

XXXX

Sistemas Distribuídos

Eduardo Furlan Miranda

2025-07-01

Adaptado de: PEREIRA, C. S. Sistemas Distribuídos.
Londrina: EDE SA, 2019. ISBN 978-85-522-1443-4.

Introdução a Sistemas Distribuídos

- Entender sistemas distribuídos é de extrema importância
- Muitas aplicações populares utilizam este modelo
- Compreender a evolução dos sistemas computacionais
- Classificar diferentes sistemas e suas características

Oportunidades em TI

- Demanda por profissionais em SD aumentou
- Principalmente em consultorias de TI especializadas
- Estágio envolve avaliar ativos de TI de clientes
- Conhecimento em SD pode ganhar a confiança do cliente

Fundamentos de Sistemas Distribuídos

- Inicia o estudo de sistemas distribuídos
- Proporciona oportunidades no mercado de trabalho
- Exemplo: o jogo on-line Smite usa arquitetura distribuída
- Compreender sistemas e aplicações complexas
- Abordar conceitos e tipos de interligações

Exemplo: Missão no Estágio

- Primeira atividade: avaliar ativos de TI de um cliente
- Cliente não possui departamento de informática
- Explicar como os computadores se comunicam
- Elaborar apresentação no formato de slides

O Que É Sistema Distribuído

- Conjunto de computadores interligados via rede
- Aparenta ser um sistema único para o usuário final
- Computadores funcionam de forma independente
- Não depende de uma única máquina
- Toda a carga é distribuída entre várias máquinas

