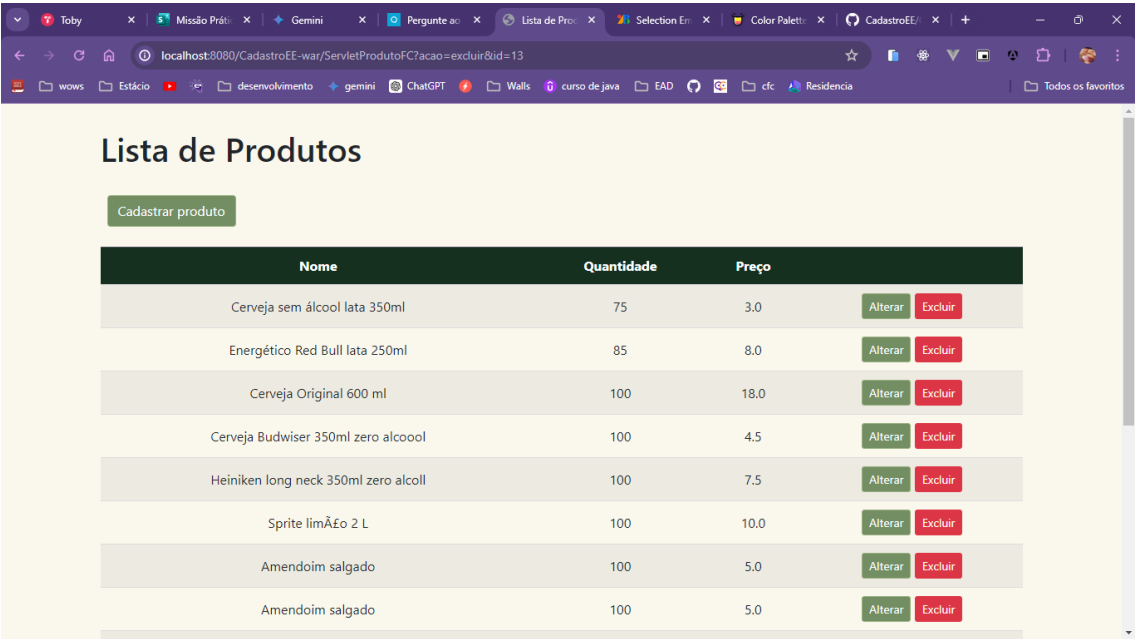
Listagem de Produtos

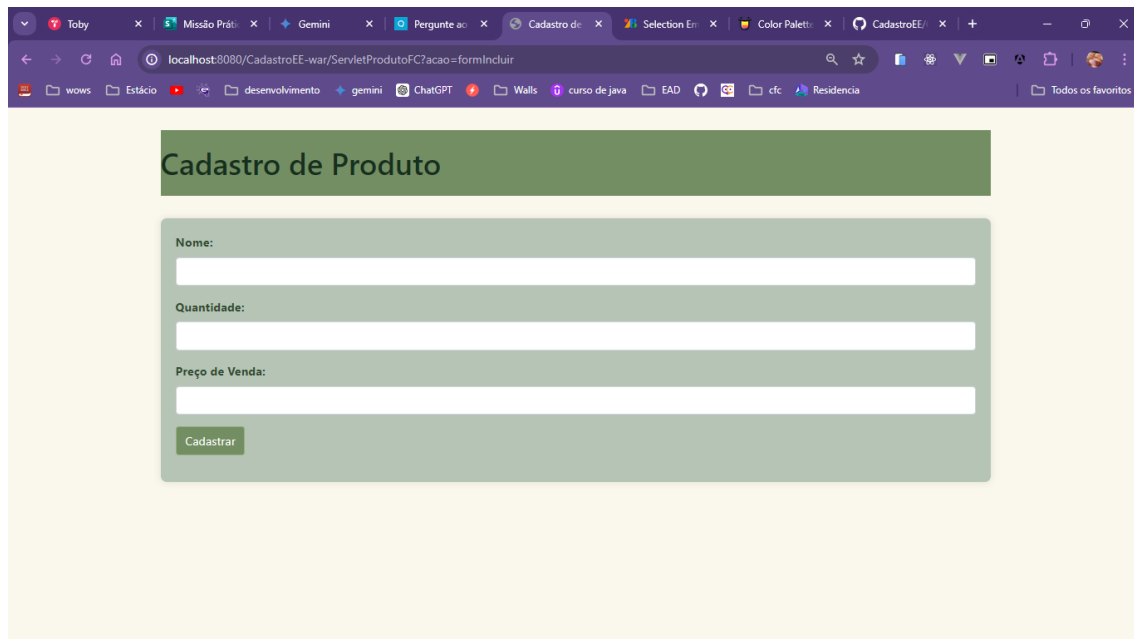
[Novo Produto](#)

