

CURSO DE ENGENHARIA DE SOFTWARE

Disciplina: Sistemas Distribuídos

CONCEITOS BÁSICOS

Prof. M.e Alexandre Tannus

Introdução

Características de Sistemas Distribuídos

Desafios de implementação

- ▶ Rede
 - ▶ Conjunto de entidades interligadas entre si



Rede de dormir

Rede de pesca



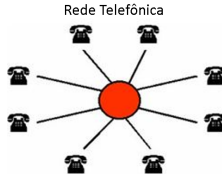
Rede de pessoas



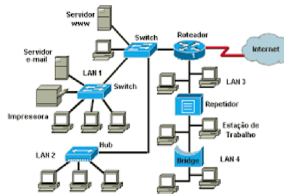
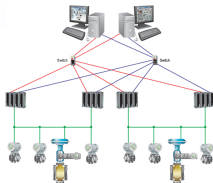
Rede de Transportes



Rede Neuronal



Rede Telefônica

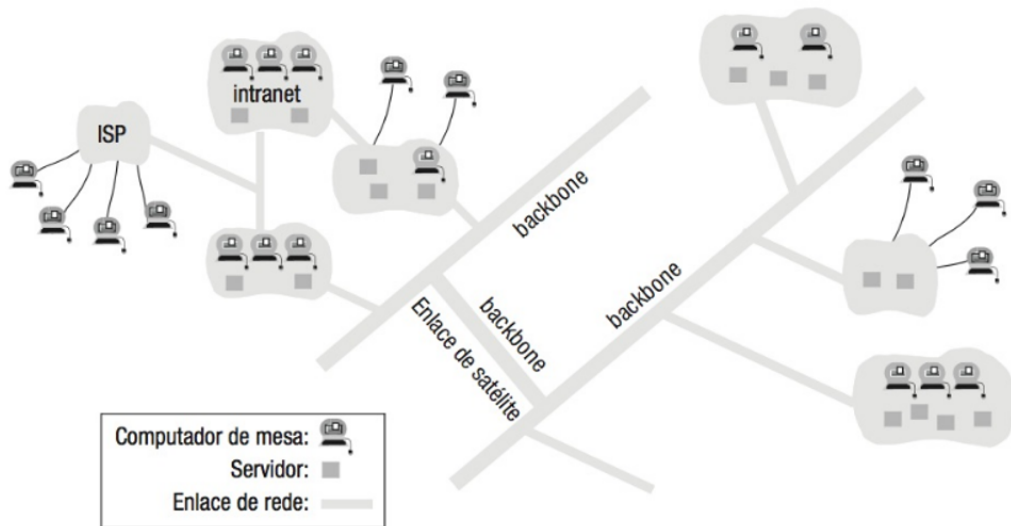


- ▶ As redes surgiram da necessidade de compartilhar dados em tempo hábil.
- ▶ Os computadores pessoais são ferramentas de trabalho ótimas para produzir dados, gráficos e outros tipos de informação, mas não possibilitam que você compartilhe rapidamente os dados que criou.

- ▶ Um conjunto de computadores e outros dispositivos conectados juntos chama-se REDE, assim como o conceito de computadores compartilhando os recursos.
- ▶ Um computador conectado a outros pode compartilhar os dados dos outros computadores, impressoras e outros dispositivos.

- ▶ Os computadores que fazem parte de uma rede podem compartilhar
 - ▶ Dados
 - ▶ Mensagens
 - ▶ Gráficos
 - ▶ Impressoras
 - ▶ Aparelhos de fax
 - ▶ Modems
 - ▶ Outros recursos de Hardware

- ▶ Um **sistema distribuído** é aquele no qual os componentes localizados em **computadores** interligados em rede se **comunicam** e coordenam suas ações apenas passando **mensagens**.





- ▶ Internet of Things (IoT)
- ▶ Sistemas Multimídia
 - ▶ Spotify
 - ▶ YouTube
 - ▶ Netflix
- ▶ Nuvem
 - ▶ Execução de aplicativos
 - ▶ Armazenamento
 - ▶ Processamento Computacional
- ▶ Computação móvel
- ▶ Grid Computing

- ▶ **Serviço:** parte distinta de um sistema computacional que gerencia um conjunto de recursos relacionados e apresenta sua funcionalidade para usuários e aplicativos.
 - ▶ Acesso limitado pelo conjunto de operações permitidas
 - ▶ Evita que sejam feitas alterações indesejadas no sistema por pessoas não autorizadas

- ▶ **Servidor:** Programa (processo) em execução em um nó da rede que aceita pedidos de execução de programas vindos de outros nós (clientes).
 - ▶ Arquitetura cliente-servidor
 - ▶ Podem ser implementados em forma de objetos
 - ▶ Exemplos de utilização de arquitetura cliente-servidor
 - ▶ WWW
 - ▶ Email
 - ▶ Impressoras

- ▶ Concorrência de componentes
 - ▶ Realização de uma tarefa por diversas máquinas
- ▶ Falta de um relógio global
 - ▶ Podem gerar falhas devido à falta de sincronização
- ▶ Falhas de componentes independentes.
 - ▶ Podem ocorrer de forma isolada, porém afetando toda a rede

- ▶ Heterogeneidade
- ▶ Sistemas abertos
- ▶ Segurança
- ▶ Estabilidade
- ▶ Tratamento de falhas
- ▶ Transparência

- ▶ Diferença entre os computadores e redes que executam os processos
- ▶ Aplicável a
 - ▶ Redes
 - ▶ Hardware
 - ▶ Sistemas operacionais
 - ▶ Linguagens de programação
 - ▶ Implementação

- ▶ Camada de software que oferece abstração de programação
- ▶ Trata das diferenças em nível dos sistemas operacionais e do hardware
- ▶ Exemplos
 - ▶ CORBA (*Common Object Request Broker*)
 - ▶ JAVA RMI

- ▶ Invocação remota de objetos
- ▶ Notificação de eventos
- ▶ Acesso a banco de dados
- ▶ Processamento de transação

- ▶ Oferecem a possibilidade de extensão e reimplementação de funções do sistema
 - ▶ Recompilar o kernel
- ▶ Em geral utilizam licenças que permitem a alteração do código pelos desenvolvedores sem nenhum custo (GPL, LGPL)

- ▶ Certos dados devem ser mantidos de forma confidencial no sistema, permitindo acesso apenas ao pessoal autorizado
- ▶ Componentes de segurança
 - ▶ Confidencialidade
 - ▶ Integridade
 - ▶ Disponibilidade

- ▶ Manutenção da eficiência do serviço com o aumento do número de usuários
- ▶ Desafios
 - ▶ Controle de custos
 - ▶ Perda de desempenho
 - ▶ Esgotamento de recursos
 - ▶ Gargalos de desempenho

- ▶ Compartilhamento de recursos pelos clientes pode ocasionar o acesso simultâneo
- ▶ Exige sincronização para manutenção da consistência

- ▶ Visão do sistema como um todo pelo usuário, sem necessidade de conhecimento dos componentes
- ▶ Tipos
 - ▶ Acesso
 - ▶ Localização
 - ▶ Concorrência
 - ▶ Replicação
 - ▶ Falhas
 - ▶ Mobilidade
 - ▶ Desempenho
 - ▶ Escalabilidade

