

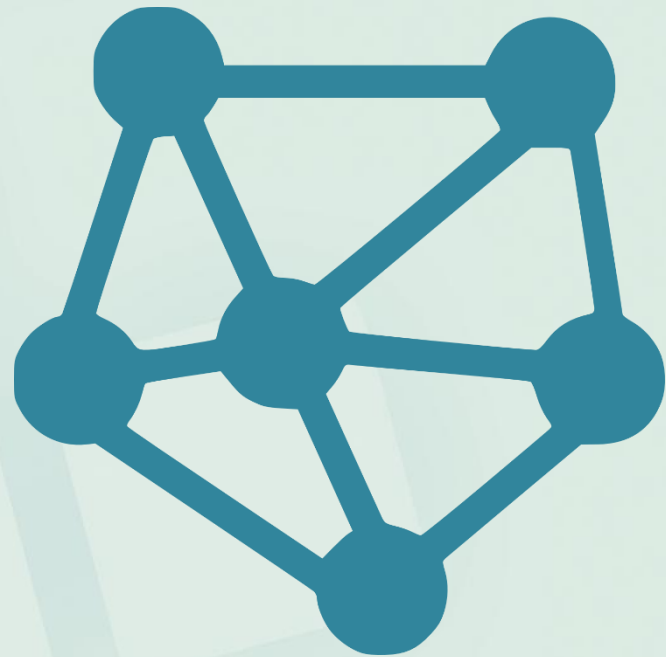


INSTITUTO FEDERAL

Sertão Pernambucano

Sistemas Distribuídos

Modelos de Sistemas Distribuídos



Prof. Heraldo Gonçalves Lima Junior



1. Introdução

1. Introdução

- Mostramos como as propriedades e os problemas de projeto de sistemas distribuídos podem ser capturados e discutidos por meio do uso de **modelos descritivos**. Cada tipo de modelo é destinado a fornecer uma descrição abstrata e simplificada, mas consistente, de um aspecto relevante do projeto de um sistema distribuído.



2. Modelos Físicos

2. Modelos Físicos

- Um modelo físico é uma representação dos elementos de hardware de um sistema distribuído, de maneira a abstrair os detalhes específicos do computador e das tecnologias de rede empregadas.

2.1. Modelo Físico Básico

- Um conjunto extensível de nós de computador interconectados por uma rede de computadores para a necessária passagem de mensagens.
- Além desse modelo básico, podemos identificar três gerações de sistemas distribuídos:

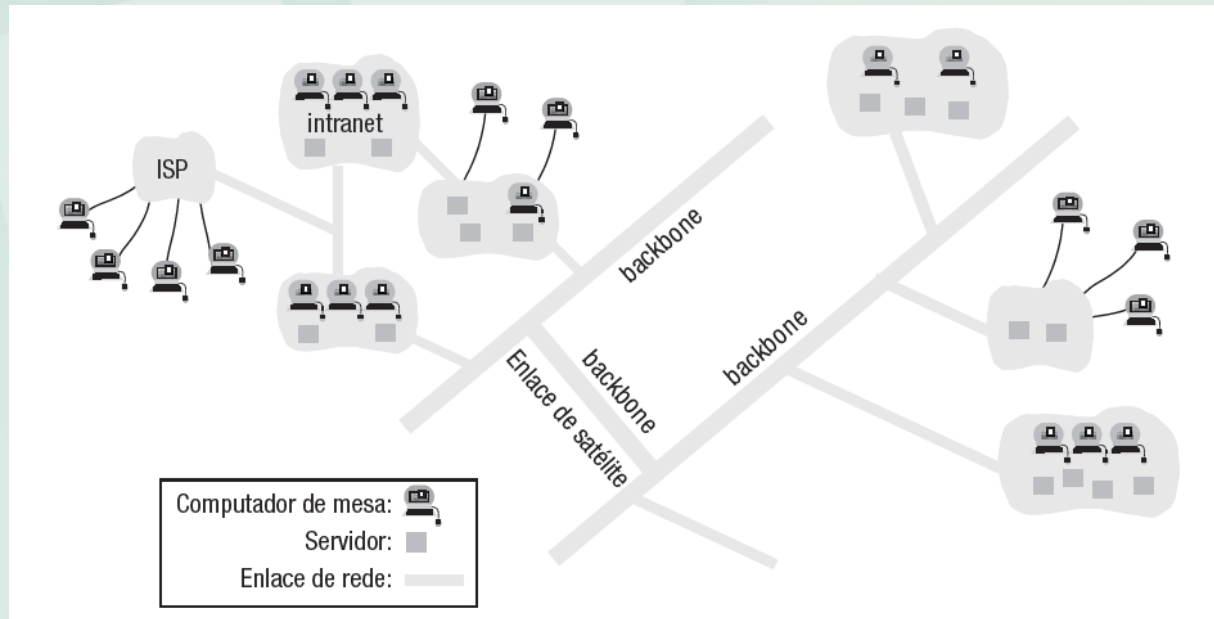
2.2. Sistemas Distribuídos Primitivos

- esses sistemas surgiram no final dos anos 1970 e início dos anos 1980 em resposta ao surgimento da tecnologia de redes locais, normalmente Ethernet.
- Normalmente consistiam em algo entre 10 e 100 nós interconectados por uma rede local, com conectividade de Internet limitada, e suportavam uma pequena variedade de serviços.

2.3. Sistemas Distribuídos Adaptados a Internet

- Aproveitando essa base, sistemas distribuídos de maior escala começaram a surgir nos anos 1990, em resposta ao enorme crescimento da Internet durante essa época.
- Esses sistemas exploravam a infraestrutura oferecida pela Internet para desenvolver sistemas distribuídos realmente globais, envolvendo potencialmente grandes números de nós.

2.3. Sistemas Distribuídos Adaptados a Internet



2.4. Sistemas distribuídos contemporâneos:

- Uma arquitetura física com um aumento significativo no nível de heterogeneidade, compreendendo, por exemplo, os menores equipamentos incorporados utilizados na computação ubíqua, por meio de elementos computacionais complexos encontrados na computação em grade (Grid).

2.4. Sistemas distribuídos contemporâneos:

- Tais sistemas também podem ser de grande escala, explorando a infraestrutura oferecida pela Internet para desenvolver sistemas distribuídos verdadeiramente globais, envolvendo, potencialmente, números de nós que chegam a centenas de milhares.

2.5. Sistemas distribuídos de sistemas

- Um sistema de sistemas pode ser definido como um sistema complexo, consistindo em uma série de subsistemas, os quais são, eles próprios, sistemas que se reúnem para executar uma ou mais tarefas em particular.



