

A camada de aplicação é a camada + alta da pilha de camadas e, portanto, utiliza serviços de todos as camadas anteriores

↳ programadores web não precisam saber sobre enlaços utilizados entre cliente e aplicação!

- Exemplo de aplicações de rede:

- Website;
- Software p/ videoconferências (ex: Zoom);
- etc.

* Arquitetura Cliente-Servidor

• Servidor: hospedeiro que está sempre ligado e tem um endereço IP permanente. Oferece algum serviço. ^{↳ geralmente!}

• Cliente: hospedeiro que pode estar conectado intermitentemente! Se comunica c/ #s servidores a fim de obter serviços/contêúdos. Ademais, pode ter IPs dinâmicos.

* Arquitetura P2P

- Nenhum servidor sempre ligado! Pares requisitam serviços entre si.
↳ estão conectados intermitentemente e trocam de endereço IP
↳ gerenciamento + complexo.
- Auto-escolhibilidade: Novos pares trazem novos dados e novos recursos.

Exemplo: Torrent.

↳ Comunicação entre processos

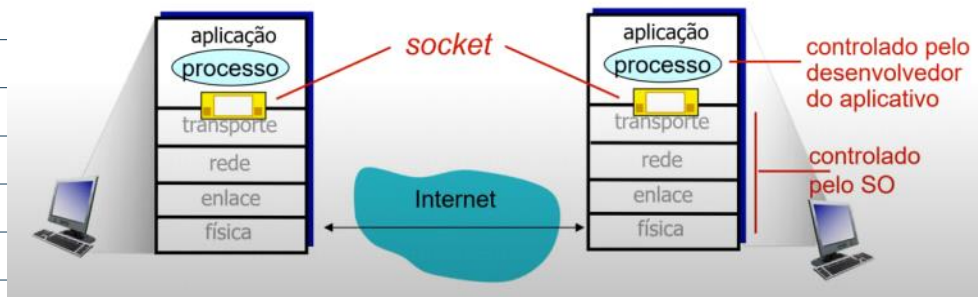
- Processo: Programa listado p/ execução em um hospedeiro
↳ Dentro do hospedeiro, #s processos se comunicam usando comunicação interprocessos, mas quando estamos falando de #s hospedeiros, precisamos usar troca de mensagens.

- Processos clientes: iniciam a comunicação.
- Processos servidores: espera ser contatado.

↳ Aplicativos que usam P2P têm tanto processos clientes, quanto processos servidores.

↳ Processo envia/recebe mensagens de/para seu socket.

* Sockets



• Processo emissor envia a mensagem usando sua porta, e confia na infraestrutura da rede (demais camadas) p/ entregá-lo ao outro lado.

↳ Identificador: endereço IP + porta do processo.
 ↳ Assim podemos nos comunicar com outros hospedeiros.

↳ O que o protocolo da camada de aplicação define?

- Tipos de mensagens trocadas.
 - Sintaxe/estrutura dos mensagens.
 - Semântica (significado) das mensagens.
 - Regras p/ o envio/recebimento das mensagens.
- ↳ a partir dos processos!

- Exemplos de protocolos:

- ↳ Abertos (constam em RFC): HTTP, HTTPS, SMTP, etc.
- ↳ Proprietários: Skype, WhatsApp, etc.