



ARQUITETURAS E CAMADA DE APLICAÇÃO

Camada de Aplicação

Atualmente, as **redes de computadores** estão presentes no cotidiano das pessoas, permitindo a interação e a realização de diversas tarefas.



Em relação às redes de comunicação, você já ouviu falar no trabalho da **camada de aplicação**?

Vamos analisar os conceitos estudados na prática? Veja o caso a seguir.

Exemplo

Quando realizamos uma compra com cartão de crédito ou débito em um estabelecimento comercial, é fundamental a existência de uma rede de comunicação, já que ela será o alicerce para execução da operação. Ao inseri-lo na máquina de cartão, precisamos colocar uma senha para confirmar a operação. Tal dado é inserido no sistema por meio de um software executado nessa máquina.

Nesse caso, o software é **executado** na **camada de aplicação**!

O software de aplicação, também conhecido como **software aplicativo**, é nossa interface com o sistema (e, por consequência, com toda a rede de comunicação que suporta essa operação). Portanto, sempre que houver um serviço na rede, virá à mente a interface com ele.

Outros exemplos de software de aplicação:



Navegador web



Cliente de e-mail



Jogos executados em rede

Ressaltamos que a camada de aplicação é aquela de mais alto nível do modelo OSI, fazendo a interface com os usuários do sistema e realizando as tarefas que eles desejam.

Arquiteturas de Aplicações

Façamos a seguinte suposição: nosso objetivo é desenvolver uma aplicação a ser executada em rede. Para criá-la, deve-se utilizar uma **linguagem de programação** que possua **comandos e/ou funções** para a **comunicação em rede**. Na maioria das linguagens, esses comandos e/ou funções estão em bibliotecas nativas da linguagem ou criadas por terceiros.

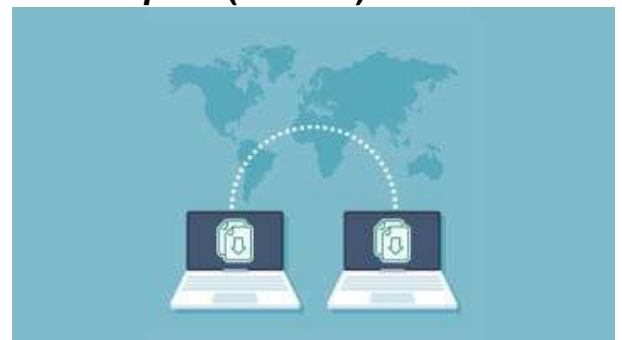


Mas não basta conhecer uma linguagem de programação e suas bibliotecas. Antes disso, é preciso definir **qual arquitetura** terá **sua aplicação**. Entre as mais conhecidas, destacam-se as seguintes:

Cliente-servidor



Peer-to-peer (ou P2P)

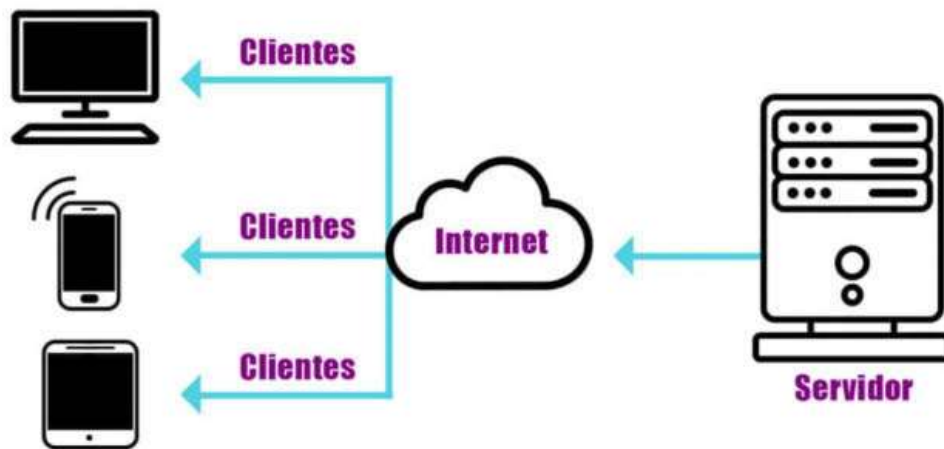


Arquitetura Cliente Servidor

Cliente-servidor

Nesta arquitetura, há pelo menos **duas entidades**: um **cliente** e um **servidor**. O servidor executa operações continuamente aguardando por requisições do(s)

cliente(s). O mais comum, é que hajam diversos clientes e um único servidor. A figura abaixo ilustra a situação.