<i>Sistemas distribuídos</i>	<i>Primitivos</i>	<i>Adaptados para Internet</i>	<i>Contemporâneos</i>
<i>Escala</i>	Pequenos	Grandes	Ultragrandes
<i>Heterogeneidade</i>	Limitada (normalmente, configurações relativamente homogêneas)	Significativa em termos de plataformas, linguagens e <i>middleware</i>	Maiores dimensões introduzidas, incluindo estilos de arquitetura radicalmente diferentes
<i>Sistemas abertos</i>	Não é prioridade	Prioridade significativa, com introdução de diversos padrões	Grande desafio para a pesquisa, com os padrões existentes ainda incapazes de abranger sistemas complexos
<i>Qualidade de serviço</i>	Em seu início	Prioridade significativa, com introdução de vários serviços	Grande desafio para a pesquisa, com os serviços existentes ainda incapazes de abranger sistemas complexos



3. Modelos de Arquitetura

3. Modelos de Arquitetura

- A arquitetura de um sistema é sua **estrutura** em termos de **componentes especificados separadamente e suas inter-relações**. O objetivo global é **garantir que a estrutura atenda às demandas** atuais e, provavelmente, às futuras demandas impostas sobre ela.
- As maiores preocupações são tornar o sistema **confiável, gerenciável, adaptável e rentável**.

3.1 Elementos arquitetônicos

- **Quais são as entidades** que estão se comunicando no sistema distribuído?
- **Como elas se comunicam** ou, mais especificamente, qual é o paradigma de comunicação utilizado?
- **Quais funções** e responsabilidades (possivelmente variáveis) estão relacionadas a eles na arquitetura global?

3.1 Elementos arquitetônicos

- **Como eles são mapeados** na infraestrutura distribuída física (qual é sua localização)?
- Vamos responder a estas perguntas agora.

3.1.1 Entidades em comunicação

- Do ponto de vista do sistema: **Processos**
- Em alguns ambientes primitivos, como nas redes de sensores, as entidades que se comunicam nesses sistemas são nós.
- Na maioria dos ambientes de sistema distribuído, os processos são complementados por threads;

3.1.1 Entidades em comunicação

- Do ponto de vista da **programação**:
- **Objetos**: os objetos foram introduzidos para permitir e estimular o uso de estratégias orientadas a objeto nos sistemas distribuídos (incluindo o projeto orientado a objeto e as linguagens de programação orientadas a objeto).

3.1.1 Entidades em comunicação

- **Componentes:** Os componentes se parecem com objetos. A principal diferença é que os componentes especificam não apenas suas interfaces (fornecidas), mas também as suposições que fazem em termos de outros componentes/interfaces que devem estar presentes para que o componente cumpra sua função.

3.1.1 Entidades em comunicação

- **Serviços Web:** Eles estão intimamente relacionados aos objetos e aos componentes, novamente adotando uma estratégia baseada no encapsulamento de comportamento e no acesso por meio de interfaces.

3.1.1 Entidades em comunicação

- O W3C (World Wide Web Consortium) define um serviço Web como:
- ... um aplicativo de software identificado por um URI, cujas interfaces e vínculos podem ser definidos, descritos e descobertos como artefatos da XML. Um serviço Web suporta interações diretas com outros agentes de software, usando trocas de mensagens baseadas em XML por meio de protocolos Internet.

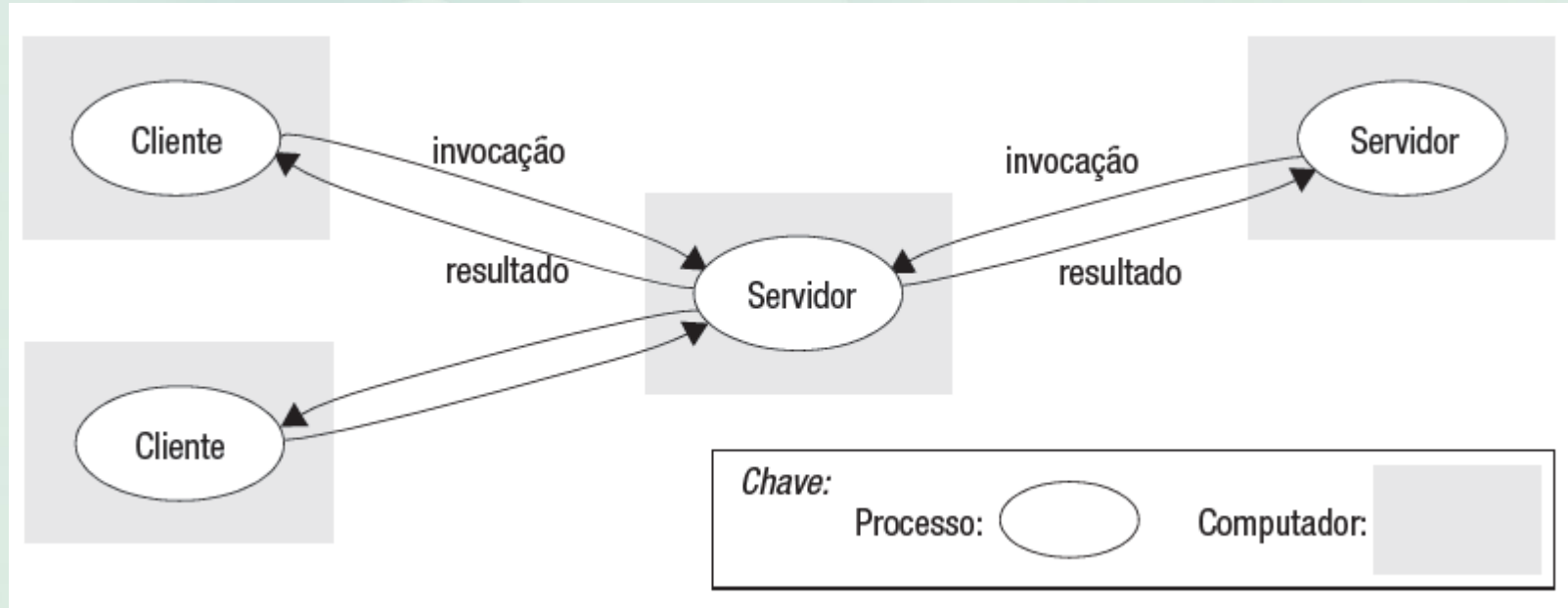
3.1.2. Funções e responsabilidades

- Em um sistema distribuído, os processos, componentes ou serviços, incluindo serviços Web, **interagem uns com os outros para realizar uma atividade útil**; por exemplo, para suportar uma sessão de bate-papo. Ao fazer isso, os processos assumem determinadas funções e, de fato, esse estilo de função é fundamental no estabelecimento da arquitetura global a ser adotada.

