

Sistemas Distribuídos

**Aplicações de sistemas
distribuídos e segurança**

Profa. Ms. Adriane Ap. Loper

- Unidade de Ensino: 4
- Competência da Unidade: Sistemas distribuídos
- Resumo: Princípios de segurança da informação para sistemas distribuídos.
- Palavras-chave : Segurança, RPC, Java, sockets, RMI.
- Título da Teleaula: Aplicações de sistemas distribuídos e segurança
- Teleaula nº: 4

Contextualizando

Você já parou para pensar quais as principais ameaças que os sistemas distribuídos podem sofrer? Será que existem certos tipos de ataques que podem afetar nossos sistemas distribuídos? Qual a melhor forma de se proteger e prevenir esses tipos de ataques?

Você se comunica com os amigo(a)s, mas você já teve alguma vez a curiosidade de pensar como essa comunicação funciona internamente em termos de programação?

VAMOS APRENDER?



Fonte: <https://bit.ly/2WbZ2mn>.

Contextualizando

Você concluiu a sua graduação na área de TI e agora pretende abrir sua própria empresa, desenvolvendo e comercializando um sistema CRM (*Customer Relationship Management*), que é a Gestão de Relacionamento com o Cliente, que possuirá várias funcionalidades inovadoras, mas também deve possuir as mesmas funcionalidades comumente presentes em sistemas de empresas concorrentes.



Fonte: <https://bit.ly/2L8DZjm>.

Contexto

Como você pretende que sua aplicação tenha compatibilidade com sistemas operacionais comumente encontrados nos laptops dos vendedores das empresas de seus futuros clientes, você decide utilizar a linguagem Java para desenvolver essa aplicação.

Você será capaz de atingir todos os objetivos conforme a necessidade de sua empresa?

Segurança em Sistemas Distribuídos

Contexto



Fonte: <https://bit.ly/2J7mkGl>.

Como o jovem empreendedor que você é, deve saber que um sistema CRM é uma aplicação utilizada sempre em ambientes corporativos, portanto, é crucial que você conheça e saiba proteger o seu sistema contra vulnerabilidades de segurança em sistemas distribuídos. Você **listará** quais são os tipos de ataques que devem ser prevenidos no desenvolvimento de um sistema CRM. Será que você consegue?

Segurança da Informação

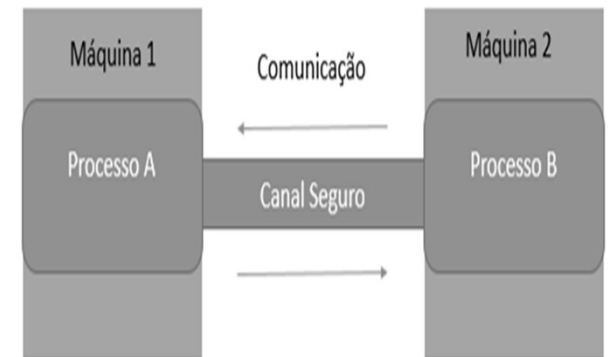
Propriedades fundamentais da informação que precisamos proteger:



Fonte: <https://bit.ly/2UPBXV7>.

Segurança da Informação

O grande objetivo de nossa aplicação distribuída é que ela possa se comunicar, ou seja, trocar dados com confiabilidade, integridade e autenticidade. Portanto, uma das principais formas de proteção é deixar a comunicação permitida apenas em máquinas e por usuários autenticados em nosso sistema e com as permissões necessárias.



Fonte: livro texto pg.146.

Ameaças

As ameaças aos sistemas distribuídos podem ser divididas nas seguintes classes (COULOURIS et al., 2013):

- ✓ *Leakage* (vazamento): acesso à informação por agentes não autorizados;
- ✓ *Tampering* (falsificação): modificação não autorizada de uma informação;
- ✓ *Vandalism* (vandalismo): interferência no funcionamento de um sistema sem ganhos para o criminoso.

Ameaças

Para produzir um sistema que seja seguro contra diversas ameaças, precisamos aprender classificar essas ameaças e entender seus os métodos de ataque. Quando nos referimos a : Interferência no funcionamento de um sistema sem ganhos para o criminoso. Estamos nos referindo à:

- a) Vazamento
- b) Falsificação
- c) Vandalismo
- d) DDos
- e) Pescaria

Obtenção de acesso

Obtenção de acesso

As estratégias de violações de segurança em sistemas distribuídos dependem da obtenção de acesso aos canais de comunicação de nosso sistema ou do estabelecimento de canais que escondem conexões, com a autoridade desejada. Fazem parte destas estratégias:

Eavesdropping: acesso a cópias de mensagem sem autorização. Geralmente essa estratégia funciona através da captura de mensagens da rede.

