

# Sistemas Distribuídos



Introdução

Edeyson Andrade Gomes

[www.edeyson.com.br](http://www.edeyson.com.br)

# SUMÁRIO

---

- ▶ Definições
- ▶ Características
- ▶ Desafios
- ▶ Vantagens
- ▶ Desvantagens



# Definições

# DEFINIÇÕES

---

- ▶ Um sistema distribuído é uma coleção de computadores independentes que aparecem para seus usuários como um único computador. [Tanenbaum]
- ▶ Um sistema distribuído é aquele em que eu não posso fazer meu trabalho, pois uma máquina que eu nem conheço, não sei onde está e nunca ouvi falar encontra-se fora do ar. [Lamport]

# DEFINIÇÕES

---

- ▶ Um sistema distribuído é uma coleção de computadores que não compartilham memória nem relógio. [Silberschatz]
- ▶ Um sistema distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas passando mensagens. [Coulouris]



# Características

# CARACTERÍSTICAS

---

## ▶ Hardware

- ▶ Os computadores são autônomos;
- ▶ Interconexão através de rede de comunicação;
- ▶ Não compartilhamento de relógio e memória.

## ▶ Software

- ▶ Visão de um sistema único;
- ▶ Troca de mensagens;
- ▶ Abstração e encapsulamento de detalhes.

# CARACTERÍSTICAS

---

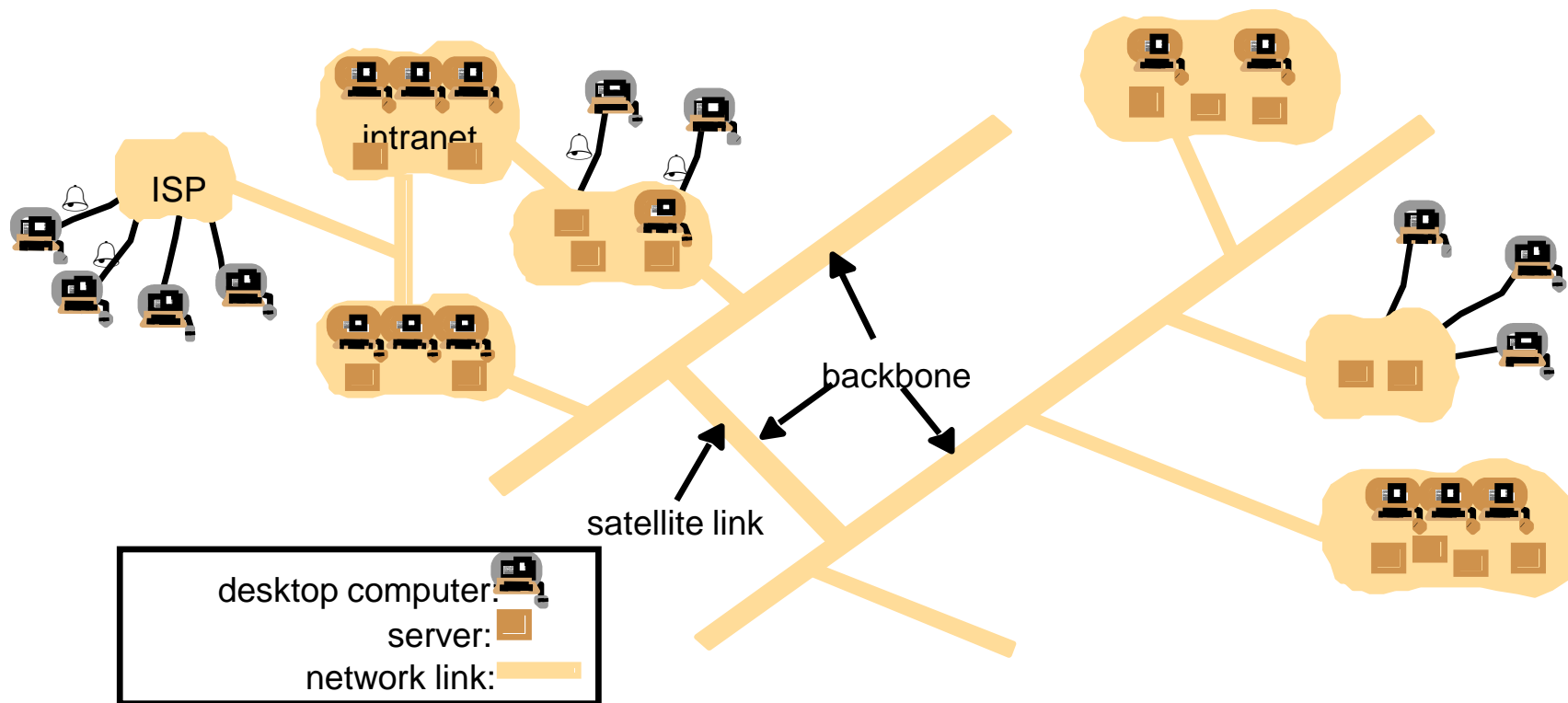
- ▶ Concorrência de Componentes;
- ▶ Falta de relógio Global;
  - ▶ Coordenação em cooperação
    - ▶ Troca de mensagens
    - ▶ Noção compartilhada de tempo
- ▶ Falha de componentes independentes.
  