Quando um dos **clientes** precisa que o trabalho seja realizado pelo servidor, ele monta uma mensagem, especificando o que deve ser realizado. A mensagem normalmente contém dados que devem ser processados pelo servidor.

Quando a mensagem está montada, é enviada ao **servidor** por intermédio de algum sistema de comunicação (internet). Este recebe a mensagem, processa seu conteúdo e envia a resposta ao cliente.

O servidor comporta-se passivamente, normalmente limita-se a aguardar solicitações, e quando estas chegam, processa as mensagens com os dados necessários e enviar o resultado do processamento de volta a seus clientes.

Servidor

Quando chega uma solicitação, o servidor pode:

- Atender imediatamente caso esteja ocioso;
- Gerar um processo-filho para o atendimento da solicitação;
- Enfileirar a solicitação para ser atendida mais tarde;
- Criar uma thread para esse atendimento.

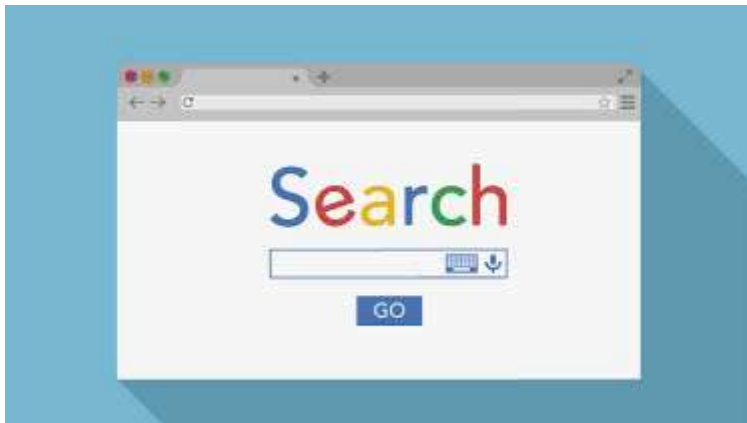
Independentemente do momento em que uma solicitação é processada, o servidor, no final, envia ao cliente uma mensagem contendo o **resultado do**

processamento.

Quem utiliza esse tipo de arquitetura é a **aplicação web**.

Vamos analisar esse processo no exemplo a seguir.

1. Você deseja fazer uma receita especial, descobrindo, em um site, aquele prato que gostaria de preparar. Ao clicar em um link, ela aparecerá. Para isso acontecer, o servidor web (software servidor do site de receitas) fica aguardando as conexões dos clientes.



2. Quando você clica no link da receita, seu browser envia uma mensagem ao servidor indicando qual delas você quer.



3. Ele faz então o processamento solicitado e devolve ao browser o resultado disso (sua receita).



É muito importante compreender, de maneira prática, o funcionamento do processamento de resultados.

Atenção!

O que **determina** se uma entidade é **cliente** ou **servidor** é a **função desempenhada** pelo software, e não o tipo de equipamento.

É fundamental saber que servidores desempenham uma função muito importante; por isso, há equipamentos apropriados para eles, com MTBF alto e recursos redundantes.

O tipo de software instalado nesse equipamento é o **responsável** por **determinar** se ele é **cliente** ou **servidor**.

Além disso, **um processo** pode **atuar simultaneamente** como **cliente** e **servidor**.



Voltemos ao exemplo da aplicação web:

Quando seu browser solicita a receita ao servidor web, aquele está atuando como cliente e este, como servidor.

Mas esse **processo nem sempre é simples**; afinal, a aplicação que executa no servidor web e realiza o processamento solicitado pode precisar de uma informação armazenada em um banco de dados externo.

Para obtê-la, esse servidor deve enviar uma mensagem ao servidor de banco de dados solicitando aqueles de que necessita para continuar. Nesse momento, ele atua como um **cliente** do **servidor de banco de dados**.