Ex.: usando a internet, um computador pode se

Obtenção de acesso

passar por outro, quando configurado com o endereço de rede de outro. Desta forma, ele pode receber as mensagens endereçadas a outro destinatário.

Masquerading (disfarce).

Message tampering (falsificação de mensagem).

Replaying.

Acesso ao sistema

É utilizado um método simples de infiltração que pode ser o uso de programas de quebra de senhas para obter as chaves de acesso de algum usuário do sistema.

- ✓ Vírus;
- ✓ *Worm*;
- ✓ Cavalo de Tróia (*Trojan Horse*).
- ✓ *Spyware*.
- ✓ *Keyloggers*.
- ✓ *Backdoor* (Porta dos Fundos).

Acesso ao sistema

É utilizado um método simples de infiltração que pode ser o uso de programas de quebra de senhas para obter as chaves de acesso de algum usuário do sistema.

- ✓ *Spam.*
- ✓ *Adware.*
- ✓ *Exploits.*
- ✓ *Hoax.*
- ✓ *Phishing.*

Ameaças

Cat. de ataque	Definição	Prevenção
Vazamento	Aquisição de informação por pessoal não autorizado.	<ul style="list-style-type: none"> • Conscientização dos colaboradores. • Investimento em <i>appliances</i> (hardware + software dedicados) de segurança. • Uso de autenticação baseada em chave privada, não apenas para usuários, mas também para máquinas e serviços. • Assinaturas digitais e chaves de sessão.

Cat. de ataque	Definição	Prevenção
Falsificação	Aquisição de informação por pessoal não autorizado.	<ul style="list-style-type: none"> • Investimento em <i>appliances</i> (hardware + software dedicados) de segurança. • Uso de autenticação baseada em chave privada, não apenas para usuários, mas também para máquinas e serviços. • Assinaturas digitais e chaves de sessão.

Resolvendo a Situação-Problema 1

Cat. de ataque	Definição	Prevenção
Vandalismo	Interferência na operação de um sistema sem obtenção de lucro pelo invasor.	<ul style="list-style-type: none">• Investimento em <i>appliances</i> (hardware + software dedicados) de segurança.• Uso de autenticação baseada em chave privada, não apenas para usuários, mas também para máquinas e serviços.• Assinaturas digitais e chaves de sessão.

**Compreenderam a
complexidade da
segurança em
sistemas
distribuídos?**

TCP/UDP

Contexto



Você consegue criar um serviço de bate-papo (chat) para o sistema CRM que a empresa está desenvolvendo?

Crie uma aplicação CLI em Java que utiliza *sockets*, relacionando com os conceitos de aplicação Cliente e Servidor, de forma que o que for digitado no console (terminal) de uma instância servidora da sua aplicação seja replicada – ou seja, apareça na tela – de uma instância cliente da sua aplicação e vice-versa.

Fonte: <https://bit.ly/2J7mkGI>.

Elementos de uma comunicação em rede

7 camadas ISO/OSI;

A camada de transporte (4 camada) pode usar 2 protocolos: TCP e UDP.

TCP (*Transmission Control Protocol*) ou
Protocolo de Controle de Transmissão

É um protocolo utilizado como base para comunicação entre máquinas, onde, caso haja alguma perda de informação, durante a transmissão, aquela informação seja retransmitida (MAIA, 2013);

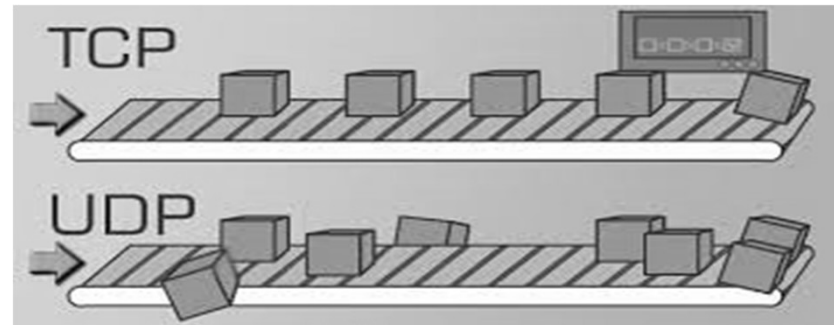
O TCP é um protocolo orientado à conexão.

Ex.: transferência em internet banking.

UDP

UDP (*User Datagram Protocol*)

É um protocolo utilizado como base para comunicação entre máquinas nas quais é importante que o atraso entre o envio e o recebimento da mensagem seja minimizado. Ex.: circuito fechado de TV (CFTV).