3.1.2.1. Cliente-servidor

- Essa é a arquitetura mais citada quando se discute os sistemas distribuídos. Historicamente, ela é a mais importante e continua sendo amplamente empregada.

3.1.2.1. Cliente-servidor

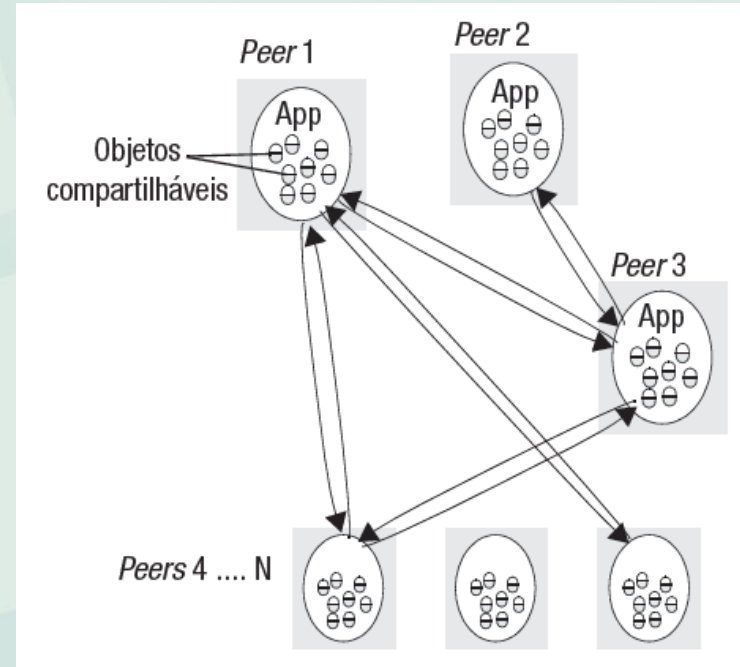


3.1.2.1. Cliente-servidor

- **Os servidores podem, por sua vez, ser clientes de outros servidores**, conforme a figura indica. Por exemplo, um servidor Web é frequentemente um cliente de um servidor de arquivos local que gerencia os arquivos nos quais as páginas Web estão armazenadas.

3.1.2.2. Peer-to-Peer

- Nessa arquitetura, todos os processos envolvidos em uma tarefa ou atividade desempenham funções semelhantes, interagindo cooperativamente como pares (peers).



3.1.2.2. Peer-to-Peer

- Não há distinção entre processos clientes e servidores, nem entre os computadores em que são executados. Em termos práticos, todos os processos participantes executam o mesmo programa e oferecem o mesmo conjunto de interfaces uns para os outros.

3.1.2.2. Peer-to-Peer

- A **centralização** de fornecimento e gerenciamento de serviços, acarretada pela colocação de um serviço em um **único computador**, não favorece um aumento de escala além daquela **limitada pela capacidade do computador** que contém o serviço e da **largura de banda** de suas conexões de rede.

3.1.2.2. Peer-to-Peer

- A principal ideia que levou ao desenvolvimento de sistemas peer-to-peer foi que a rede e os recursos computacionais pertencentes aos usuários de um serviço também poderiam ser utilizados para suportar esse serviço. Isso tem a consequência vantajosa de que **os recursos disponíveis para executar o serviço aumentam com o número de usuários.**

3.1.2.2. Peer-to-Peer

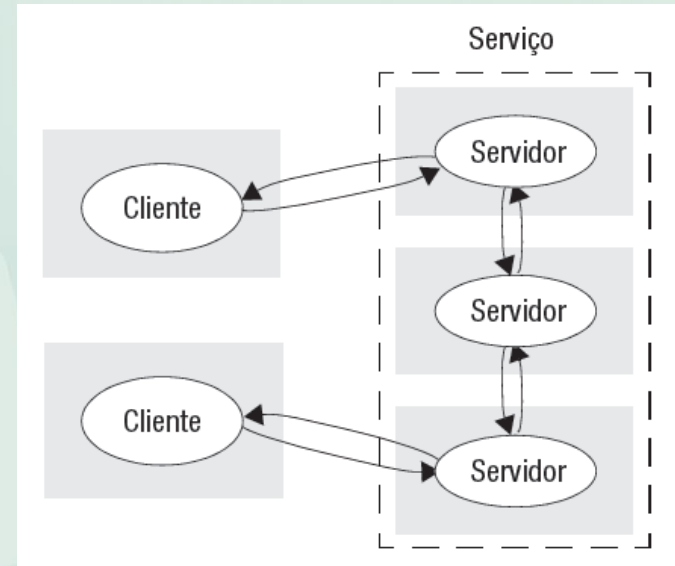


3.1.3. Posicionamento

- O posicionamento é fundamental em termos de determinar as propriedades do sistema distribuído, mais obviamente relacionadas ao desempenho, mas também a outros aspectos, como confiabilidade e segurança.
- A questão de onde colocar determinado cliente ou servidor em termos de máquinas e os processos dentro delas é uma questão de projeto cuidadoso.

3.1.3. Posicionamento

- **Mapeamento de serviços em vários servidores:** os serviços podem ser implementados como vários processos servidores em diferentes computadores, interagindo para fornecer um serviço para processos clientes.



3.1.3. Posicionamento

- A Web oferece um exemplo comum de particionamento de dados no qual cada servidor Web gerencia seu próprio conjunto de recursos. **Um usuário pode usar um navegador para acessar um recurso em qualquer um desses servidores.**

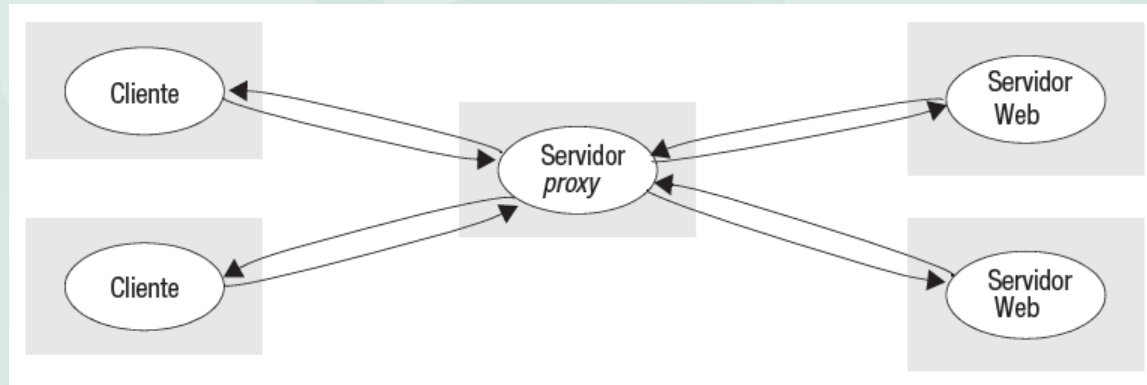


3.1.3. Posicionamento

- **Uso de cache:** uma cache consiste em realizar um **armazenamento de objetos de dados recentemente usados** em um **local mais próximo** a um cliente, ou a um conjunto de clientes em particular, do que a origem real dos objetos em si. Quando um novo objeto é recebido de um servidor, ele é adicionado na cache local, substituindo, se houver necessidade, alguns objetos já existentes.

