

# Revisao

---

## 1. Introdução a Sistemas Distribuídos

- **Definição e Importância:** Sistemas distribuídos são formados por vários computadores que cooperam para realizar tarefas como se fossem uma única entidade. Exemplos incluem bolsas de valores, onde a propagação de eventos precisa ser quase em tempo real, e plataformas de streaming, que requerem uma alta disponibilidade para os usuários.
- **Casos de Uso:** Um exemplo destacado foi o uso de sistemas distribuídos para automação em mercados financeiros, onde o software detecta e propaga rapidamente mudanças nos preços das ações.

## 2. Arquitetura dos Sistemas Distribuídos

- **Cliente-Servidor:** Nessa arquitetura, o servidor oferece um serviço enquanto os clientes fazem requisições. Esse modelo é comum, por exemplo, em aplicações web, onde o servidor espera por solicitações e os clientes as iniciam.
- **P2P (Peer-to-Peer):** Nesse sistema, cada nó atua tanto como cliente quanto como servidor. Usado em torrents, onde todos os participantes compartilham e obtêm arquivos simultaneamente, ajudando a distribuir a carga de dados.

## 3. Desafios e Tendências

- **Mobilidade e Onipresença:** Com a mobilidade, os sistemas distribuídos devem se adaptar para que o usuário tenha acesso contínuo ao serviço, mesmo que mude de rede (Wi-Fi para 4G, por exemplo).
- **Heterogeneidade:** Dispositivos e sistemas operacionais diversos precisam se comunicar. Para isso, usa-se o middleware, que atua como um intermediário e facilita a comunicação entre sistemas diferentes.
- **Escalabilidade:** Capacidade de aumentar o número de usuários ou cargas sem comprometer a performance. É fundamental em sistemas distribuídos de larga escala, como serviços de streaming.
- **Confiabilidade e Tolerância a Falhas:** Em um sistema distribuído, uma falha de um componente não deve comprometer o sistema inteiro, pois as falhas são tratadas de forma independente.
- **Serviços em Nuvem:** A computação em nuvem permite que usuários aluguem recursos (processamento e armazenamento) em vez de comprá-los, sendo uma tendência crescente devido à economia e flexibilidade proporcionadas.

## 4. Middleware

O middleware é uma camada de software entre o sistema operacional e as aplicações que esconde detalhes técnicos (como a rede e o tipo de hardware) e facilita a comunicação entre componentes distribuídos. Ele cuida da **heterogeneidade** do sistema, permitindo que diferentes dispositivos e redes trabalhem juntos.

## 5. Exemplos de Sistemas Distribuídos

- **Sistemas de Multimídia:** Como Netflix e YouTube, que distribuem conteúdo de vídeo para milhares de usuários. Esses sistemas exigem caching e distribuição geográfica para melhorar a eficiência.
- **Jogos Online Massivos:** Jogos com muitos jogadores simultâneos, que demandam uma infraestrutura escalável para manter a experiência de jogo em tempo real.

## 6. Modelos de Comunicação e Conexões

- **TCP e UDP:** Dois protocolos usados em comunicação entre cliente e servidor:
  - **TCP** (Transmission Control Protocol): Protocolo orientado à conexão que garante a entrega de dados com confiabilidade. Muito usado em aplicações onde a ordem dos dados e a confirmação da entrega são importantes.
  - **UDP** (User Datagram Protocol): Protocolo não orientado à conexão, usado onde a velocidade é mais importante que a confiabilidade, como em transmissões de vídeo ao vivo.

## Questões para Estudo

#estudo/RevisaoSD

1. **O que é um sistema distribuído? Dê um exemplo e explique a importância de sua utilização.**  
?

Um sistema distribuído é uma rede de computadores que trabalham juntos para realizar tarefas complexas, dando ao usuário a impressão de que é um único sistema. Cada computador (ou “nó”) executa parte das operações, permitindo que grandes volumes de dados e processos sejam divididos. Por exemplo, o Google é um sistema distribuído: ele armazena dados em servidores ao redor do mundo e consegue fornecer respostas rápidas a milhões de usuários simultaneamente. A importância dos sistemas distribuídos está em sua capacidade de atender uma grande quantidade de usuários e realizar operações em larga escala, garantindo eficiência, escalabilidade e, muitas vezes, maior segurança.