Fonte: <https://bit.ly/2ZHDnVd>.

Comunicação através de *sockets*

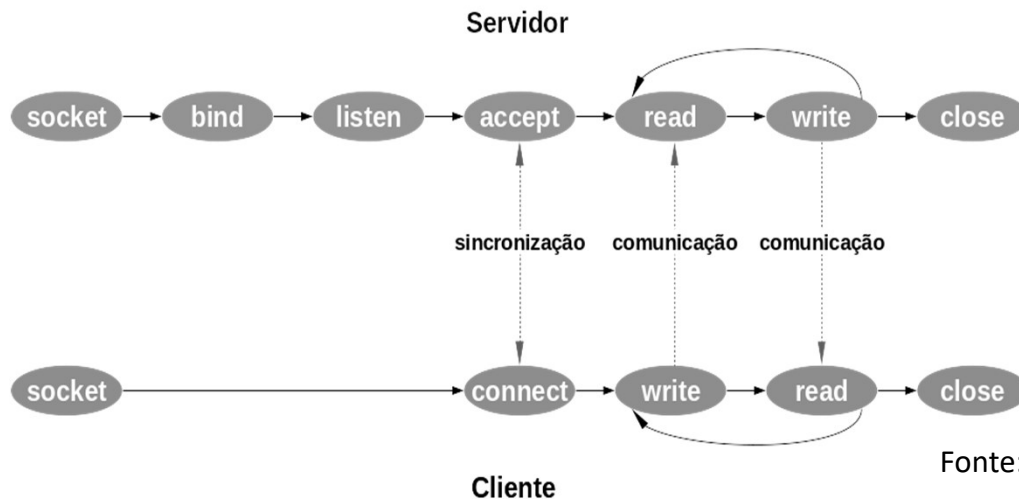
- É uma das tecnologias/mecanismos mais comuns para comunicação entre máquinas;
- Um *socket* é uma combinação de endereço IP e porta de comunicação;
- Conforme observa Coulouris et al. (2013), para um processo de enviar e receber mensagens, seu *socket* precisa estar atrelado a uma porta e a um endereço IP roteável na máquina onde esse processo está sendo executado;

Sockets

- Os *sockets* utilizam o TCP (ou UDP) para realizar a comunicação entre aplicações que estejam sendo executadas em um sistema operacional, razão pela qual essa comunicação é chamada de **interprocessos**;
- Aplicações que estejam sendo executadas em um sistema operacional são chamadas de **processos**;
- Para que essa abstração possa ocorrer, existem funcionalidades (por vezes chamadas de **primitivas**) que normalmente são fornecidas por qualquer implementação de *socket*;

Funcionalidades

Em qualquer linguagem de programação orientada a objetos, essas funcionalidades representam métodos, já implementados, em determinadas classes relativas à comunicação via rede.



Fonte: livro texto pg. 160.

Chat

A proposta é desenvolver uma aplicação CLI (modo texto) em Java que utilize *sockets*, relacionando os conceitos de aplicação Cliente e Servidor, de forma que o que for digitado no console (terminal) de uma instância servidora da sua aplicação seja replicada – ou seja, apareça na tela – de uma instância cliente da sua aplicação; e vice-versa, nesta ordem.

A ideia aqui é que você possa aplicar os conceitos aprendidos de forma que tenha autonomia para modificar o código de para atender aos requisitos apresentados.

Como, primeiramente, o servidor envia uma mensagem para o cliente, para que este último responda, o que deve ser feito é simplesmente inverter a ordem de escrita/leitura de ambas aplicações.

