

# **INTRODUÇÃO AOS SISTEMAS DISTRIBUÍDOS**

Prof. Me. Pietro Martins de Oliveira

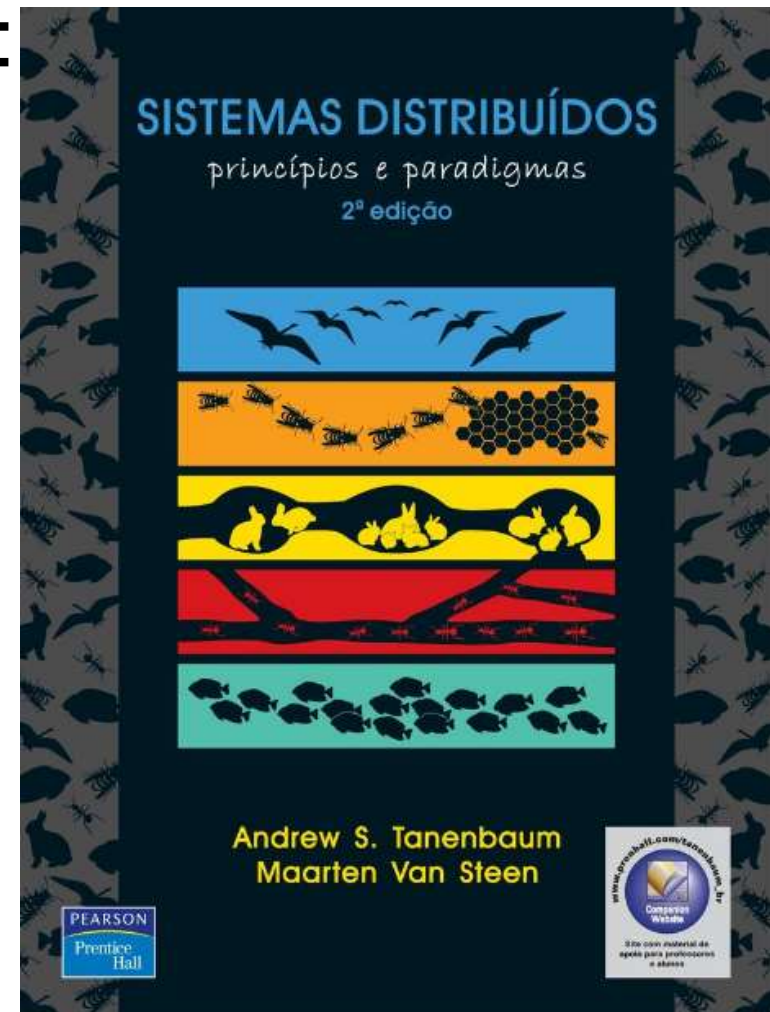
- De acordo com Andrew S. Tanenbaum (2007):

“Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente.”
- Outras definições (COULOURIS et. al., 2011) nos dizem que:

“Sistemas distribuídos são aqueles nos quais **componentes** são **alocados** em **diferentes redes** de computadores que, por sua vez, se **comunicam** e **coordenam** suas **ações** trocando mensagens entre si.”

- **Arquitetura de Sistemas Distribuídos:** forma como organiza-se a rede de computadores que forma um sistema (software & hardware).
- **Modelos de arquitetura:**
  - Cliente-servidor
  - Peer-to-peer (Ponto-a-ponto ou P2P)
  - Híbrido
- **Aplicações:**
  - Instant Messaging; Aplicações WEB; Aplicações Bancárias; Serviços de Streaming; Torrents; Games; Criptocurrencies; etc.