Referência: Camadas SD

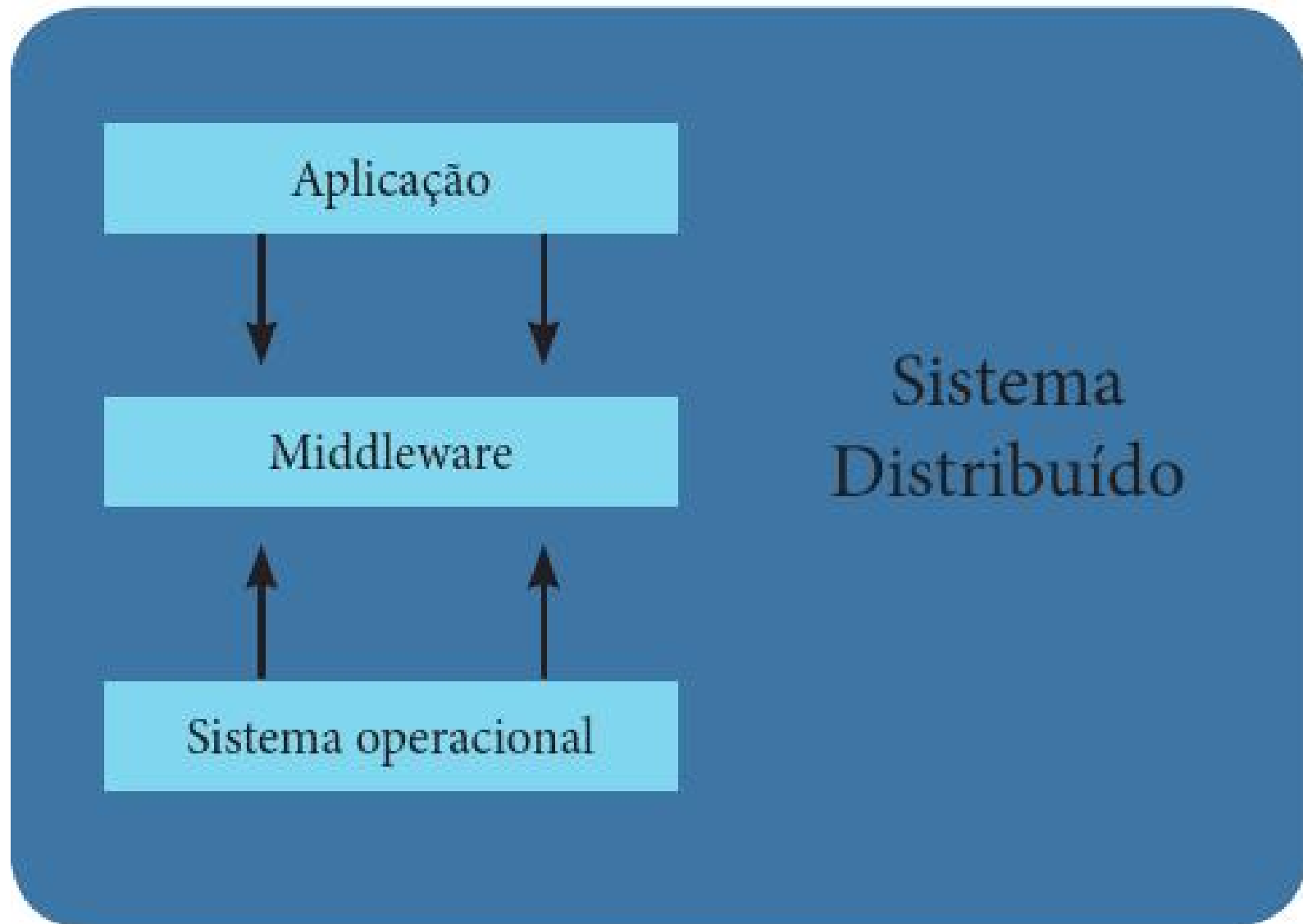


Figura 1.1 | Camadas que compõem um sistema distribuído

Função do Middleware

- Camada principal para aplicações distribuídas
- Software entre programas e sistema operacional
- Permite gerenciamento de dados e comunicação
- Funciona como uma camada de tradução

Exemplos de Aplicações SD

- Redes sociais com conteúdo multimídia
- Sites de pesquisas e plataformas de vídeos online
- Modelos de sistemas distribuídos definem a interação de objetos

Objetivos dos Sistemas Distribuídos

- Alta disponibilidade e fácil acesso ao sistema
- Ocultar ao usuário que os recursos são distribuídos
- Sistema deve ser aberto e expansível
- Facilitar a inclusão de novas máquinas e recursos

Exemplo Prático de SD

- Portal de notícias com modelo de sistemas distribuídos
- Página fornecida por um conjunto de servidores
- Para o usuário, os servidores aparentam ser únicos
- Em caso de falha de um servidor, os outros assumem a função

Arquiteturas de Redes

- Existem três tipos de arquiteturas de computadores
- Arquitetura Cliente-Servidor (a “clássica”)
- Arquitetura Ponto a Ponto (P2P)
- Arquitetura Descentralizada (híbrida)