**Máquinas
Cliente/Servidor**

Contexto

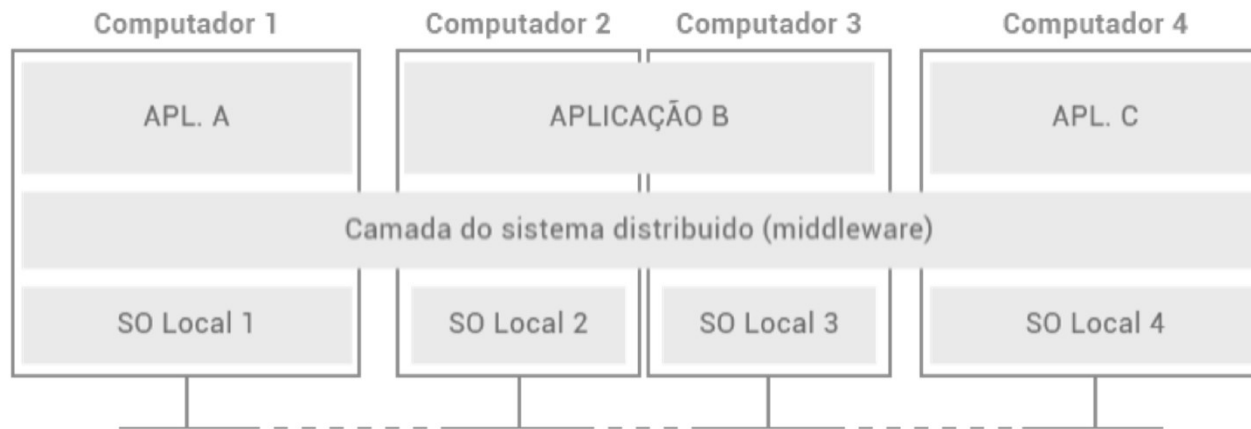


Você deve criar uma funcionalidade inovadora em seu sistema de CRM na qual, caso os laptops dos vendedores dependam de recursos que estejam disponíveis apenas em algum servidor da empresa, utilize o RPC via Java RMI para fazer uma consulta em uma lista de nomes, disponíveis em uma máquina que terá o papel de máquina servidora e...

...com esse arquivo de texto, no formato .txt, e utilize ao menos três outras máquinas, que terão o papel de máquinas cliente (que estejam conectadas em rede com o servidor), que farão uma chamada a um método remota, disponível apenas no servidor, de maneira que o objeto Java, que possibilita o uso desse método no código da máquina servidora, possa ser utilizado localmente nas máquinas clientes.

Fonte: <https://bit.ly/2J7mkG1>

Middleware



Fonte: livro texto pg. 170

A camada de middleware está interligando aplicações e sistemas operacionais que estão sendo executados em diferentes computadores; É um framework – RPC.

Comunicação via RPC

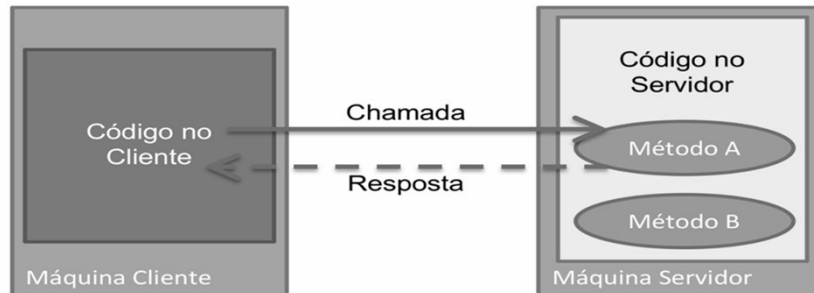
Envolve as sete camadas do modelo de referência ISO/OSI;

É uma forma mais granular de comunicação entre máquinas, uma vez que, diferentemente da comunicação via *sockets*, na qual executamos toda a aplicação, através do uso de RPC, podemos executar apenas um (ou mais) métodos de interesse, implementados em uma máquina, através de outra máquina.

RPC

Modelo de Comunicação	Implementado através de:
RPC <i>(Remote Procedure Call)</i>	Linguagens de programação estruturadas.
RMI <i>(Remote Method Invocation)</i>	Linguagens de programação orientadas a objeto.
MOM <i>(Message Oriented Middleware)</i>	Linguagens de programação para web.

Modelos de comunicação entre máquinas



Fonte: livro texto pg. 172.

Exemplo de código “Interface.java”

```
1 import java.rmi.Remote;  
2  
3 public interface Interface extends Remote {  
4  
5     public String converterParaMaiuscula(String palavra) throws Exception;  
6  
7 }
```

- Esse método remoto deve possuir a seguinte assinatura: `public String verificarNomes(String nome)` e deve apresentar, no console (terminal) de cada máquina cliente, a quantidade de vezes que o nome informado aparece na lista.
- Código da classe “Interface.java”

```
1 import java.rmi.Remote;  
2  
3 public interface Interface extends Remote {  
4  
5     public String verificarNomes(String nome) throws Exception;  
6  
7 }
```

- Código da classe “Servidor.java”;
- Código da classe “Cliente.java”

```
1 import java.rmi.Naming;
2
3 public class Cliente {
4
5     public static void main(String[] args) throws Exception {
6
7         Interface objetoRemoto = (Interface) Naming.lookup("//10.116.201.48/obj");
8
9         System.out.println(objetoRemoto.verificarNomes("Souza"));
10    }
11 }
```


**Entenderam que
em S.D são
possíveis diversas
implementações?**

Recapitulando

- ✓ Segurança em Sistemas Distribuídos;
- ✓ Utilizando Sockets com Java;
- ✓ Utilizando RPC com Java.



Fonte: Shutterstock