ID	Nome	Quantidade	Preço	Ações
4	Pepsi lata 350ml	90	2.5	Alterar Excluir
5	Guaraná Antarctica lata 350ml	110	3.0	Alterar Excluir

Resultado da alteração

Mas para modificar aquela estilização mostrada pelo sway eu quis dar um toque de estilo





Eu sei que esse car com as informações antigas do produto não foi pedido na missão, mas eu quis colocar como uma forma de deixar o projeto mais parecido como seria na vida real, daí eu coloquei para informar ao usuário os dados antigos do produto, para que a alteração seja feita somente onde desejado e os outros dados sejam mantidos como eram caso não for mexer neles!

Conclusão

Desenvolver essa missão prática foi bem difícil porque Sway não foi bem estruturado, penso que as instruções poderiam ser mais claras, senti muita dificuldade para iniciar o projeto e durante o desenvolvimento da aplicação!

Além de que as tecnologias usadas estão muito defasadas, por exemplo o próprio Glassfish que hoje em dia é substituído pelo Apache TomCat, outra tecnologia bastante obsoleta é o Apache NetBeans que atualmente no mercado é usado o JetBrains IntelliJ,. outra coisa que dificultou bastante foi o SQL Server pois é um banco muito difícil de implementar pois existem várias configurações que devem ser feitas para que ele funcione corretamente, usar o mesmo banco da missão 2 sem mudar configuração alguma, ou estrutura alguma para esta missão é um erro gravíssimo, por exemplo, na missão 2, não é solicitado que o id do produto seja auto incrementado pelo IDENTITY ou SEQUENCE, só que quando se vai cadastra um novo produto, o id precisa ser gerado automaticamente, pois o usuário não pode ter acesso às informações

dos ids de todos os produtos do banco de dados e sem que ele tenha acesso ele não vai conseguir imaginar um id que não tenha sido cadastrado anteriormente!, se houvessem essas instruções de modificação do banco de dados, seria muito mais fácil de desenvolver este projeto.

Usar tecnologias velhas assim não faz com que o estudante aprenda como lidar com o mercado atual desenvolvimento web, por exemplo, hoje em dia se usa o Spring boot, um banco de dados mais fácil de implementar como o Postgree SQL e se for trabalhar com desktop o TomCat no lugar do Glassfish e para desenvolver o front end se usa tecnologias de front end como o React ou o Angular coisa que nesta missão não podemos aplicar nessa missão, pois não era o proposito dela, mas deveria, pois, o curso deveria deixar o aluno mais preparado para o mercado de trabalho!

O NetBeans também atrapalhou bastante o desempenho da aplicação pois a versão 21 contava com alguns bugs que dificultaram a recreação e o desenvolvimento do projeto seria muito melhor usar outra ideia para isso, a solução encontrada foi usar uma versão mais antiga do NetBeans que não tinha esses bugs e que se tornou muito mais fácil o desenvolvimento da aplicação, por exemplo uma coisa que deveria haver na versão 21 e que não existe, a versão que foi utilizada para construir esse projeto no Glassfish que foi a 7.0.12 é não existe no NetBeans 21 i na versão 20 tem e foi a única versão que funcionou do GlassFish no projeto. Porém tentar implementar o glassfish dentro do inteligente é mais dificultoso do que dentro do NetBeans, pois o NetBeans possui um suporte melhor a glassfish do que o Inteli j, se fosse para usar o Inteli j era melhor usar o TomCat como servidor de aplicação pois o Inteli j conta com suporte melhor ao TomCat, é mais fácil adicionar Tom Cat no no InteliJ do que o GlassFish

Mas depois que as configurações erradas foram corrigidas, a aplicação foi montada corretamente, desenvolveu o projeto se tornou mais fácil por isso eu eu consegui terminar a missão

Apesar de ser uma missão individual, eu precisei de ajuda dos colegas. pois se não fosse por eles também eu não teria conseguido terminar a missão. pois eles conseguiram fazer a missão e puderam me dar algumas dicas e me ajudar

a botar o meu projeto para funcionar sem eles eu não teria conseguido fazer essa missão nem começar a missão conseguiria então eu tive que consultá-los e pedir a ajuda deles e mostrar os erros que eu estava tendo para que eles pudessem me ajudar. E essa prática de colaboração dentro do mercado de desenvolvimento é fundamental atualmente, os programadores se ajudam muito mutuamente uma está sempre aprendendo com o outro, **mas vale ressaltar que eles não fizeram o meu trabalho! Eles apenas me ajudaram a desenvolver o meu trabalho em pontos que eu tive dificuldade!**