Arquitetura *Peer-to-peer*

Peer-to-peer

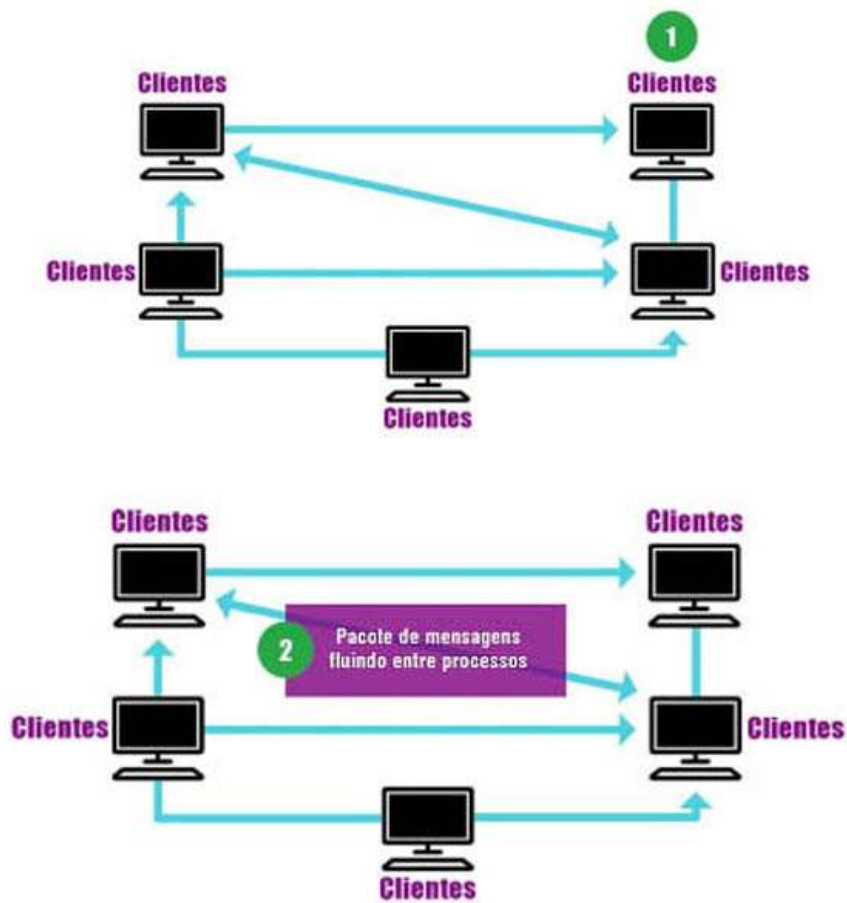
Enquanto existe uma distinção bem clara entre os processos que trocam informações na arquitetura cliente-servidor, na peer-to-peer – também conhecida como arquitetura P2P –, todos os processos envolvidos desempenham funções similares.

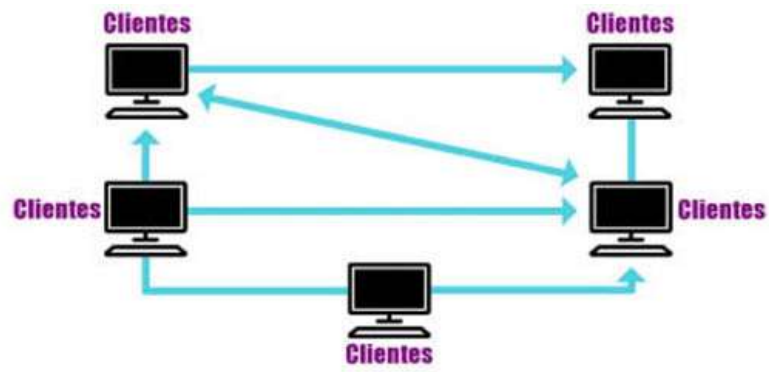
Em geral, nesses sistemas, os processos não são uma propriedade de corporações. Quase todos os participantes (senão todos) são provenientes de **usuários comuns** executando seus programas em desktops e notebooks.

Tanto o processamento quanto o armazenamento das informações são distribuídos entre os hospedeiros. Isso lhes confere maior escalabilidade em comparação à arquitetura cliente-servidor.

Modelo de arquitetura *peer-to-peer*.

Na ordem, 1, 2 e 3, o fluxo de informações e como elas se organizam.





3

Esta arquitetura foi bastante impulsionada pelo surgimento dos computadores pessoais de alto desempenho e das redes banda larga