- Para conhecer melhor o poder e os detalhes dos sistemas distribuídos:
  - TANENBAUM, Andrew S.; STEEN, Maarten Van.  
**Sistemas distribuídos: princípios e paradigmas.**  
2. ed. São Paulo: Pearson Prentice Hall, 2009. 402 p.  
ISBN 978-85-7605-142-8



- COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Sistemas Distribuídos - Conceitos e Projeto**. 4. Ed. Porto Alegre: Bookman, 2007.
- TANENBAUM, Andrew S.; STEEN, Maarten Van. **Sistemas distribuídos: princípios e paradigmas**. 2. ed. São Paulo: Pearson Prentice Hall, 2009.

# INTRODUÇÃO AOS SISTEMAS DISTRIBUÍDOS

Prof. Me. Pietro Martins de Oliveira

# SOCKETS

Prof. Me. Pietro Martins de Oliveira

- **Network Socket (Soquete de rede):**

- Forma de conectar um processo a outro de maneira remota (ou não).
- É um *endpoint*.

- Origem primária ou destino final.



Fonte: pixabay.com

- **APIs e Bibliotecas**

- Existem bibliotecas para a ampla maioria das linguagens de programação:
  - C, C++, Java, Python, etc.
- Disponíveis para a maioria das plataformas:
  - Microsoft, Linux, Apple, etc.



- **Protocolos:**

- IP

- TCP: confiável
    - UDP: não-confiável

- **Endereço** (IP) + **Porta** (TCP ou UDP)