- ▶ Exemplos
  - ▶ Internet;
  - ▶ Intranet
  - ▶ Computação móvel e ubíqua (pervasiva).





Exemplos

# A Internet



Fonte: Coulouris, 2007

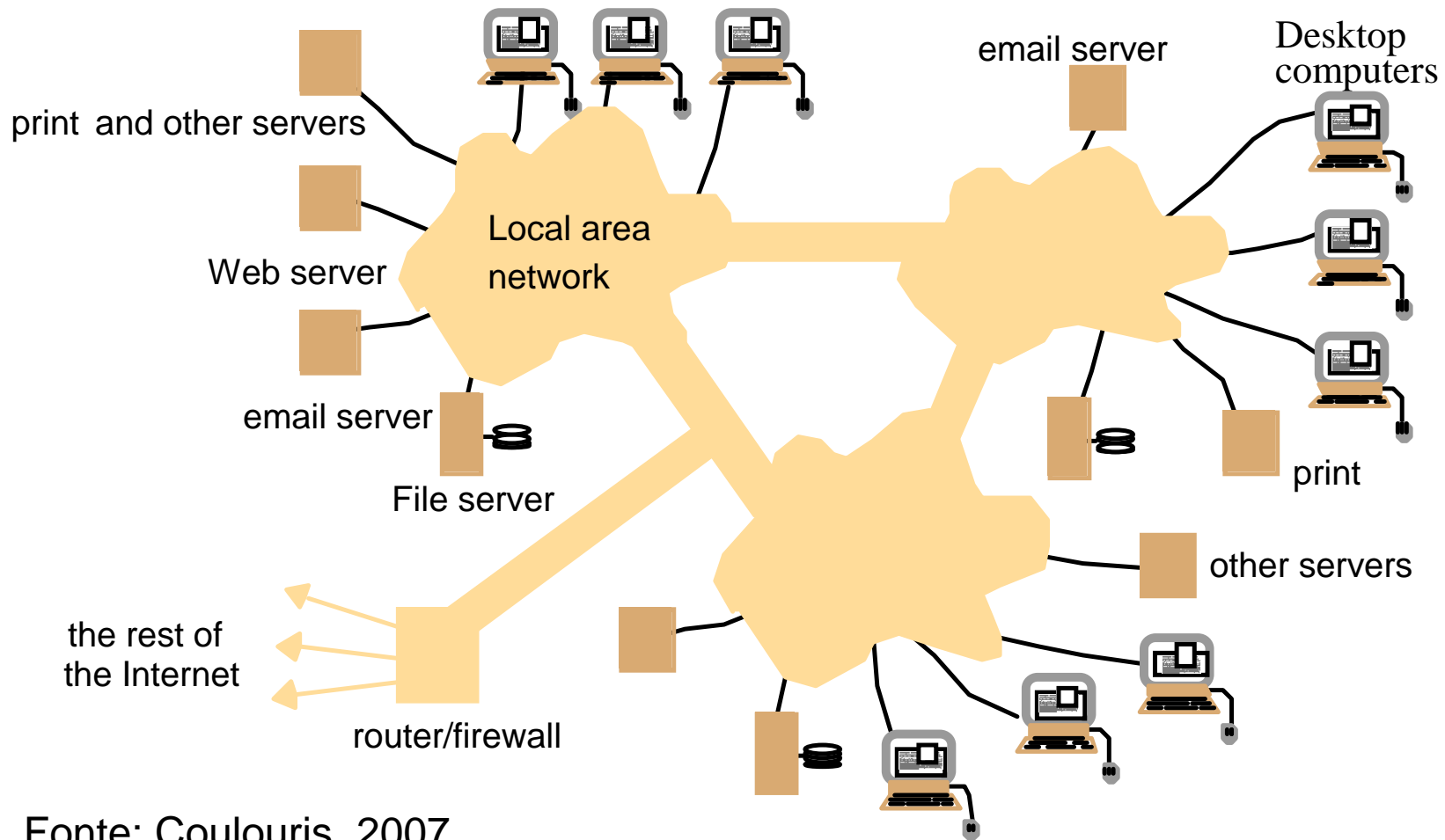
Conjunto de Intranets

# A Internet

---

- ▶ Conjunto de redes de computadores
  - ▶ Heterogeneidade
  - ▶ Interconexão
  - ▶ Interação entre processo
    - ▶ Troca de mensagens
      - Meio comum
  - ▶ Provedor de serviços
    - ▶ WWW, e-mail, transferência de arquivos.
  - ▶ Multimídia
    - ▶ Limite de banda
      - Reserva x qualidade

