



Universidade Tecnológica Federal do Paraná
Departamento Acadêmico de Computação
Bacharelado em Ciência da Computação

Sistemas Distribuídos

Modelos de Sistemas

Prof. Rodrigo Campiolo

14/03/22

Tópicos

- Introdução
- Modelos de Arquitetura
- Modelos Fundamentais
 - Modelo de interação
 - Modelo de falha
 - Modelo de segurança

Introdução

- Apresentar modelos que possibilitam tratar propriedades e problemas presentes no projeto de vários sistemas distribuídos.
- Três tipos de modelos:
 - Físicos: descreve o hardware e a interconexão em rede.
 - Arquiteturais: definem a forma como os componentes se relacionam entre si e como são mapeados em uma rede de computadores.
 - Fundamentais: possibilitam racionalizar de forma abstrata sobre características de sistemas distribuídos.

Introdução

- Dificuldades e ameaças aos SDs
 - Variação nas formas de utilização
 - variação de carga, partes desconexas, especificidades das aplicações.
 - Variação de ambientes
 - heterogeneidade de hardware, SO e redes
 - diferenças de escala
 - Problemas internos
 - Relógios não sincronizados, atualizações conflitantes, falhas de software e hardware.
 - Ameaças externas
 - Ataques à integridade de dados e sigilo, DoS

Definições

- **Arquitetura de software:** descreve a organização e interação dos componentes de software; centrada na organização lógica do software.
- **Arquitetura de sistema:** descreve a disposição dos componentes de software no ambiente físico (máquinas e redes de computadores) e como se relacionam entre si.

Arquiteturas de software

- Camadas
 - 3-camadas
- Baseada em objetos
 - Objetos distribuídos
- Orientadas a serviços
 - SOA
 - Microserviços
- Baseadas em recursos
 - RESTFul
- Baseadas em eventos
 - Publish-Subscriber

Atividades

- Defina e compare os estilos arquiteturais: RESTFul, SOA, Microsserviços.
- Faça a leitura e a resenha Azure Application Architecture Guide → Architecture Styles.

<https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/>

▼ Architecture styles

Overview

Big compute

Big data

Event-driven architecture

Microservices

N-tier application

Web-queue-worker

Arquiteturas de sistema

- Questões básicas:
 - Quais são as entidades que estão se comunicando no SD?
 - Qual o paradigma de comunicação utilizado?
 - Quais funções e responsabilidades estão relacionadas as entidades na arquitetura?
 - Como os componentes são mapeados na infraestrutura distribuída física (qual é sua localização)?
- Podem ser classificadas em:
 - Centralizadas
 - Descentralizadas
 - Híbridas