3.1.3. Posicionamento

- Um servidor proxy Web (Figura 2.5) fornece uma cache compartilhada de recursos Web para máquinas clientes de um ou vários sites.

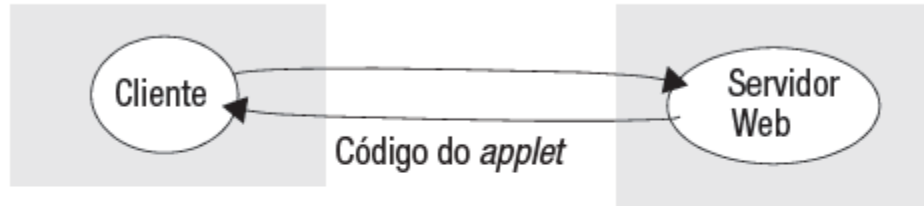


3.1.3. Posicionamento

- **Código Móvel:** Os applets representam um exemplo bem conhecido e bastante utilizado de código móvel – o usuário, executando um navegador, seleciona um link que aponta para um applet, cujo código é armazenado em um servidor Web; o código é carregado no navegador e posteriormente executado.

3.1.3. Posicionamento

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



b) O cliente interage com o *applet*



3.1.3. Posicionamento

- O uso de código móvel é uma ameaça em potencial aos recursos locais do computador de destino. Portanto, os navegadores dão aos applets um acesso limitado a seus recursos locais.

3.1.3. Posicionamento

- **Agentes móveis:** um agente móvel é um programa em execução (inclui código e dados) que **passa de um computador para outro em um ambiente de rede**, realizando uma tarefa em nome de alguém, como uma coleta de informações, e finalmente retornando com os resultados obtidos a esse alguém.

3.1.3. Posicionamento

- Os agentes móveis (assim como o código móvel) **são uma ameaça em potencial à segurança para os recursos** existentes nos computadores que visitam. O ambiente que recebe um agente móvel deve decidir, com base na identidade do usuário em nome de quem o agente está atuando, qual dos recursos locais ele pode usar.

ATIVIDADE

- 1. Descreva e ilustre a arquitetura cliente-servidor de um ou mais aplicativos de Internet importantes (por exemplo, Web, correio eletrônico ou news).

3.2. Padrões Arquitetônicos

- Os padrões arquitetônicos baseiam-se nos elementos de arquitetura mais primitivos discutidos anteriormente e fornecem estruturas recorrentes compostas que mostraram bom funcionamento em determinadas circunstâncias.
- oferecem ideias parciais que, quando combinadas a outros padrões, levam o projetista a uma solução para determinado domínio de problema.

3.2. Padrões Arquitetônicos

- Os padrões arquitetônicos baseiam-se nos elementos de arquitetura mais primitivos discutidos anteriormente e fornecem estruturas recorrentes compostas que mostraram bom funcionamento em determinadas circunstâncias.
- oferecem ideias parciais que, quando combinadas a outros padrões, levam o projetista a uma solução para determinado domínio de problema.

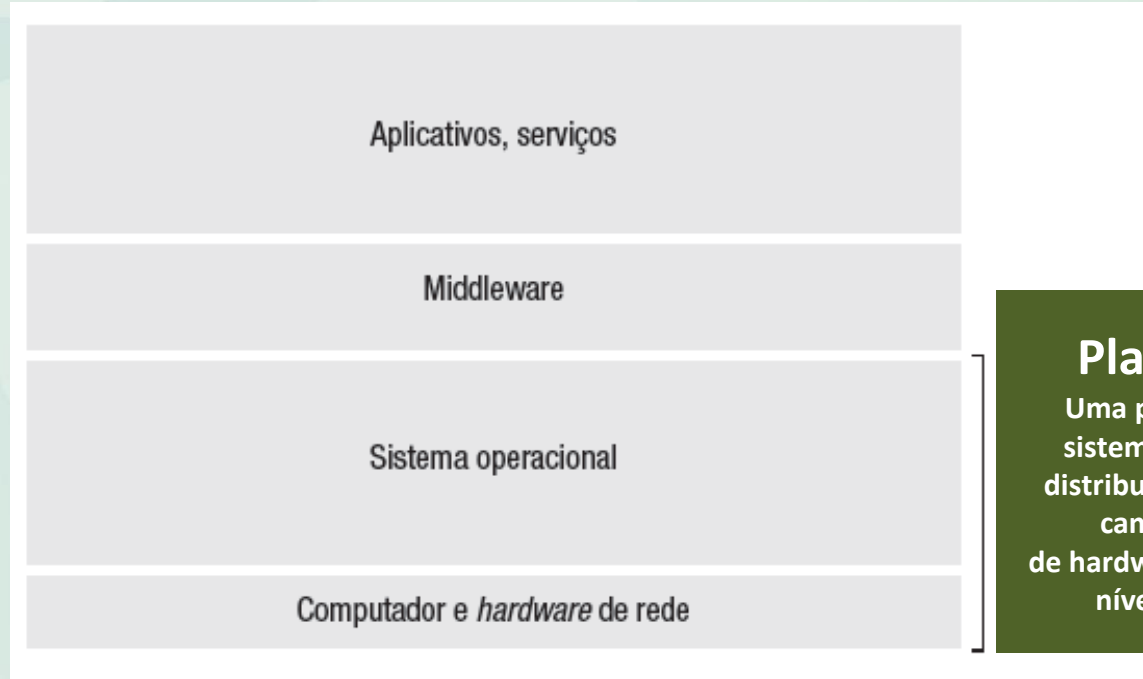
3.2.1. Camadas Lógicas

- O conceito de camadas lógicas é bem conhecido e está intimamente relacionado à abstração. Em uma estratégia de camadas lógicas, **um sistema complexo é particionado em várias camadas, com cada uma utilizando os serviços oferecidos pela camada lógica inferior.**

3.2.1. Camadas Lógicas

- Em termos de sistemas distribuídos, isso se equipara a uma organização vertical de serviços em camadas lógicas. Um serviço distribuído pode ser fornecido por um ou mais processos servidores que interagem entre si e com os processos clientes para manter uma visão coerente dos recursos do serviço em nível de sistema.