# Intranets



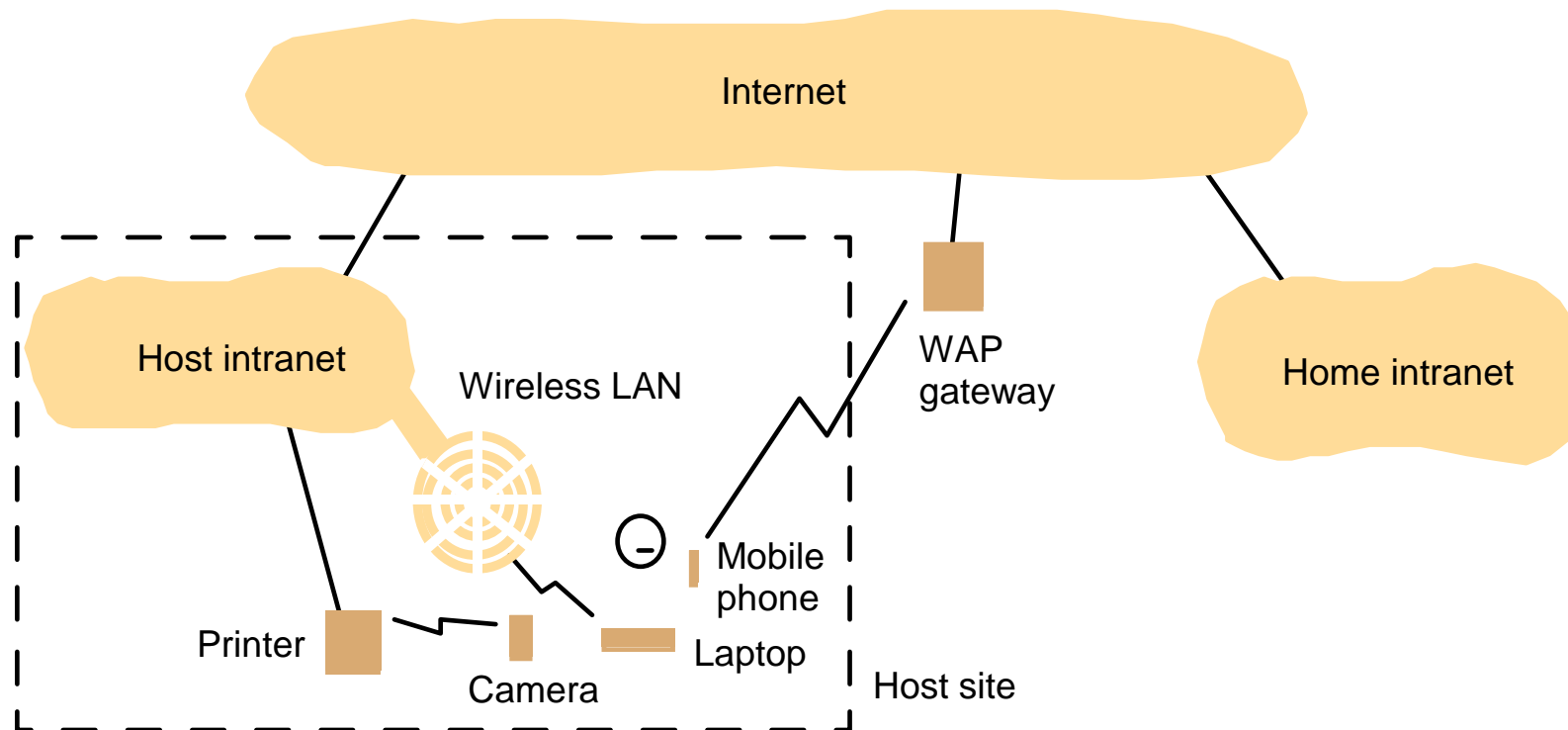
Fonte: Coulouris, 2007

# Intranets

---

- ▶ **Parte da Internet**
  - ▶ Administração particular
    - ▶ Limite e segurança
- ▶ **Conexão à Internet**
  - ▶ Roteador
  - ▶ Firewall

# Computação Móvel e Ubíqua



Fonte: Coulouris, 2007

# Computação Móvel e Ubíqua

---

- Computação Móvel (Nômade)
  - Dispositivos móveis
    - Redes sem fio
    - Miniaturização
      - PDA, celular, câmeras, dispositivos embarcados, etc.
    - Portabilidade
  - Movimentação do usuário com execução de tarefas
  - Computação com reconhecimento de localização ou contexto

# Computação Móvel e Ubíqua

---

- Computação Ubíqua (Pervasiva)

- ▶ Dispositivos computacionais pequenos e baratos
- ▶ Pervasivo
  - → Comportamento transparente



# Compartilhamento e a Web

---

- ▶ Cliente/Servidor
  - ▶ Compartilhamento de informação
    - ▶ Transparência
- ▶ No cliente (*browser*) novos formatos de documentos (não HTML) podem ser manipulados por *pluggins* ou *helpers* sem modificação no código do browser.
- ▶ O servidor apenas entrega documentos e não se preocupa com interfaces de usuário ou formatos de documentos.