Referência: Cliente-Servidor

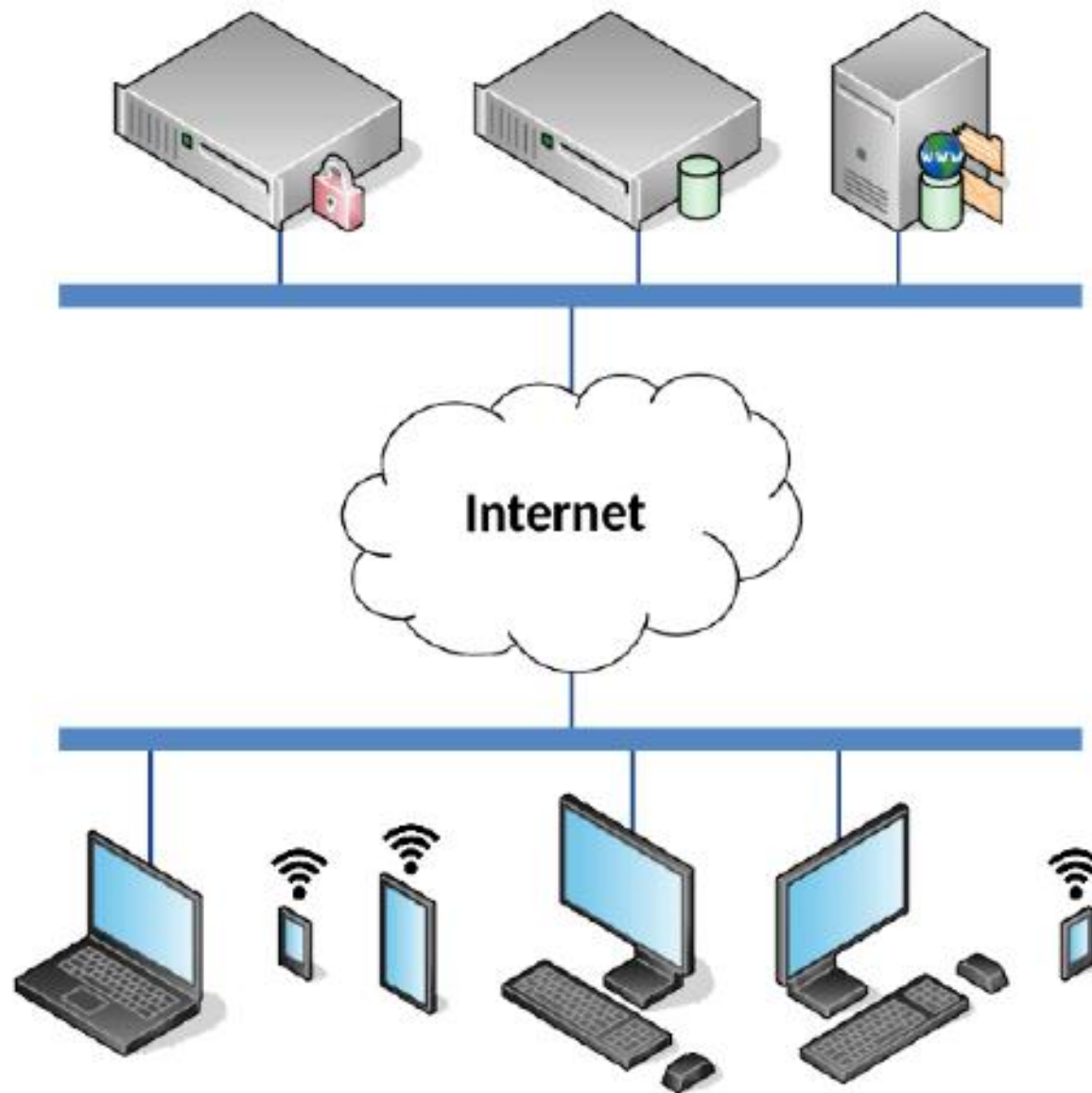


Figura 1.2 | Arquitetura cliente-servidor

Arquitetura Cliente-Servidor

- Origem na década de 70
- É a arquitetura mais conhecida e utilizada
- Serviços e recursos compartilhados em servidores
- Servidores podem ser multisserviço ou segregados

Cliente-Servidor: Conexão e Usos

- Servidores conectados via equipamentos de rede
- Acesso remoto por máquinas cliente
- Funciona também em rede local (LAN)
- Uso comum: acessar websites de comércio eletrônico
- Uso comum: verificar e-mails no smartphone
- Uso comum: jogar games online (ex: Call of Duty)