- Para cada um dos nós da rede

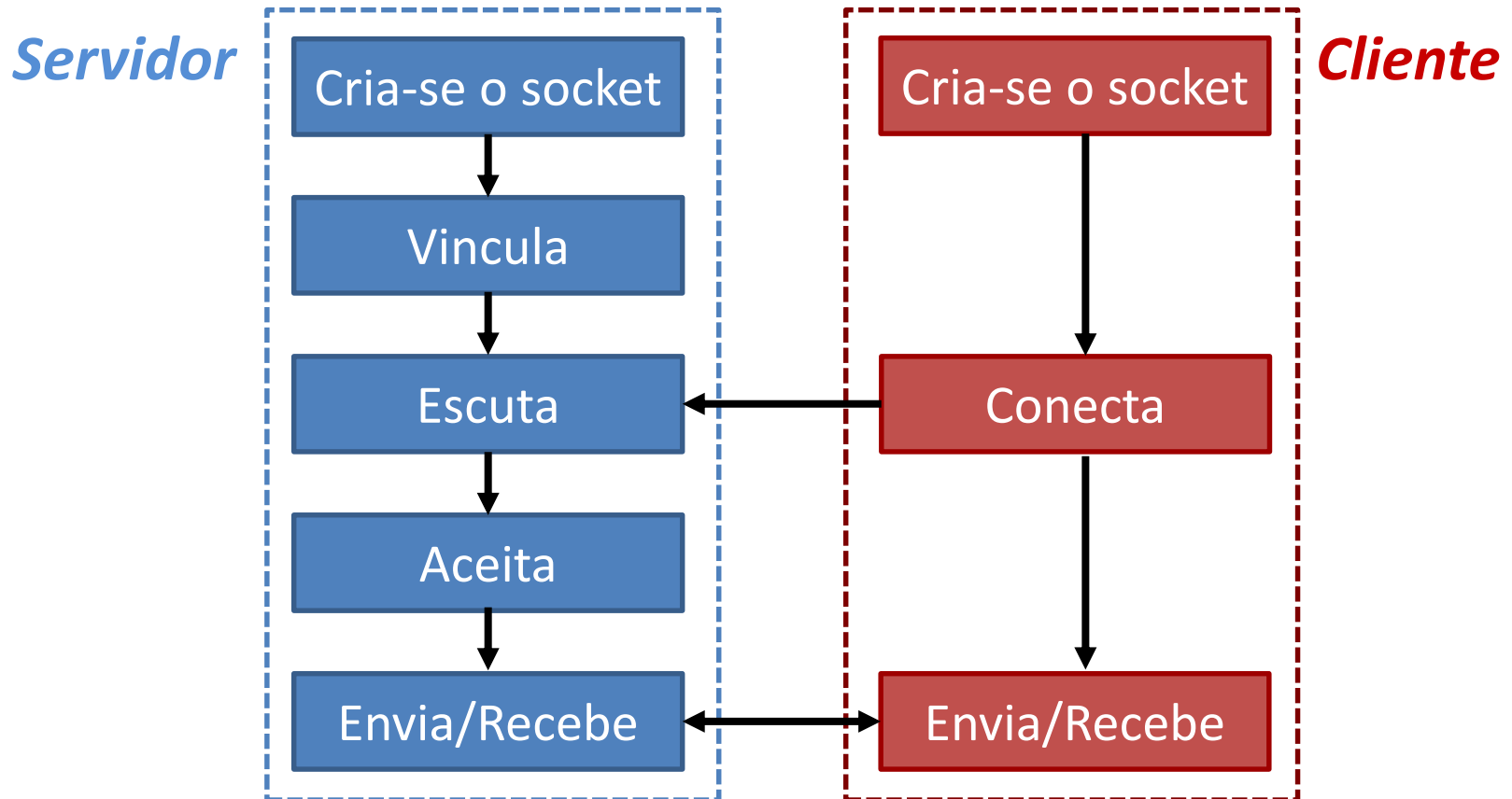
- Seja P2P ou Cliente-servidor

- Exemplo: 192.168.0.1:8080

- ***Importante***: dependendo da linguagem selecionada e/ou plataforma de aplicação, existirão detalhes específicos (APIs).

- **Manipulando Sockets**
  - Plataforma Windows
  - Linguagem C
  - IDE: Dev C++
  - API: WinSock (DLL do Windows)
- **Importante:**
  - É preciso adicionar as bibliotecas **libwsck32.a** (ou **libws2\_32.a**) ao *linker*,
  - Pode ser preciso alterar as opções do *linker*, adicionando o parâmetro **-lws2\_32**

- **Esquema geral – Cliente-servidor:**



## ***Windows:***

```
//Inicia a DLL WinSock em um processo
int WSAStartup(
    WORD wVersionRequested, //versão do
    WinSock
    LPWSADATA lpWSADATA //Configs.
);

//Captura o último erro ocorrido
int WSAGetLastError();

//Finaliza a DLL WinSock em um processo
int WSACleanup();
```

## ***Servidor:***

```
//Criação do socket
int socket(int af, //Domínio (IPv4, IPv6...)
           int type, //TCP ou UDP (SOCK_STREAM
           ou SOCK_DGRAM)
           int protocol); //IP: 0, ICMP: 1

//Faz a vinculação do socket ao respectivo
endereço IP e porta
int bind(int sockfd, //ref. ao socket
         const struct sockaddr *addr,
         //endereço
         socklen_t addrlen); //tamanho do
end., em bytes
```

## ***Servidor (continuação):***

```
int listen(int sockfd, //ref. ao socket
           int backlog); //fila de conexões
                        pendentes

int accept(int sockfd, //ref. ao socket
           struct sockaddr *addr, //endereço
           socklen_t *addrlen); //tamanho do
                        end., em bytes
```

## ***Cliente:***

```
int connect(int sockfd, //ref. ao socket
            const struct sockaddr *addr,
            socklen_t addrlen);
```

***Nó qualquer (cliente ou servidor):***

```
int send(int sockfd, //ref. ao socket
        const char* buf, //Buffer: dados
        recebidos
        int len, //Tamanho do buffer
        int flags); //Modos de envio
```

```
int recv(int sockfd, //ref. ao socket
        char* buf, //Buffer: dados recebidos
        int len, //Tamanho do buffer
        int flags); //Modos de recebimento
```

# SOCKETS

Prof. Me. Pietro Martins de Oliveira



# **INTRODUÇÃO AOS SISTEMAS DISTRIBUÍDOS**

**Prof. Me. Pietro Martins de Oliveira**