3.2.1. Camadas Lógicas



Plataforma:

Uma plataforma para sistemas e aplicativos distribuídos consiste nas camadas lógicas de hardware e software de nível mais baixo.

3.2.1. Camadas Lógicas



3.2.2. Arquitetura de camadas físicas

- Enquanto as camadas lógicas lidam com a organização vertical de serviços em camadas de abstração, as camadas físicas representam **uma técnica para organizar a funcionalidade de determinada camada lógica e colocar essa funcionalidade nos servidores apropriados** e, como uma consideração secundária, nos nós físicos.

3.2.2. Arquitetura de camadas físicas

- Vamos examinar primeiro os conceitos da arquitetura de duas e três camadas físicas. Para ilustrar isso, consideremos a decomposição funcional de determinada aplicação, como segue:
 - A **lógica de apresentação** ligada ao tratamento da interação do usuário e à atualização da visão do aplicativo, conforme apresentada a ele;

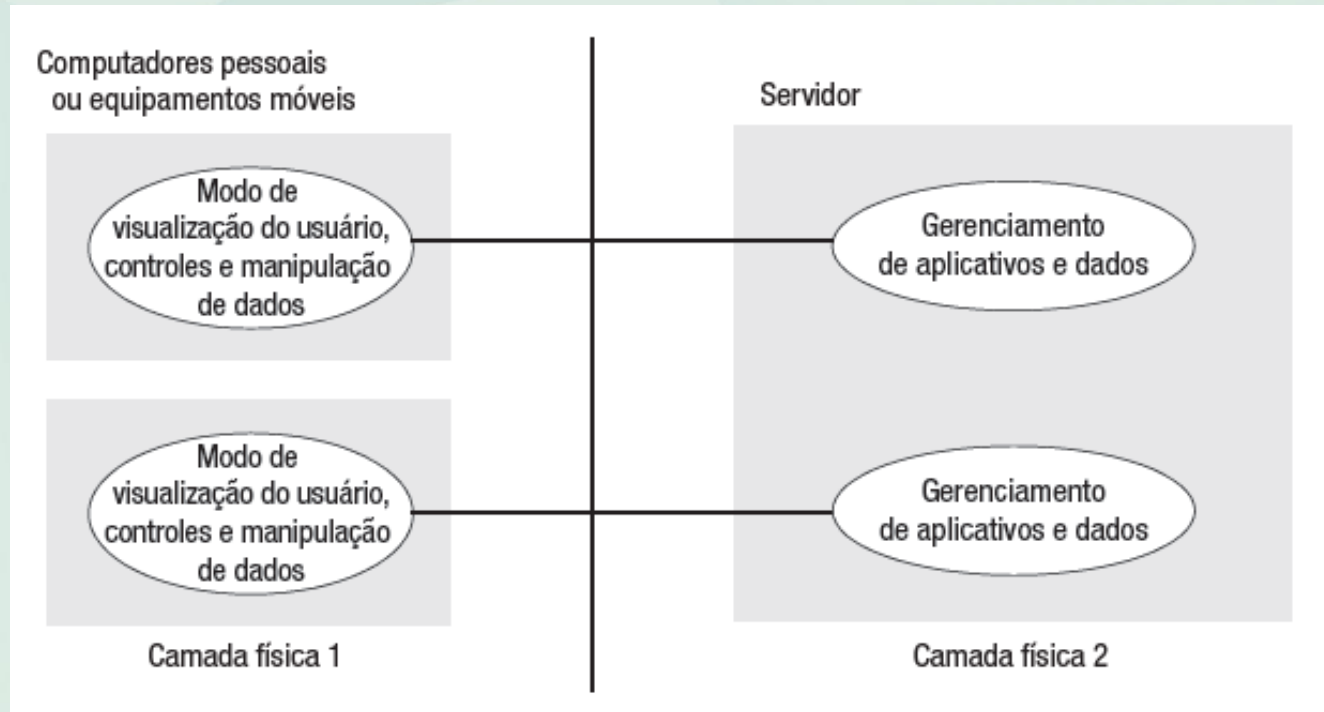
3.2.2. Arquitetura de camadas físicas

- A lógica associada à aplicação ligada ao seu processamento detalhado (também referida como **lógica do negócio**);
- A lógica dos dados ligada ao **armazenamento persistente do aplicativo**, normalmente em um sistema de gerenciamento de banco de dados.

