

# Desenvolvimento Web

Semana 2

---

Eduardo Mangeli

05 de Março de 2023

1. Protocolos

Arquitecturas

# Protocolos

---

# O que é um protocolo?

## Protocolos humanos:

- "Que horas são?"
- "Tenho uma dúvida"
- apresentações

# O que é um protocolo?

## Protocolos humanos:

- "Que horas são?"
- "Tenho uma dúvida"
- apresentações

... mensagens específicas são enviadas

... ações específicas são realizadas  
quando se recebe uma mensagem, ou  
ocorre outro evento.

# O que é um protocolo?

## Protocolos humanos:

- "Que horas são?"
- "Tenho uma dúvida"
- apresentações

... mensagens específicas são enviadas

... ações específicas são realizadas  
quando se recebe uma mensagem, ou  
ocorre outro evento.

## Protocolos de rede

- máquinas no lugar de gente
- toda a comunicação na internet é regida por protocolos

# O que é um protocolo?

## Protocolos humanos:

- "Que horas são?"
- "Tenho uma dúvida"
- apresentações

... mensagens específicas são enviadas

... ações específicas são realizadas quando se recebe uma mensagem, ou ocorre outro evento.

## Protocolos de rede

- máquinas no lugar de gente
- toda a comunicação na internet é regida por protocolos

### Formalmente

Protocolos definem o formato, a ordem das mensagens enviadas e recebidas entre os membros de uma rede, além das ações realizadas na transmissão e recepção de mensagens.

# O que é um protocolo?

## Protocolos humanos:

- "Que horas são?"
- "Tenho uma dúvida"
- apresentações

... mensagens específicas são enviadas

... ações específicas são realizadas quando se recebe uma mensagem, ou ocorre outro evento.

## Protocolos de rede

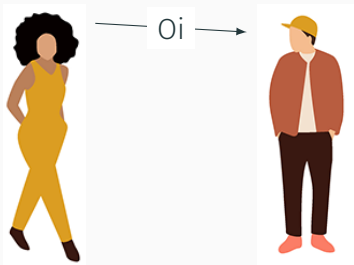
- máquinas no lugar de gente
- toda a comunicação na internet é regida por protocolos

### Formalmente

Protocolos definem o **fomato**, a **ordem** das **mensagens enviadas e recebidas** entre os membros de uma rede, além das **ações realizadas** na transmissão e recepção de mensagens.



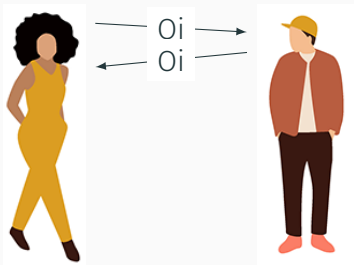
# Exemplos de protocolos



tempo

A vertical line with a downward-pointing arrow at the bottom, representing the progression of time.

# Exemplos de protocolos



tempo



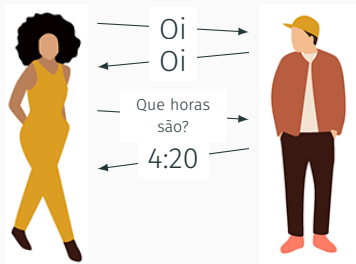
# Exemplos de protocolos



tempo

A vertical line with a downward-pointing arrow, representing the passage of time.

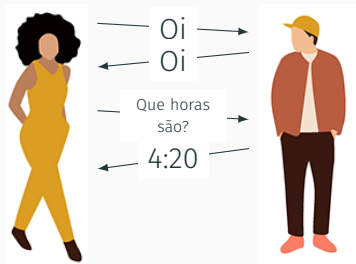
# Exemplos de protocolos



tempo

A vertical line with a downward-pointing arrow, indicating the progression of time.

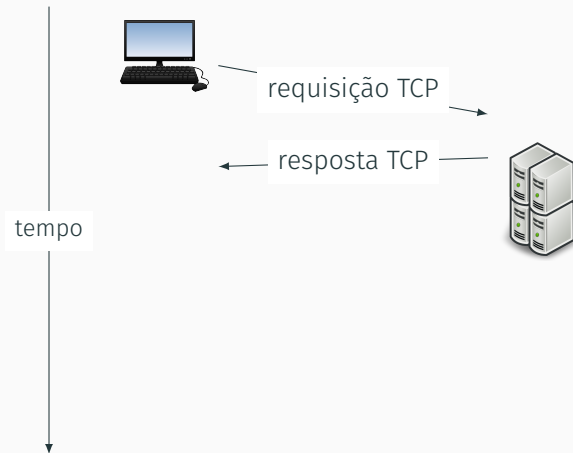
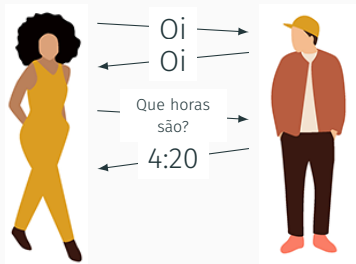
# Exemplos de protocolos