Arquiteturas de sistema

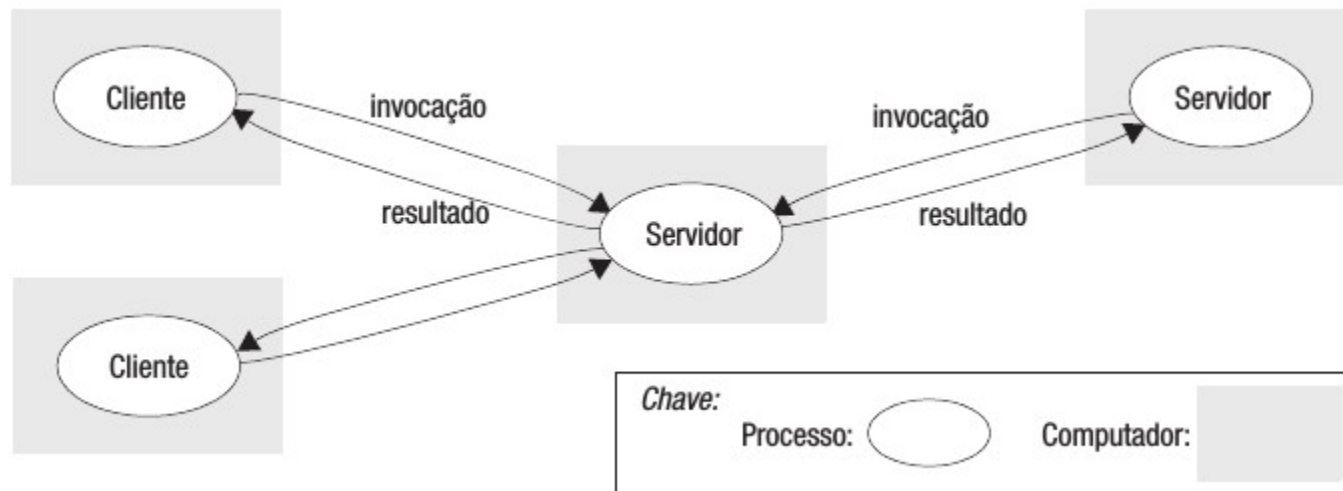
- Entidades e paradigmas de comunicação

Entidades em comunicação (o que se comunica)		Paradigmas de comunicação (como se comunicam)		
Orientados a sistemas	Orientados a problemas	Entre processos	Invocação remota	Comunicação indireta
Nós	Objetos	Passagem de mensagem	Requisição-resposta	Comunicação em grupo
Processos	Componentes	Soquetes	RPC	Publicar-assinar
	Serviços Web	Multicast	RMI	Fila de mensagem
				Espaço de tupla
				DSM

Figura 2.2 Entidades e paradigmas de comunicação.

Arquiteturas de sistema

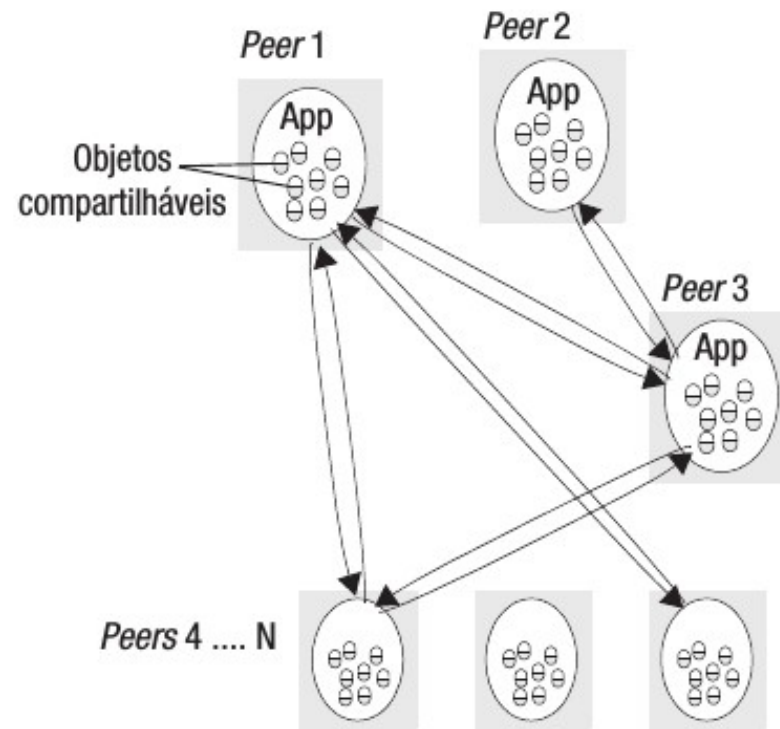
- Cliente/Servidor



- Processos com papéis bem definidos.
- Processos clientes solicitam serviços para processos servidores.