# Compartilhamento e a Web

---

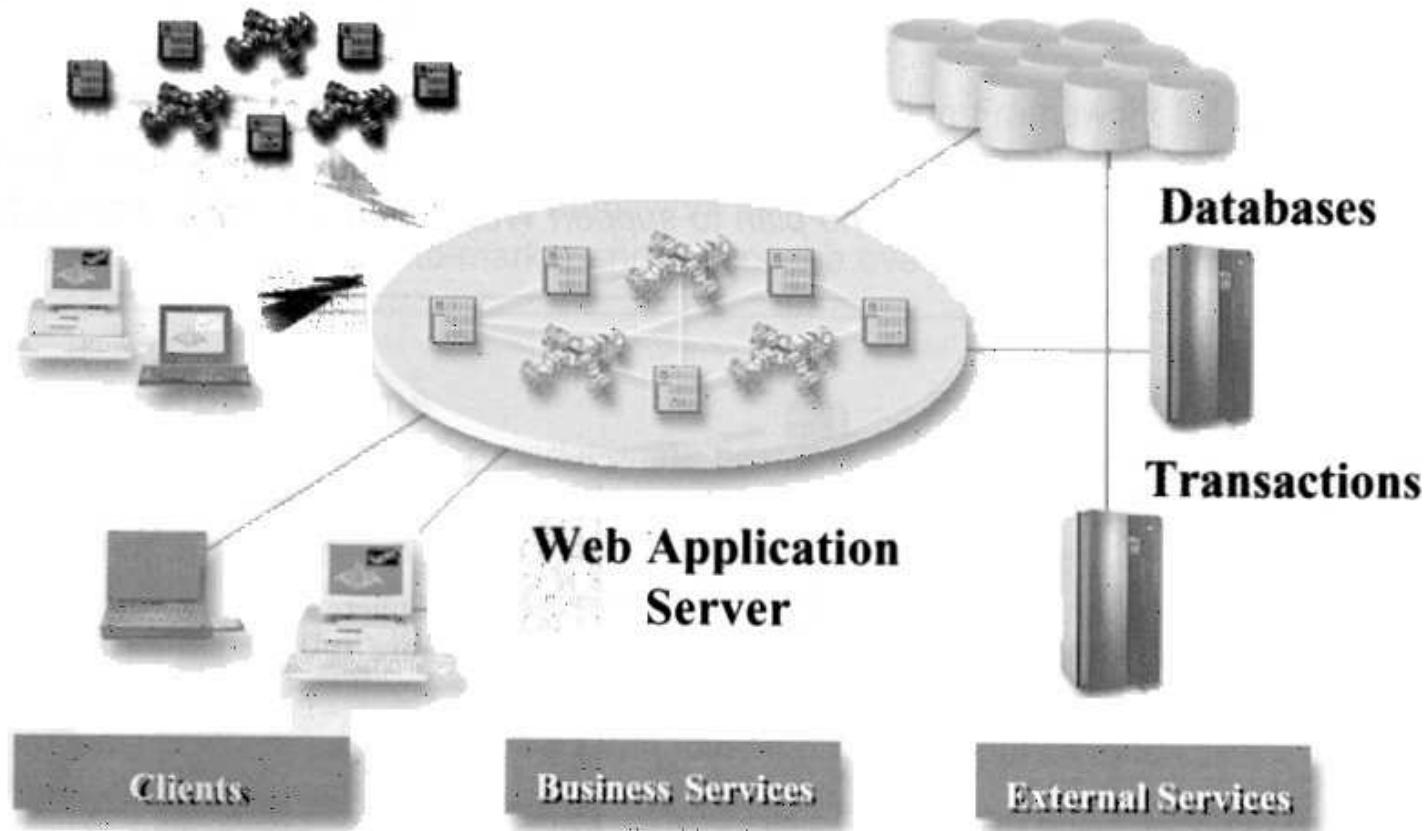
## ▶ Serviço

- ▶ Gerência de conjunto de recursos relacionados
- ▶ Arquivos, impressão, pagamento eletrônico
- ▶ Encapsulamento
  
