O que falar

Slide 1, Slide 2, Slide 3

ler o ppt

Slide 4

Meu trabalho tem como foco uma aplicação web simples e de fácil usabilidade, possibilitando quaisquer usuários a utilizar. O objetivo principal é demonstrar a utilização da linguagem Java, juntamente com outras tecnologias da atualidade como: HTML5, CSS3, JavaScript e Boostrap.

Para construir a arquitetura da aplicação, optei por não utilizar nenhum framework, afim de aprofundar alguns conceitos existentes.

Slide 5

Ler o ppt

Slide 6

Para que uma aplicação web funcione, ela depende de um servidor web, de solicitações realizadas pelos usuários, do uso de protocolos e métodos (normalmente o HTTP) e da resposta do protocolo.

A aplicação deve **permitir que os usuários consigam fazer uma solicitação e receber algo em resposta.** Ou seja, elas precisam mediar essa interação de forma natural, devolvendo o que o usuário deseja como resultado — por exemplo, se a pessoa pede para abrir uma foto, é preciso que isso seja devolvido a ela o que foi solicitado, neste caso (a foto), e não a abertura de uma página aleatória.

Slide 7, Slide 8

Ler o ppt

Slide 9

O driver está na linha 16, a Url é representada na linha 19, na linha 22 define-se o usuário que irá utilizar o banco de dados e na linha 25 a senha de acesso ao banco.

Slide 10

No padrão mvc o processamento mais pesado a cargo da camada **model,** responsável por acessar o banco de dados. A camada **view** é responsável pela interface com o usuário, ou seja, é a visualização do documento gerado de forma dinâmica. Em um projeto web estes documentos que são renderizados pelo navegador neste caso o html, css e Javascript. No meio destas duas camadas coordenando e controlando o fluxo de dados temos a camada **controller** que basicamente trabalha com requisições e respostas.

Um servidor java, que além de hospedar sites também consegue executar a linguagem java, ou seja, quando o usuário digita uma URL no navegador web e esta URL aponta para um servidor java, ele direciona esta requisição a um container que possui a aplicação, por fim este container procura um arquivo index.html. Na camada **view** além de arquivos com extensão **.html .css .js** a linguagem java permite que utilizemos arquivos com extensão **.jsp**

Dentro da camada model existem duas classes **JavaBeans** e a classe **DAO**.

A classe **JavaBeans** por definição é uma classe **java** que segue um conjunto de especificações e tem como principal objetivo a segurança isolando e encapsulando os dados. Estre conceito está relacionado a permitir que o usuário faça uso dessa classe em um determinado momento e possa salvar o seu estado para o uso posterior, partindo do mesmo ponto.

A classe **DAO** (data access object) é um padrão que encapsula os mecanismos de acesso aos dados escondendo os detalhes da execução, sendo a única classe capaz de estabelecer uma conexão com o banco de dados.

Slide 11

Tela principal do sistema exibindo a lista de contato registrados, e também as possíveis ações que o usuário pode executar:

Adicionar um novo contato, exportar a lista de contatos para pdf, editar um contato e por fim excluir um contato.

Slide 12, Slide 13, Slide 14