



# Capítulo 1: Introdução a Sistemas Distribuídos

## Introdução a Sistemas Distribuídos

### Definição

- Conjuntos de computadores independentes que trabalhar juntos como se fossem um só, para aparecer para os usuários como um único sistema, mas estão fisicamente distribuídos em diferentes locais.
- Sem os sistemas distribuídos, seriam inviáveis ou necessitariam de hardware muito caro e complexo
- **Exemplos:**
  1. **Internet com seus protocolos:** Conecta várias redes menores, formando uma vasta rede global, com diversas aplicações distribuídas (Web, E-mail, Jogos Online, Mensagens Instantâneas) e sua estrutura de rede (borda de rede e núcleo da rede)
  2. **BitTorrent:** Sistemas como o BitTorrent permitem que cada nó na rede atue como Cliente e servidor
  3. **Blockchain:** computadores que validam transações.
  4. **Computação em Nuvem:** Onde recursos de computação e armazenamento são distribuídos entre muitos servidores e são acessíveis pela internet.
  5. **Sistemas de Arquivos Distribuídos:** Onde dados são armazenados em vários computadores, facilitando o acesso e a redundância.
  6. **Sistemas de Banco de Dados Distribuídos:** armazenados em múltiplos servidores para melhorar a disponibilidade e

escalabilidade.

**OBS:** Um navegador web, por si só, **NÃO** é considerado um sistema distribuído. Ele é uma aplicação cliente que interage com sistemas distribuídos (os servidores web)

## VANTAGEM / MOTIVAÇÃO DE SISTEMAS DISTRIBUIDOS

- **Objetivo:** Facilitar o compartilhamento, comunicação e coordenação de recursos e serviços, melhorar a eficiência e aumentar a escalabilidade

### 1. Compartilhamento de Recursos Remotos:

- **Hardware:** Recursos e dispositivos que não estão fisicamente presentes e Acesso a banco de dados distribuídos

### 2. Concorrência

- **Paralelismo:** Podendo executar tarefas em paralelo, utilizando múltiplos processadores ou máquinas para dividir o trabalho → Melhora a eficiência

### 3. Melhor Desempenho

- **Cache:** Uso de cache diminui o tempo de acesso a dados frequentemente utilizados → Melhora desempenho do sistema

### 4. Maior Confiabilidade e Redundância

- **Redundância:** Manter várias cópias de dados ou serviços em diferentes locais → Se uma parte falhar, outras partes podem funcionar
- **Falhas Parciais:** A falha de um componente não necessariamente implica na falha de todo o sistema.

### 5. Escalabilidade:

- **Expansão Horizontal:** Adicionar mais máquinas ou nós para aumentar a capacidade do sistema.

## **6. Flexibilidade e Adaptabilidade**

- **Diversidade de Recursos:** Pode integrar diferentes tipos de hardware e software, adaptando-se a diversas necessidades e tecnologias.

## **7. Transparência:**

- O sistema distribuído **oculta a complexidade da interação** entre os computadores, fazendo com que **pareça que o sistema é único e coeso**.

- Tipos:

1. Transparência de acesso
2. Transparência de localização e concorrência
3. Transparência de Desempenho
4. Transparência de falhas
5. Transparência de Replicação

- **Autonomia dos Computadores:**

- Cada computador (ou **nó**) **é independente, mas trabalha em conjunto** com outros para fornecer serviços ou executar tarefas.