Referência: Ponto a Ponto

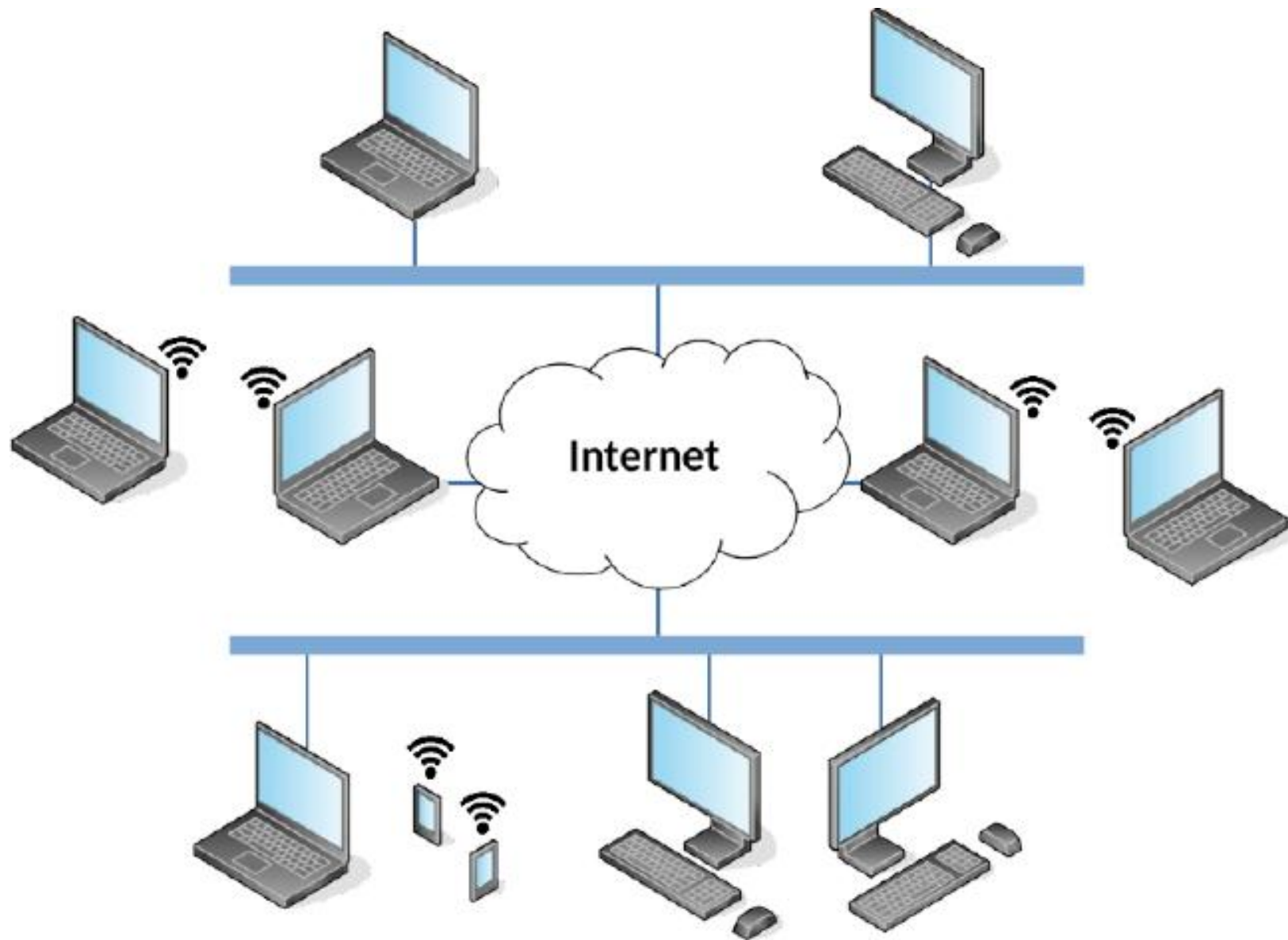


Figura 1.3 | Arquitetura ponto a ponto

Arquitetura Ponto a Ponto

- Conhecida como P2P, origem na década de 80
- Computadores têm o mesmo papel na rede
- Funcionam como dispositivos finais e como servidores
- Podem disponibilizar e consumir recursos

Ponto a Ponto: Conectividade e Usos

- Conectados à Internet via equipamentos de rede
- Funciona também em rede local
- Exemplo: compartilhar foto via Bluetooth
- Exemplo: download de arquivo grande via torrent
- Vantagem: download mais rápido de diversas fontes