- ▶ Servidor oferece serviços

# WWW

---

## World Wide Web



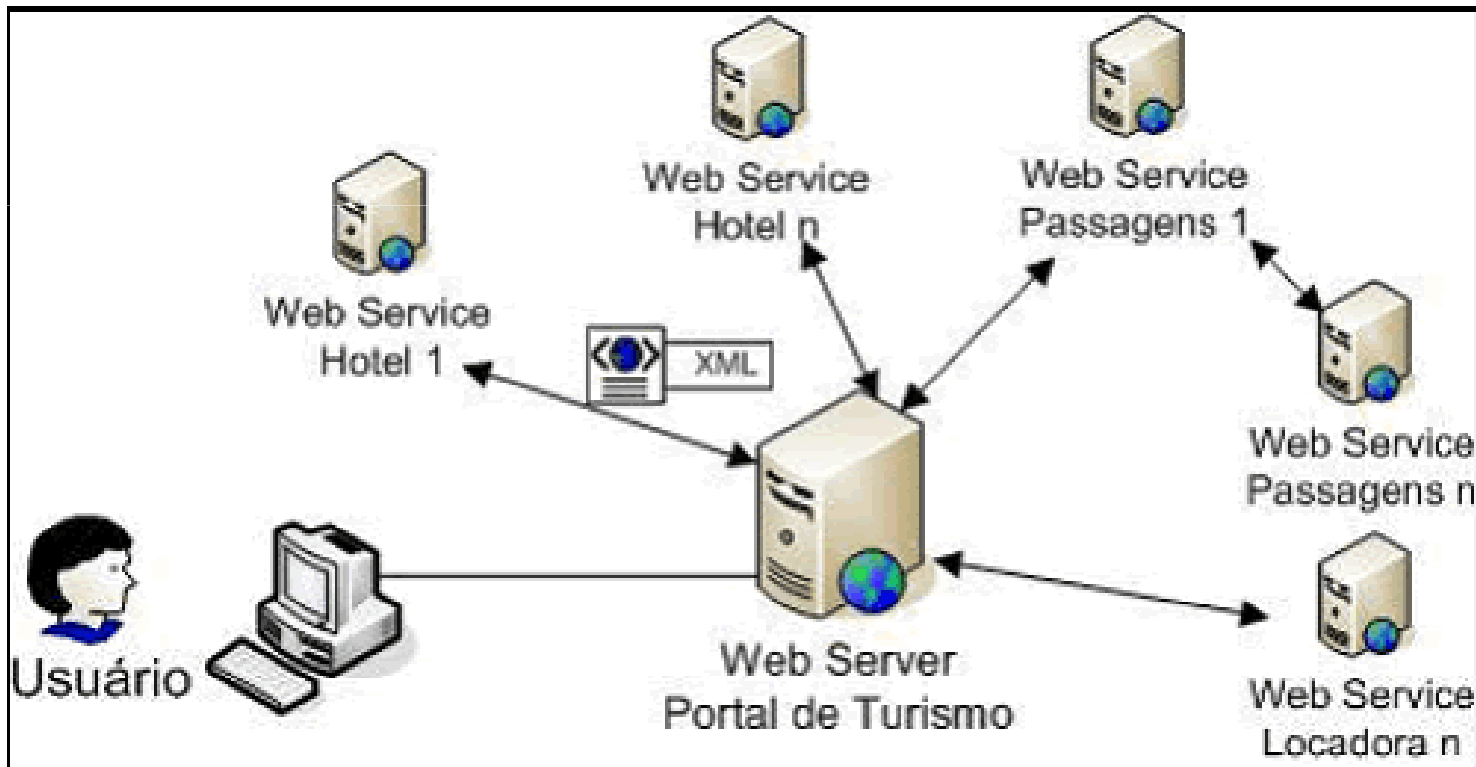
# WWW

---

- ▶ “Sistema em evolução para a publicação e o acesso a recursos e serviços pela Internet” – Coulouris.
  - ▶ *Browsers*
    - ▶ *Hipertexto*
      - *Links*

# Web Service

---



# Mais Exemplos

---

- ▶ Sistema de controle de tráfego aéreo;
- ▶ Sistemas bancários e comércio eletrônico;
- ▶ Robôs interconectados em indústria;
- ▶ Agentes móveis;
- ▶ Correio eletrônico;
- ▶ Vídeo-conferência;
- ▶ Jogos, etc.



Desafios

# DESAFIOS

---

- ▶ O projeto de um sistema distribuído deve levar em consideração os seguintes aspectos:
  - ▶ Heterogeneidade;
  - ▶ Sistemas abertos;
  - ▶ Segurança;
  - ▶ Escalabilidade;
  - ▶ Tratamento de falhas;
  - ▶ Concorrência;
  - ▶ Transparência.





Heterogeneidade