**OBS:** sistemas distribuídos são projetados para serem **ESCALÁVEIS**, uma das vantagens de SDs

## **Desvantagens de Sistemas Distribuídos**

- **Complexidade:**

- Custo maior de desenvolvimento

- Necessidade de coordenar e sincronizar múltiplos componentes e nós.
- **Debugging e Testes:** Encontrar e corrigir bugs pode ser mais difícil, pois problemas podem ocorrer devido à comunicação entre nós, falhas de rede, ou inconsistências nos dados.
- **Qualidade e Desempenho (QoS):**
  - **Comunicação de Rede:** comunicação entre diferentes nós pode introduzir latência, afetando o desempenho geral do sistema. A latência de rede pode variar e afetar o tempo de resposta das operações.
- **Consistência e Sincronização:**
  - Dados Replicados, Latência de Rede, Sincronização de Relógios e Overhead
  - **Ausência de Relógio Global:** Cada nó tem seu próprio relógio local, e esses relógios podem não estar sincronizados. Não há um único relógio físico que todos os nós do sistema distribuído
  -
- **Segurança:**
  - **Ataques e Vulnerabilidades:** Sistemas distribuídos podem ser mais vulneráveis a ataques, como ataques de negação de serviço (DoS)
  - **Controle de Acesso:** Gerenciar a segurança e o controle de acesso em um ambiente distribuído é mais complexo do que em sistemas centralizados.
- **Gerenciamento de Falhas:**
  - **Recuperação de Falhas e Tolerância a Falhas:** Implementar mecanismos eficazes de tolerância a falhas e recuperação pode ser desafiador e consumir recursos.

***OBS:*** distribuição implica em custos, complexidade (sincronização, coordenação) e riscos (falhas, segurança)

## Tendências em Sistemas Distribuídos

- Tendências atuais em sistemas distribuídos:
- **Pervasivas:** "Todo lugar, o tempo todo" – uma rede que está em todo lugar, conectando tudo.
- **Ubíqua e Mobilidade:** "Computação em qualquer lugar" – em poder acessar tudo de qualquer lugar, como seu smartphone sempre disponível.
- **Multimídia:** "Vídeos e Áudio em alta demanda" – popularidade do streaming e como os sistemas devem se adaptar.
- **Serviço Público:** "Tecnologia acessível para todos" – sistemas distribuídos como um serviço que todos podem usar e acessar facilmente.

## Middleware

- Camada de software que atua como um intermediário entre sistemas e aplicações, fornecendo funcionalidades como comunicação, gerenciamento de transações e serviços de autenticação
  - "Tradutor ou facilitador" que ajuda diferentes programas e sistemas a se comunicarem e trabalharem juntos, mesmo que eles estejam rodando em máquinas/arquiteturas diferentes.
  - Em vez de os aplicativos se comunicarem diretamente com outros aplicativos ou sistemas, eles interagem com o middleware.

*Ex: O aplicativo de processamento de pedidos envia uma solicitação ao middleware, que então comunica com o aplicativo de banco de dados e retorna a resposta ao aplicativo de processamento de pedidos.*

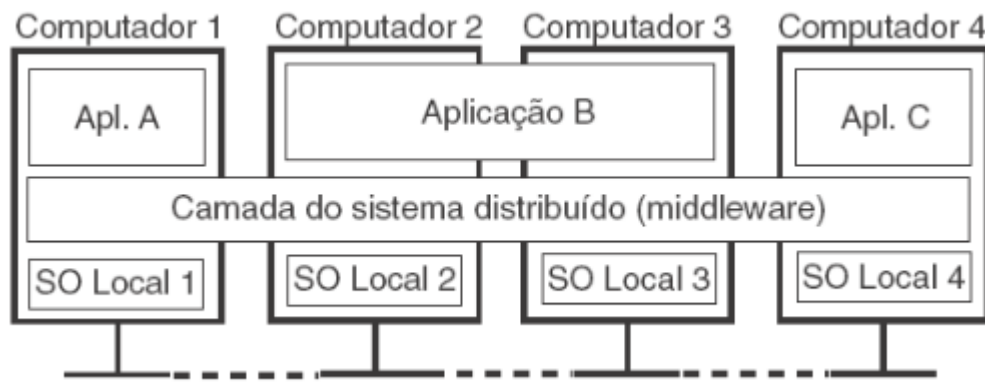


Figura 1: Middleware (Intermediário) que se estende por várias máquinas e oferece uma interface única para todos os aplicativos.

### Perguntas

"Como sincronizar os componentes sem a existência de um relógio global?"

"Como gerenciar a concorrência no acesso aos recursos compartilhados?"

"Como gerenciar falhas em componentes independentes?"