Arquiteturas de sistema

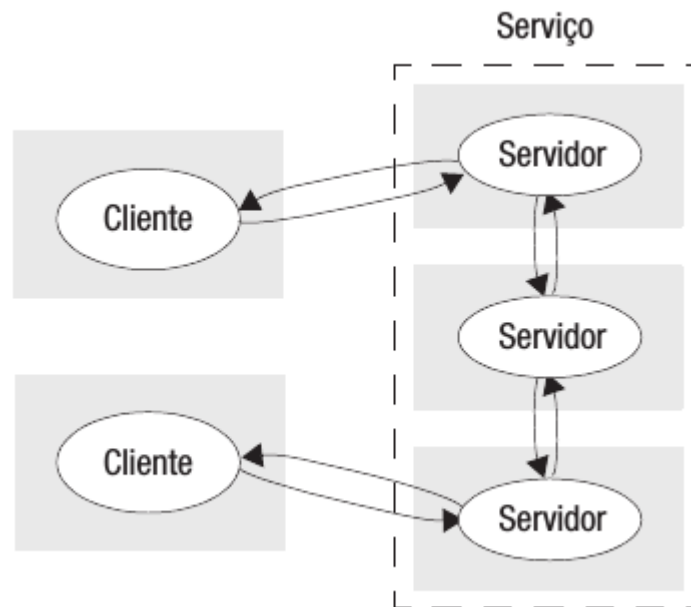
- Peer-to-Peer (P2P)



- Processos com papéis semelhantes.
- Processos atuam como pares, não há distinção entre cliente e servidor.

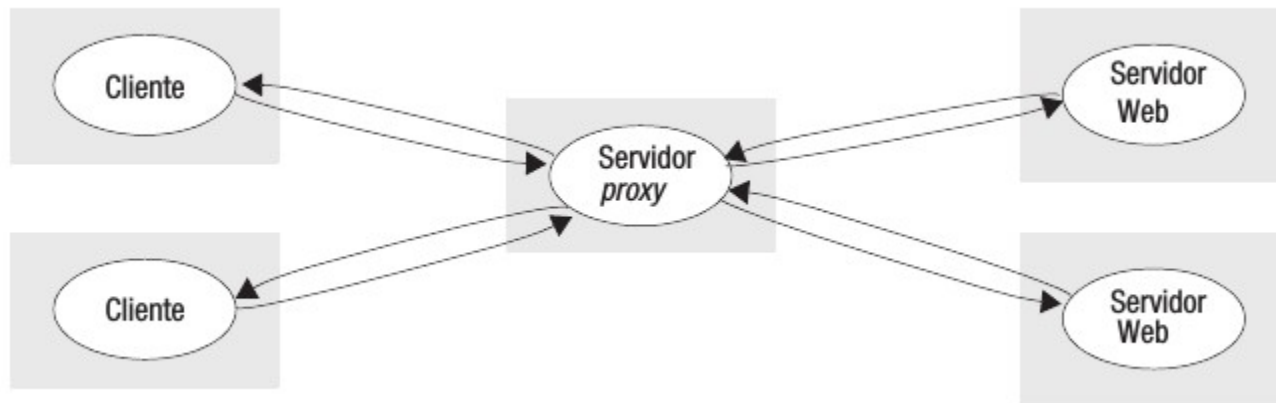
Arquiteturas de sistema

- Variações com Cliente/Servidor
 - Múltiplos servidores



Arquiteturas de sistema

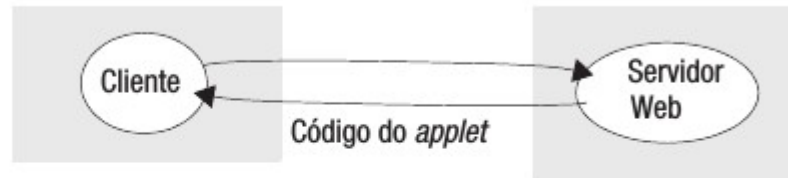
- Variações com Cliente/Servidor
 - Uso de cache