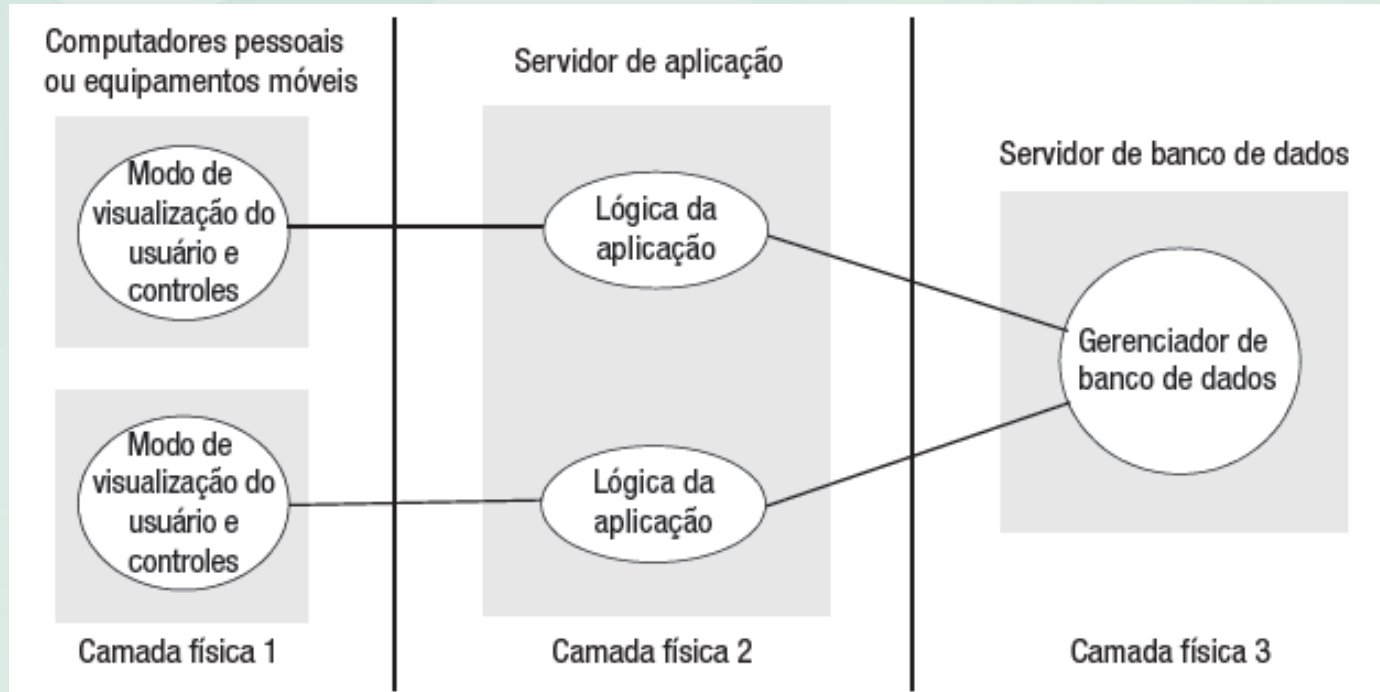
3.2.2. Arquitetura de camadas físicas

- A lógica associada à aplicação ligada ao seu processamento detalhado (também referida como **lógica do negócio**);
- A lógica dos dados ligada ao **armazenamento persistente do aplicativo**, normalmente em um sistema de gerenciamento de banco de dados.

3.2.2. Arquitetura de camadas físicas



3.2.2. Arquitetura de camadas físicas



3.2.2. Arquitetura de camadas físicas



WIKIPÉDIA
A enciclopédia livre

3.2.2. Arquitetura de camadas físicas

- Imagine como acontecem as interação entre front-end e back-end na WEB.
- Toda vez que precisamos atualizar alguma informação da página, toda a página deve ser carregada novamente.

3.2.2. Arquitetura de camadas físicas

- A introdução da Javascript, uma linguagem de programação independente de navegador e de plataforma, e que é baixada e executada no navegador, constituiu um primeiro passo na eliminação dessas restrições.



3.2.2. Arquitetura de camadas físicas

- O AJAX é o segundo passo inovador que foi necessário para permitir o desenvolvimento e a distribuição de importantes aplicativos Web interativos.



3.2.2. Arquitetura de camadas físicas

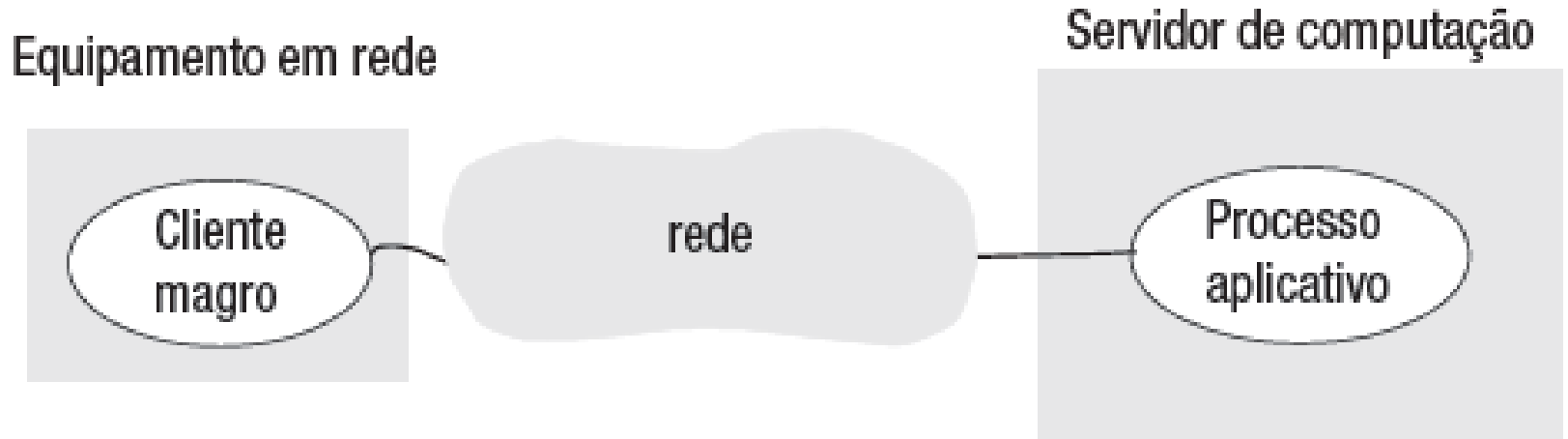
- A tendência da computação distribuída é retirar a complexidade do equipamento do usuário final e passá-la para os serviços da Internet. Isso fica mais aparente na mudança para a computação em nuvem, mas também pode ser visto em arquiteturas de camadas físicas.



3.2.3. Cliente Magro

- O termo se refere a uma camada de software que suporta uma interface baseada em janelas que é local para o usuário, enquanto executa programas aplicativos ou, mais geralmente, acessa serviços em um computador remoto.

3.2.3. Cliente Magro



3.2.3. Cliente Magro



Sistemas Distribuídos – IF SertãoPE Campus Salgueiro

ATIVIDADE

- Frequentemente, os computadores usados nos sistemas peer-to-peer são computadores desktop dos escritórios ou das casas dos usuários. Quais são as implicações disso na disponibilidade e na segurança dos objetos de dados compartilhados que eles contêm e até que ponto qualquer vulnerabilidade pode ser superada por meio da replicação?



4. Modelos Fundamentais

4. Modelos fundamentais

- Genericamente, um modelo fundamental deve conter apenas os **ingredientes essenciais que precisamos considerar para entender e raciocinar a respeito de certos aspectos do comportamento de um sistema.**

4. Modelos fundamentais

- O objetivo de um modelo é:
 - **Tornar explícitas todas as suposições** relevantes sobre os sistemas que estamos modelando.
 - Fazer generalizações a respeito **do que é possível ou impossível**, dadas essas suposições.

4. Modelos fundamentais

- Os aspectos dos sistemas distribuídos que desejamos considerar em nossos modelos fundamentais se destinam a nos ajudar a discutir e raciocinar sobre: **interação, falha e segurança.**

4.1. Modelo de interação

- Em um sistema distribuído, as atividades são realizadas por processos que interagem entre si, porém cada processo tem seu próprio estado, que consiste no conjunto de dados que ele pode acessar e atualizar, incluindo suas variáveis de programa.

