Aula 3

Disciplina	LP2A4
Tipo	Conteúdo
Data Aula	@24/08/2022

Tema: Aplicações WEB

Aula reservada para estudo de servidores WEB, tratando assuntos como HTTP, TCP/IP, CGI(Common Gateway Interface).

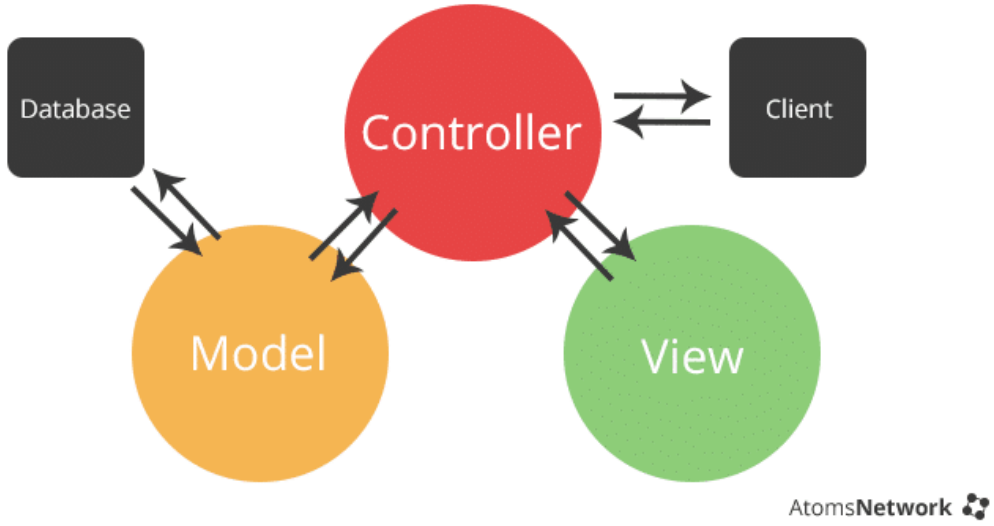
páginas dinâmicas

Aplicações WEB

São aplicações centralizadas com acesso distribuído, possuem alta disponibilidade e bom nível de controle de atualizações.

Padrão MVC

- Model-View-Controller
- padrão de design de software mundialmente reconhecido e utilizado



Servidor Web

aplicação que recebe e trata requisições HTTP.

Container Web

parte do servidor web que trata e gerencia requisições e o ambiente de execução para aplicações.

Server-Side

Representa o conjunto de tecnologias que são interpretadas/processadas diretamente no servidor. Quando um cliente web(navegador) acessa uma página web, uma solicitação é enviada ao servidor através do protocolo HTTP para que o servidor envie a resposta. O Servidor além de rodar os aplicativos, o lado servidor também é um repositório de páginas estáticas, que serão enviados ao cliente quando solicitado. Supondo que haja uma página JSP, esta será processada pelo servidor e encaminhado uma resposta ao cliente (Navegador).

Client-Side

O cliente-side de uma aplicação é o local onde ela é processada, ou seja, no caso da web, executa no navegador do cliente que é o responsável por interagir com o Servidor HTTP.

Hospedagem e execução

App Server: é uma aplicação intermediária que suporta uma determinada tecnologia e fornece um ambiente integrado para execução de serviços e apps.

Java EE (Jakarta EE)

plataforma de desenvolvimento em Java com recurso para a criação de aplicações que executarão em servidores.

Estende o Java Plataform, Standard Edition (Java SE)

Tema: Socket

Socket pode ser considerado o ponto de conexão de duas partes, podendo transmitir dados e mensagens entre máquinas através de Socket usando o padrão TCP. Em Java, é necessário a classe que é o servidor é a que é o cliente. Podemos criar sockets usando o pacote java.net.

ServerSocket

Classe que age como servidor e espera conexão do cliente, em seu construtor é passado a porta que irá escutar.

ServerSocket servidor = new ServerSocket(3322);

accept() → método que escuta uma conexão e aceita ela caso encontrada, ele bloqueia todo o restante até que essa conexão seja realizada, logo, fica em um modo de espera pela conexão. No momento em que a conexão é aceita, ele retorna um Socket.

Socket cliente = servidor.accept();

close() → fecha a conexão;

Socket

Classe que age como o cliente.

getOutputStream() → método que envia dados para o servidor

getInputStream() → método que recebe dados do servidor

Tema: Concorrência e Paralelismo

Contexto

Um contexto de tarefa é o conjunto de dados mínimos que a tarefa usa que devem ser salvos para permitir uma interrupção de task em algum momento e a continuação dessa execução.

Processo

É um programa em execução. Esse processo pode conter várias threads rodando no mesmo contexto, ou seja, compartilhando os recursos de memória. Vale lembrar que processos não compartilhar memória.

Threads

Significa “corrente” ou “fluxo”, é uma sequência de instruções sendo executadas. As threads podem ser executadas em paralelo com outras threads e podem ser interrompidas quando necessário e voltarão exatamente de onde pararam. Diferente do processo que possui um contexto de memória separado, a thread pode compartilhar a memória de seu contexto com outras threads.

Concorrência

É sobre a execução sequencial e disputada de um conjunto de tarefas independentes. No sistema, o responsável por fazer esse gerenciamento é o escalonador de processos, já em alguma linguagem de programação, o responsável por fazer esse gerenciamento é o scheduler interno dessa linguagem.

Paralelismo

É a execução paralela de tarefas, ou seja, depende de forma simultânea da quantidade de núcleos do processador, quanto mais núcleos mais tarefas podem ser executadas.

Programação Síncrona

Na programação síncrona, uma operação precisa ser finalizada para que outra possa ser executada, desse modo, as tarefas são executadas de modo linear.

Programação Assíncrona

Na programação assíncrona, uma tarefa não precisa esperar que outra termine para que ela seja executada.

Multicore

Arquitetura em que existe uma CPU que possui mais de um core, logo, possibilita a execução simultânea de tarefas. Para usufruir dessa arquitetura, é necessário usar de Multitasking, ou seja, a capacidade de existir diversas tarefas e processos simultaneamente.

Multithreading

É como se fosse uma melhoria da Multitasking. Na multithreading, o sistema divide as tarefas em unidades chamadas de threads.