Arquiteturas de sistema

- Variações com Cliente/Servidor
 - Código móvel

a) Requisição do cliente resulta no download do código de um *applet*

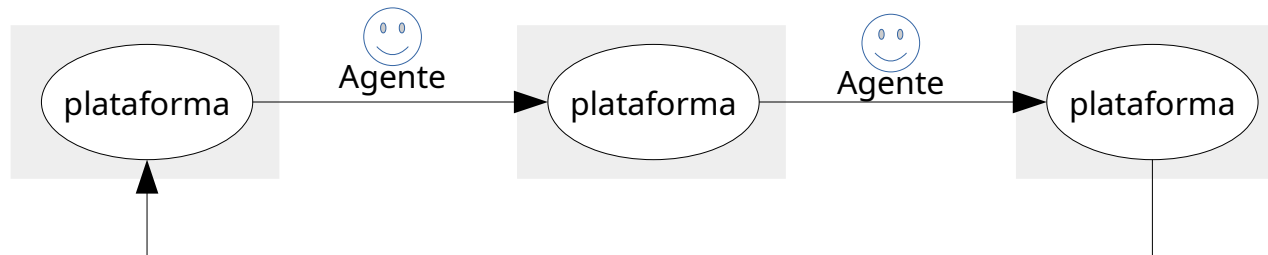


b) O cliente interage com o *applet*



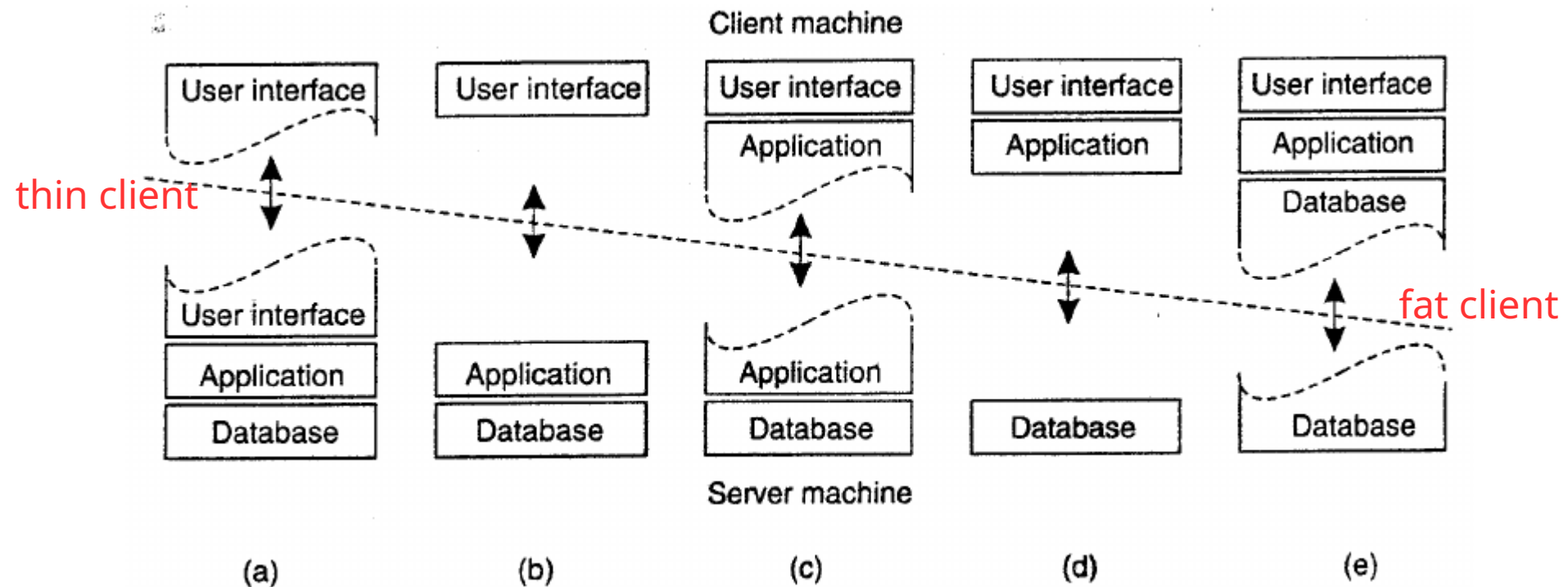
Arquiteturas de sistema

- Variações com Cliente/Servidor
 - Agentes móveis



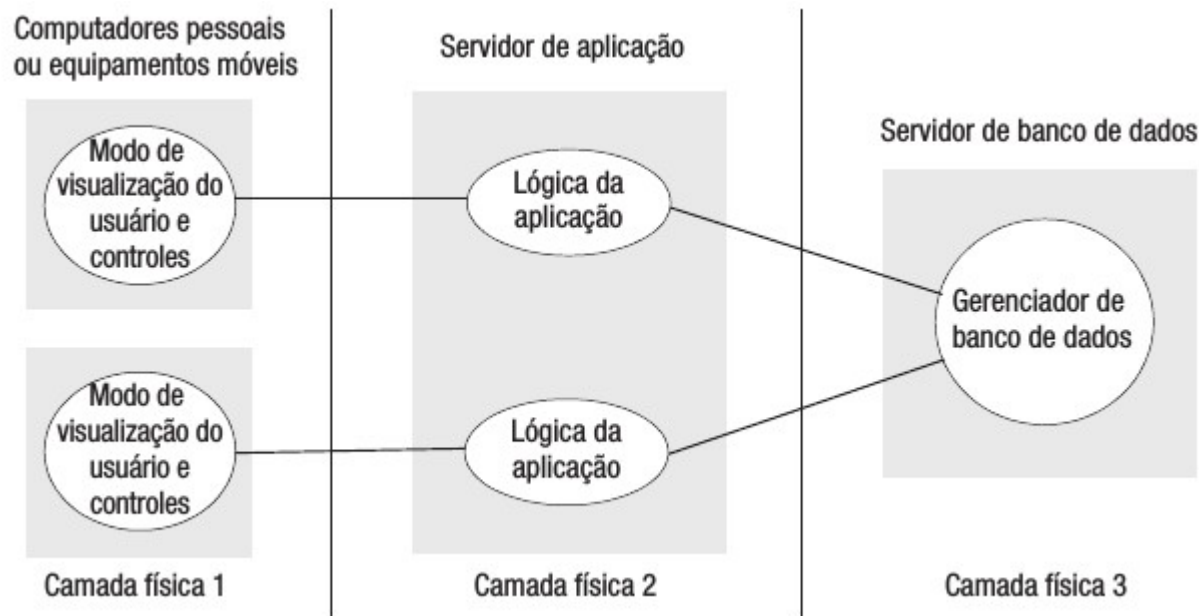
Arquiteturas de sistema

- Variações com Cliente/Servidor
 - Variações com duas camadas físicas (2-tier)



Arquiteturas de sistema

- Variações com Cliente/Servidor
 - Três camadas físicas (3-tier)



Arquiteturas de sistema

- Uso de ***middlewares***
 - Camada de software
 - Mascarar a heterogeneidade
 - Prover um modelo de programação
 - Prover serviços comuns a aplicações distribuídas



Arquiteturas de sistema

- Exemplo de ***middlewares***
 - Objetos distribuídos (CORBA, JAVA RMI)
 - Componentes distribuídos (Servidores de aplicação, Componentes leves)
 - Sistemas publish-subscribe (Java JMS)
 - Sistemas de filas de mensagens (Websphere MQ)
 - Serviços Web (Apache Axis)
 - P2P (Overlay, específicos de aplicação)

Modelos fundamentais

- Os modelos fundamentais possibilitam:
 - fazer suposições sobre os sistemas a serem modelados
 - Fazer generalizações e realizar provas
- Modelos
 - Modelo de interação
 - Modelo de falhas
 - Modelo de segurança

Modelo de interação

- Aborda a interação entre processos em um SD
- Duas principais questões
 - Desempenho da comunicação
 - Ausência de um relógio global

Modelo de interação

- Desempenho da comunicação
 - Latência (processo, acesso à rede, transmissão)
 - Taxa de transmissão máxima
 - Flutuação na entrega de pacotes (*jitter*)
- Ausência de relógio
 - Timestamps diferentes
 - Sincronização dos relógios

Modelo de interação

- Sistemas distribuídos síncronos
 - Definição de limites de tempo superiores e inferiores para processamento, atraso das mensagens e desvio do tempo real.
- Sistemas distribuídos assíncronos
 - Não estabelece limites de tempo para processamento, atraso das mensagens e desvio do tempo real.