# Heterogeneidade

---

- ▶ **Aspectos:**
  - ▶ Redes de comunicação.
    - ▶ Protocolos
  - ▶ Hardware de computador.
    - ▶ Alta Ordem x Baixa Ordem
    - ▶ Representação de Caracteres
    - ▶ Pilha
  - ▶ Sistemas operacionais.
    - ▶ Kernel x Interface
  - ▶ Linguagens de Programação.
    - ▶ Representação (Decoradores em Java)
  - ▶ Desenvolvedores.
    - ▶ Padrões comuns.

# Heterogeneidade

---

## ○ Middleware

- Camada de Software
  - Abstração de programação
  - Mascaramento de heterogeneidade de rede, HW, SO e LP.
- Exemplos:
  - Corba
  - Java RMI
    - Uma única linguagem
  - MTS

## ○ Provê um modelo computacional uniforme

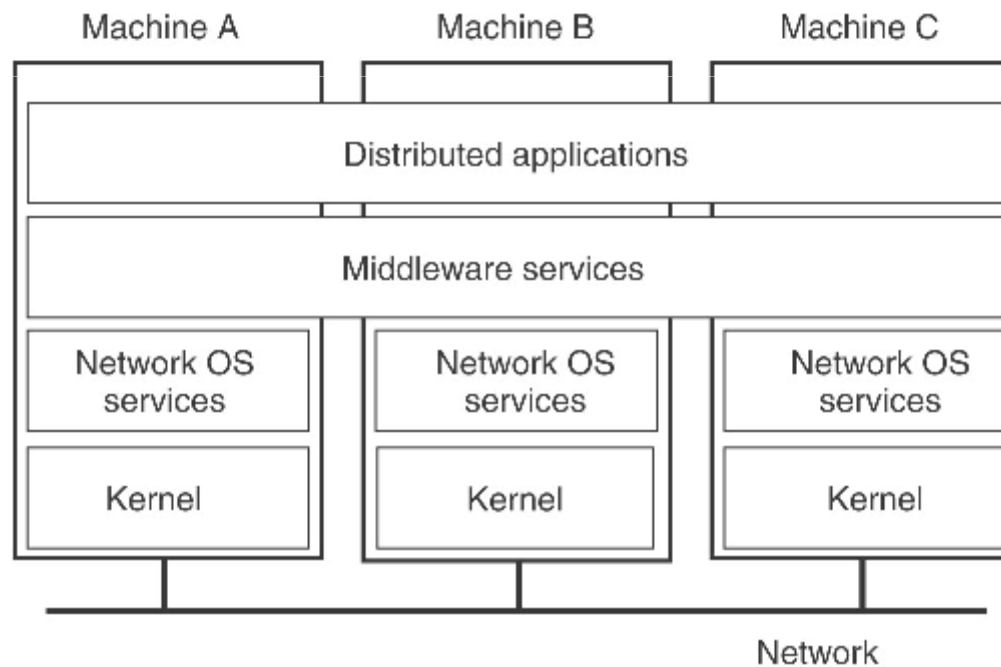
- Invocação remota de objetos;
- Notificação remota de eventos;
- Acesso remoto a SGBD;
- Processamento distribuído de transações.