4.1. Modelo de interação

- Discutiremos dois fatores que afetam significativamente a interação de processos em um sistema distribuído:
 - o **desempenho** da comunicação, que é, frequentemente, um fator limitante;
 - a impossibilidade de manter uma noção global de **tempo** única.

4.1.1. Modelo de interação: Desempenho da comunicação

- A comunicação em uma rede de computadores tem as seguintes características de desempenho relacionadas à latência, largura de banda e jitter*:
 - A **latência** é o atraso decorrido entre o início da transmissão de uma mensagem em um processo remetente e o início da recepção pelo processo destinatário.

4.1.1. Modelo de interação: Desempenho da comunicação

- A **largura de banda** de uma rede de computadores é o volume total de informações que pode ser transmitido em determinado momento. Quando um grande número de comunicações usa a mesma rede, elas compartilham a largura de banda disponível.

4.1.1. Modelo de interação: Desempenho da comunicação

- **Jitter** é a variação no tempo exigida para distribuir uma série de mensagens. O jitter é crucial para dados multimídia. Por exemplo, se amostras consecutivas de dados de áudio são reproduzidas com diferentes intervalos de tempo, o som resultante será bastante distorcido.

4.1.2. Modelo de interação: Relógios de computador e eventos de temporização

- Cada computador possui seu próprio relógio interno, o qual pode ser usado pelos processos locais para obter o valor atual da hora.



4.1.3. Modelo de interação: Duas variantes do modelo de interação

- **Sistemas distribuídos síncronos:** Os sistemas distribuídos síncronos podem ser construídos, desde que se garanta que os processos sejam executados de forma a respeitar as restrições temporais impostas.

4.1.3. Modelo de interação: Duas variantes do modelo de interação

- **Sistemas distribuídos assíncronos:** O modelo assíncrono não faz nenhuma consideração sobre os intervalos de tempo envolvidos em qualquer tipo de execução.

4.1.4. Modelo de interação: Ordenação de eventos

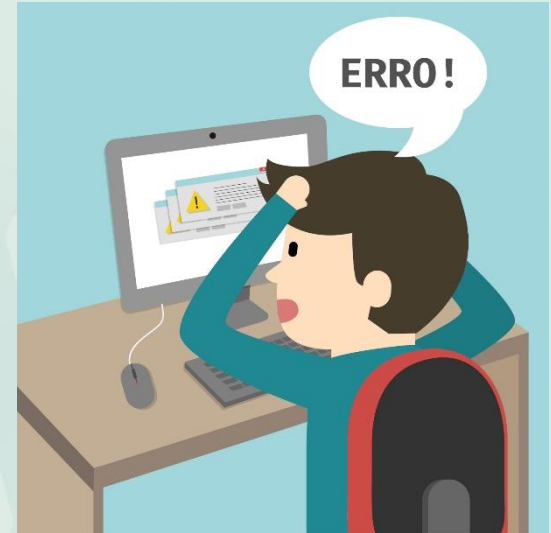
- Em muitos casos, estamos interessados em saber se um evento (envio ou recepção de uma mensagem) ocorreu em um processo **antes, depois ou simultaneamente** com outro evento em outro processo.

4.1.4. Modelo de interação: Ordenação de eventos

Caixa de entrada:		
Item	De	Assunto
23	Z	Re: Reunião
24	X	Reunião
25	Y	Re: Reunião

4.2. Modelo de falhas

- O modelo de falhas define como uma falha pode se manifestar em um sistema, de forma a proporcionar um entendimento dos seus efeitos e consequências.



4.2.1. Modelo de falhas: Falhas por omissão

- As falhas classificadas como falhas por omissão se referem aos casos em que um **processo ou canal de comunicação deixa de executar as ações que deveria.**

4.2.1. Modelo de falhas: Falhas por omissão

- As falhas classificadas como falhas por omissão se referem aos casos em que um **processo ou canal de comunicação deixa de executar as ações que deveria.**
 - **Falhas por omissão de processo:** a principal falha por omissão de um processo é quando ele entra em colapso, parando e não executando outro passo de seu programa.

4.2.1. Modelo de falhas: Falhas por omissão

- **Falhas por omissão na comunicação:** a principal falha por omissão de um processo é quando ele entra em colapso, parando e não executando outro passo de seu programa.

4.2.1. Modelo de falhas: Falhas por omissão



4.2.1. Modelo de falhas: Falhas por omissão

- **Falhas por omissão de envio:** perda de mensagens entre o processo remetente e o buffer de envio;
- **Falhas por omissão de recepção:** perda de mensagens entre o buffer de recepção e o processo destino;
- **Falhas por omissão de canal:** perda de mensagens no meio de comunicação.

4.2.2. Modelo de falhas: Falhas arbitrárias

- Uma falha arbitrária de um processo é aquela em que ele omite arbitrariamente passos desejados do processamento ou efetua processamento indesejado. Portanto, as falhas arbitrárias não podem ser detectadas verificando-se se o processo responde às invocações, pois ele poderia omitir arbitrariamente a resposta.

4.2.3. Modelo de falhas: Falhas de temporização

- As falhas de temporização são aplicáveis aos sistemas distribuídos síncronos em que limites são estabelecidos para o tempo de execução do processo, para o tempo de entrega de mensagens e para a taxa de desvio do relógio.