Modelo de falhas

- Aborda como uma falha pode se manifestar no sistema e quais os seus efeitos e consequências
- Tipos de falhas
 - Falhas por omissão (processo, comunicação)
 - Falhas arbitrárias
 - Falhas de temporização

Modelo de falhas

Classe da falha	Afeta	Descrição
Parada por falha	Processo	O processo pára e permanece parado. Outros processos podem detectar esse estado.
Colapso	Processo	O processo pára e permanece parado. Outros processos podem não detectar esse estado.
Omissão	Canal	Uma mensagem inserida em um <i>buffer</i> de envio nunca chega no <i>buffer</i> de recepção do destinatário.
Omissão de envio	Processo	Um processo conclui um envio, mas a mensagem não é colocada em seu <i>buffer</i> de envio.
Omissão de recepção	Processo	Uma mensagem é colocada no <i>buffer</i> de recepção de um processo, mas esse processo não a recebe efetivamente.
Arbitrária (bizantina)	Processo ou canal	O processo/canal exhibe comportamento arbitrário: ele pode enviar/transmitir mensagens arbitrárias em qualquer momento, cometer omissões; um processo pode parar ou realizar uma ação incorreta.

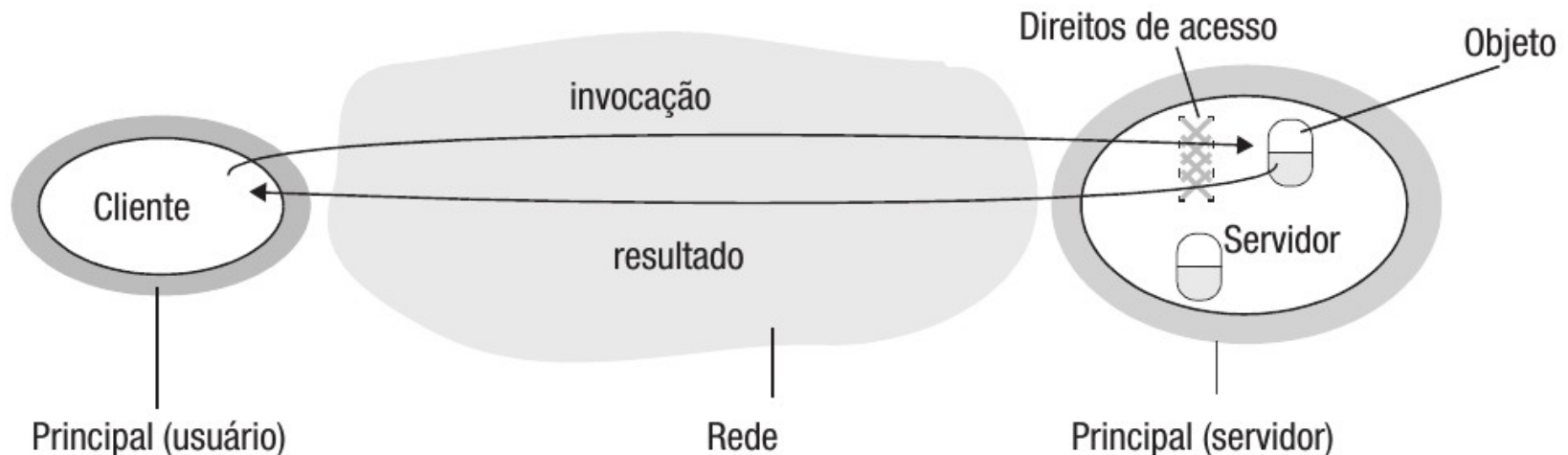
Modelo de falhas

- Comunicação confiável
 - Validade
 - Mensagem enviada é entregue ao destino.
 - Integridade
 - Mensagem recebida é idêntica à enviada e não é entregue mais de uma vez.



Modelo de segurança

- Aborda a proteção dos processos, canais de comunicação e o acesso não autorizado.



Referências

COULOURIS, George F; DOLLIMORE, Jean; KINDBERG, Tim; BLAIR, Gordon. **Sistemas distribuídos: conceitos e projeto**. 5. ed. Porto Alegre: Bookman, 2013.

TANENBAUM, Andrew S.; VAN STEEN, Maarten. **Sistemas distribuídos: princípios e paradigmas**. 2. ed. São Paulo, SP: Pearson Prentice Hall, 2007.