# Heterogeneidade

---

## ▶ Middleware

- ▶ Common Object Request Broker - CORBA
- ▶ Remote Method Invocation – Java RMI





# Sistemas Abertos

# Sistemas Abertos

---

- ▶ Um sistema computacional é aberto quando ele pode ser estendido e reimplementado de várias maneiras.
  - ▶ Novos serviços para usuários
- ▶ A especificação e documentação das principais interfaces de software dos componentes de um sistema são publicadas.
- ▶ Mecanismo de comunicação uniforme.
- ▶ Independência de fornecedores.

# Sistemas Abertos

---

- ▶ Documentos de projetistas dos protocolos Internet
  - ▶ RFC - Request For Comments
    - ▶ Identificado por número
  - ▶ Pública
  
- ▶ Locais de publicações
  - ▶ Internet: [www.ietf.org](http://www.ietf.org) (RFC)
  - ▶ CORBA: [www.omg.org](http://www.omg.org)



Segurança



# Segurança

---

- ▶ **Confidencialidade**

- ▶ Proteção contra exposição para pessoas não autorizadas.

- ▶ **Integridade**

- ▶ Proteção contra alteração ou dano.

- ▶ **Disponibilidade**

- ▶ Proteção contra interferência com os meios de acesso aos recursos.

# Segurança

---

## ▶ Desafio

- ▶ Troca de mensagens de forma segura.
- ▶ Ocultar mensagem X validar usuário

## ▶ Mecanismos de proteção

- ▶ Firewall: proteção da Intranet.
- ▶ Criptografia: o destinatário é o correto?
- ▶ Autenticação: quem é o usuário?

## ▶ Ataques complicados

- ▶ Denial of Service (DoS): interrupção do serviço.
- ▶ Segurança de código móvel: execução perigosa!



Escalabilidade

# Escalabilidade

---

- ▶ Um sistema é descrito com o **escalável** se permanece eficiente quando há um aumento significativo no número de recursos e no número de usuários.
- ▶ Ex.: Internet.

<i>Date</i>	<i>Computers</i>	<i>Web servers</i>
1979, Dec.	188	0
1989, July	130,000	0
1999, July	56,218,000	5,560,866
2003, Jan.	171,638,297	35,424,956

Fonte: Coulouris, 2007

# Escalabilidade

---

<i>Date</i>	<i>Computers</i>	<i>Web servers</i>	<i>Percentage</i>
1993, July	1,776,000	130	0.008
1995, July	6,642,000	23,500	0.4
1997, July	19,540,000	1,203,096	6.2
1999, July	56,218,000	6,598,697	12.4
2001, July	125,888,197	31,299,592	25
		42,298,371	

Introdução aos Sistemas  
Disciplina  
www.edson.com.br

Fonte: Coulouris, 2007

# Escalabilidade

---

- ▶ **Desafios no projeto de sistemas distribuídos:**
  - ▶ Controlar o custo de recursos físicos.  $O(n)$ .
    - ▶ Ampliação dos recursos a custos razoáveis para atender a demanda.
    - ▶ Servidor de arquivos  $X$  acessos.
      - Adição de servidores
      - Se um servidor atende a  $y$  usuários, 2 servidores atendem a  $2y$ ?
    - ▶ Seja  $n$  o número de usuários do sistema.