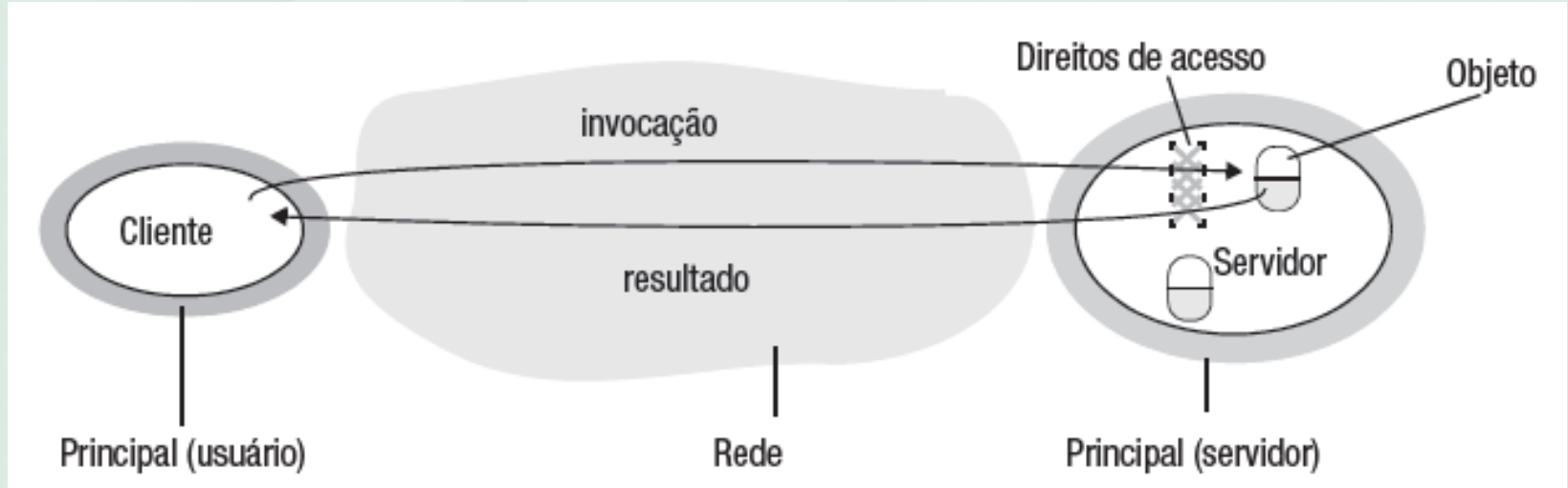
4.2.4. Modelo de falhas: Mascaramento de falhas

- Um serviço mascara uma falha ocultando-a completamente ou convertendo-a em um tipo de falha mais aceitável.
- O conhecimento das características da falha de um componente pode permitir que um novo serviço seja projetado de forma a mascarar a falha dos componentes dos quais ele depende.

4.3. Modelo de segurança

- A segurança de um sistema distribuído pode ser obtida tornando seguros os processos e os canais usados por suas interações e protegendo contra acesso não autorizado os objetos que encapsulam.

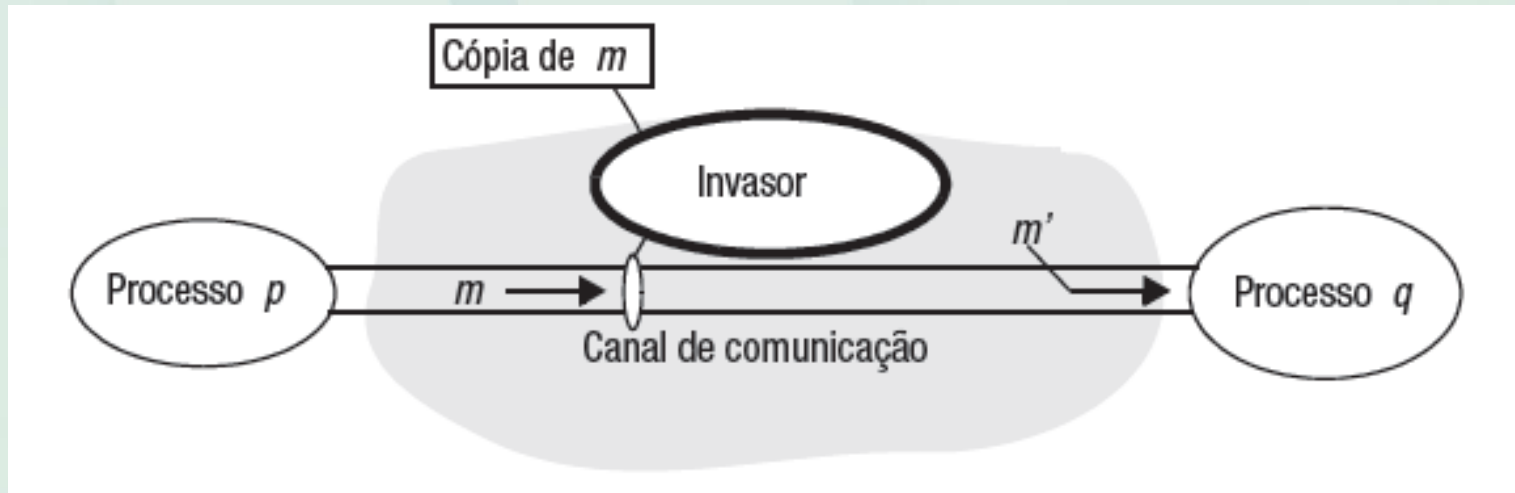
4.3.1. Proteção de objetos



4.3.2. Tornando processos e suas interações seguros

- Os processos interagem enviando mensagens. As mensagens ficam expostas a ataques, porque o acesso à rede e ao serviço de comunicação é livre para permitir que quaisquer dois processos interajam.

4.3.2. Tornando processos e suas interações seguros



4.3.2. Tornando processos e suas interações seguros

- **Ameaças aos processos:** um processo projetado para tratar pedidos pode receber uma mensagem de qualquer outro processo no sistema distribuído e não ser capaz de determinar com certeza a identidade do remetente.

4.3.2. Tornando processos e suas interações seguros

- **Ameaças aos canais de comunicação:** um invasor pode copiar, alterar ou injetar mensagens quando elas trafegam pela rede e em seus sistemas intermediários (roteadores, por exemplo). Tais ataques representam uma ameaça à privacidade e à integridade das informações quando elas trafegam pela rede e à própria integridade do sistema.

4.3.2. Tornando processos e suas interações seguros

- **Criptografia e segredos compartilhados:**
 - Criptografia é a ciência de manter as mensagens seguras, e cifragem é o Processo de embaralhar uma mensagem de maneira a ocultar seu conteúdo.

4.3.2. Tornando processos e suas interações seguros

- **Autenticação:**
 - o uso da criptografia fornece a base para a autenticação de mensagens – provar as identidades de seus remetentes. A técnica de autenticação básica é incluir em uma mensagem uma parte cifrada que possua conteúdo suficiente para garantir sua autenticidade.

4.3.2. Tornando processos e suas interações seguros

- **Canais seguros:**
 - criptografia e autenticação são usadas para construir canais seguros como uma camada de serviço a mais sobre os serviços de comunicação já existentes. Um canal seguro é um canal de comunicação conectando dois processos, cada um atuando em nome de um principal.

ATIVIDADE

- Defina a propriedade de integridade da comunicação confiável e liste todas as possíveis ameaças à integridade de usuários e de componentes do sistema. Quais medidas podem ser tomadas para garantir a propriedade de integridade diante de cada uma dessas fontes de ameaças?

Obrigado!
Vlw! Flw!



INSTITUTO FEDERAL
Sertão Pernambucano