2. **Qual a diferença entre os modelos Cliente-Servidor e Peer-to-Peer?**  
?

No modelo Cliente-Servidor, o servidor é responsável por fornecer serviços (como armazenar dados ou processar informações) enquanto o cliente faz as solicitações. Por exemplo, ao acessar um site, o computador do usuário (cliente) faz uma solicitação ao servidor, que responde enviando a página solicitada. Já no modelo Peer-to-Peer (P2P), todos os computadores (ou “pares”) podem agir tanto como clientes quanto como servidores. Um exemplo é o sistema de torrents, onde cada usuário pode tanto enviar quanto receber partes de arquivos de outros. A principal diferença entre os dois modelos é que, no Cliente-Servidor, há uma distinção clara de papéis (servidor fornece e cliente consome), enquanto no P2P todos os participantes colaboram ativamente.

3. **Explique o papel do middleware em sistemas distribuídos.**  
?

O middleware é como um intermediário que facilita a comunicação e a integração entre diferentes sistemas e dispositivos em um sistema distribuído. Ele esconde detalhes técnicos, como o tipo de rede ou sistema operacional, permitindo que aplicativos em plataformas distintas funcionem juntos. Por exemplo, o middleware permite que um smartphone com sistema Android e um computador com sistema Windows compartilhem informações sem problemas. Ele faz isso “traduzindo” as informações, tornando o sistema mais eficiente e fácil de programar. Sem o

middleware, os desenvolvedores precisariam lidar com muitos detalhes complexos de hardware e rede.

**4. Por que a escalabilidade é importante em sistemas distribuídos?**

?

A escalabilidade é essencial em sistemas distribuídos porque permite que o sistema aumente sua capacidade de atender mais usuários ou processar mais dados sem perder desempenho. Em uma rede social como o Facebook, por exemplo, milhões de usuários podem estar online ao mesmo tempo; se o sistema não fosse escalável, ele ficaria lento ou até pararia de funcionar. Um sistema escalável consegue "crescer" conforme a demanda aumenta, o que é crucial para grandes aplicações e serviços que precisam manter uma boa experiência para os usuários, mesmo com um número muito alto de acessos.

**5. Quais são as vantagens de utilizar serviços em nuvem em sistemas distribuídos?**

?

Os serviços em nuvem permitem que as empresas "aluguem" recursos, como espaço de armazenamento ou capacidade de processamento, ao invés de comprar e manter servidores físicos. Isso reduz custos e aumenta a flexibilidade, já que é possível ajustar a quantidade de recursos conforme a necessidade. Outra vantagem é a acessibilidade: os dados e aplicativos ficam disponíveis na internet, permitindo que os usuários acessem de qualquer lugar. Por exemplo, uma empresa que usa o Google Drive pode armazenar e compartilhar documentos online, o que facilita o trabalho em equipe e reduz a necessidade de infraestrutura própria.

**6. Qual a principal diferença entre os protocolos de comunicação TCP e UDP?**

?

A principal diferença entre TCP e UDP está na forma como eles tratam a entrega de dados. O TCP (Transmission Control Protocol) é um protocolo confiável que garante que os dados cheguem ao destino sem erros e na ordem correta. É como enviar uma carta registrada, em que cada etapa é confirmada. Por isso, é usado em transmissões onde a ordem e a confiabilidade dos dados são importantes, como no carregamento de páginas da web. Já o UDP (User Datagram Protocol) é mais rápido, mas menos confiável, pois não garante que os dados cheguem nem que cheguem na ordem correta. Ele é utilizado em situações onde a velocidade é mais importante que a confiabilidade, como em transmissões ao vivo e jogos online, onde pequenas perdas de dados não comprometem a experiência.