# Escalabilidade

---

- Desafios no projeto de sistemas distribuídos:
  - Controlar a perda de desempenho.
    - Estruturas hierárquicas X lineares
      - Gerenciamento de dados
      - Seja  $n$  o tamanho do conjunto de dados.
      - Perda de desempenho máxima:  $O(\log n)$ .
  - Impedir que os recursos de software se esgotem.
    - Ex.: Endereçamento IP (32 bits).
    - Superestimar é correto?
  - Evitar gargalos de desempenho.
    - Arquivo, HOSTS, página Web, etc.
    - Algumas técnicas com dados replicados, uso de cache, distribuição de vários servidores, etc.



# Tratamento de Falhas



# Tratamento de Falhas

---

- ▶ Falhas em um sistema distribuído
  - ▶ Parciais
    - ▶ Alguns componentes falham, enquanto outros continuam funcionando.
      - Tratamento difícil.
- ▶ Técnicas:
  - ▶ Detecção de falhas.
    - ▶ CRC
  - ▶ Mascaramento de falhas.
    - ▶ Retransmissão de mensagens
    - ▶ Replicação de dados.
    - ▶ Piores casos?
      - Todos os dados falham, retransmissão fora do prazo.

# Tratamento de Falhas

---

- ▶ **Técnicas:**
  - ▶ Tolerância a falhas.
    - ▶ Tratar tudo de todos os componentes?
      - Custo?
    - ▶ Navegadores Web
      - Informação ao cliente
      - Delega decisão
  - ▶ Recuperação de falhas.
    - ▶ Banco de dados x Checkpoint X Log
  - ▶ Redundância
    - ▶ Rotas entre roteadores, Clusters, etc.
  
- ▶ **Disponibilidade de Hardware**



Concorrência

# Concorrência

---

- ▶ **Acesso compartilhado ao mesmo recurso no mesmo tempo.**
  - ▶ IRPF, Leilão, etc.
- ▶ **Processos ou Threads.**
  - ▶ Controle de concorrência.
- ▶ **Operações sincronizadas.**
  - ▶ Mantém a consistência dos dados.
  - ▶ Técnicas padrão como semáforo.



Transparência

# Transparência

---

- ▶ O sistema é percebido como um todo, em vez de uma coleção de componentes independentes.
  - ▶ Esconde do usuário e do programador de aplicação a separação de componentes em um SD.
- ▶ **Transparência de rede. (Ex.: correio na Internet)**
  - ▶ **Transparência de acesso.**
    - ▶ Exemplo: Visão única de estrutura de diretório em SAD.
    - ▶ Contra exemplo: Sistemas de arquivos de pastas que necessitam de FTP.
  - ▶ **Transparência de localização.**
    - ▶ Ex.: URL, mas não quanto a mobilidade – apontadores em outros sites)
- ▶ **Transparência de concorrência.**

# Transparência

---

- ▶ **Transparência de replicação.**
- ▶ **Transparência de falhas.**
  - ▶ Ex.: e-mail
  - ▶ Geralmente, o *middleware* converte as falhas de rede e processos em exceções em nível de programação.
- ▶ **Transparência de mobilidade.**
  - ▶ Ex.: celular
- ▶ **Transparência de desempenho.**
- ▶ **Transparência de escalabilidade.**



Vantagens



# Vantagens

---

- ▶ **Economia**
  - ▶ Melhor relação custo X benefício.
- ▶ **Eficiência**
  - ▶ Maior poder de computação.
- ▶ **Maior confiabilidade e disponibilidade**
  - ▶ Balanceamento de carga, independência
- ▶ **Solução para problemas naturalmente distribuídos.**
- ▶ **Maior escalabilidade.**
- ▶ **Crescimento incremental.**



Desvantagens

# Desvantagens

---

- ▶ **Infraestrutura de comunicação**
  - ▶ Dependência total das redes de comunicação
  - ▶ Vulnerabilidade das redes
    - ▶ Erros, falhas, etc.
    - ▶ Sobrecarga, performance, etc.
- ▶ **Segurança**
  - ▶ Proteção dos dados.
  - ▶ Várias portas de acesso.
- ▶ **Que SO, LP e aplicação é apropriado para SD?**

## Referências

---

- ▶ **Couloris, G. & Dollimore, J. & Kindberg T.** Sistemas Distribuídos: Conceitos e Projetos. Porto Alegre: Bookman, 4<sup>a</sup> edição, 2007.
- ▶ **Tanenbaum, A. S. & Steen, M.V.** Distributed Systems – Principles and Paradigms. Prentice-Hall, 1<sup>a</sup> edição. 2001.