Referência: Descentralizada

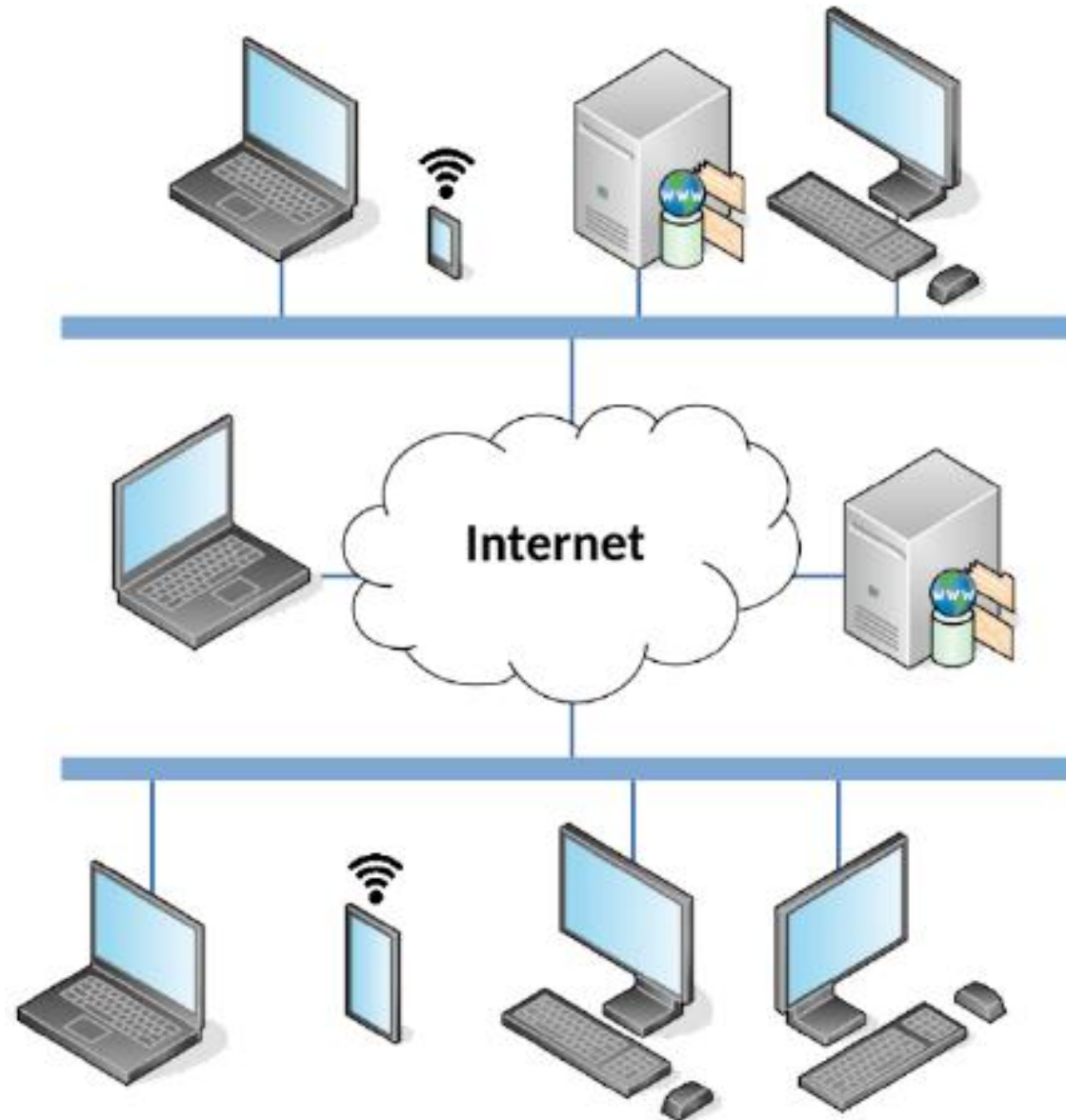


Figura 1.4 | Arquitetura descentralizada

Arquitetura Descentralizada

- Arquitetura mais recente, a partir dos anos 2000
- Considerada uma arquitetura híbrida
- Computadores são os próprios servidores da aplicação
- O estado da aplicação é replicado entre computadores
- Existe um consenso entre eles na rede

Descentralizada: Características e Vantagens

21/30

- Pode haver computadores com papel puramente de “clientes”
- Não há uma entidade que controle a aplicação
- Garante transparência aos usuários

Descentralizada: Aplicações Comuns

- Utilizada por plataformas baseadas em Blockchain
- Aplicações são chamadas de DApps
- Exemplos em sistemas de transações financeiras
- Criptomoedas, como o Bitcoin
- Exemplo de rede social: Steemit

Integração de Sistemas

- Troca de informação entre sistemas computacionais
- Comunicação para compartilhamento de dados
- Exemplo: aplicativo do Facebook e seus servidores

Front-end e Back-end

- Aplicações, principalmente web, divididas em duas partes
- Front-end (client-side): parte de interação do usuário
- Back-end (server-side): cuida das regras de negócio
- Lida com banco de dados e validações

Integração via Protocolo HTTP

- A integração entre front-end e back-end ocorre via HTTP
- Protocolo padrão utilizado na navegação web
- Dados são enviados para validação de informações
- HTTP é uma das formas de integração, não a única

Situação-Problema: Acesso Web

- Navegar na Internet é uma tarefa rotineira
- Acessar um site de previsão meteorológica
- O site utiliza arquitetura cliente-servidor
- Descrever as etapas da comunicação cliente-servidor

Comunicação Cliente-Servidor: Passos

- O processo pode ser descrito em seis passos
- 1 - Máquina Cliente: o navegador chama a URL do site
- 2 - Máquina Cliente: envia uma requisição do tipo HTTP
- 3 - Máquina Servidor: recebe a requisição HTTP

Comunicação Cliente-Servidor: Resposta

- 4 - Máquina Servidor: envia uma resposta via protocolo HTTP com a página
- 5 - Máquina Cliente: recebe a resposta via protocolo HTTP do servidor
- 6 - Máquina Cliente: baixa a página e apresenta o resultado
- A página terá conteúdo multimídia (imagens, vídeos, links)

Características dos Sistemas Distribuídos

- A aplicação é replicada ou distribuída entre máquinas
- Comporta-se como se estivesse rodando em uma única máquina
- Possuem um elemento central: o middleware
- Utilizados em aplicações modernas de grande porte
- Facilidade para incluir novas máquinas em funcionamento

Desvantagem Cliente-Servidor

- Máquinas clientes podem gerar requisições
- Não podem oferecê-las a outras máquinas
- Pode gerar uma sobrecarga no servidor