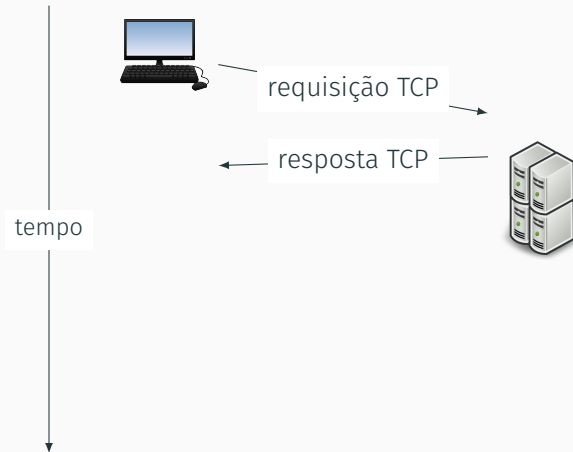
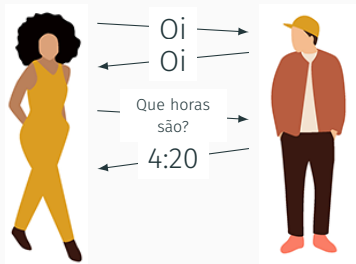
tempo



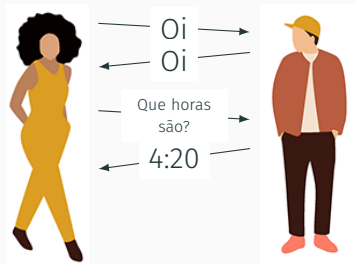
# Exemplos de protocolos



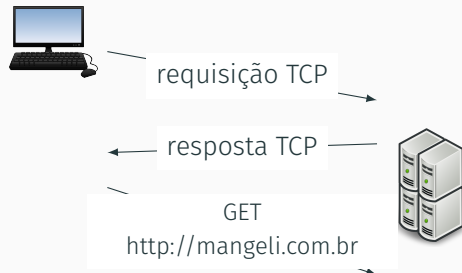
# Exemplos de protocolos



# Exemplos de protocolos

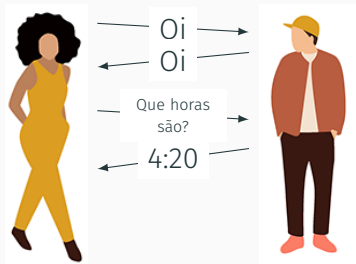


tempo

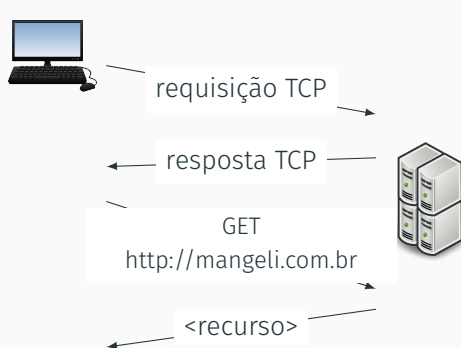




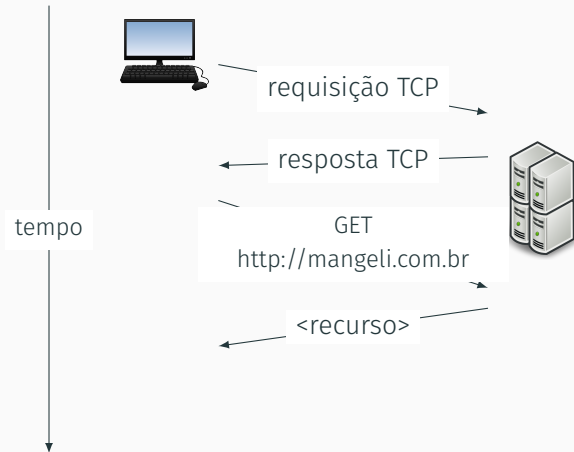
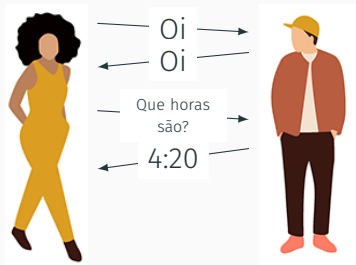
# Exemplos de protocolos



tempo



# Exemplos de protocolos



Outros protocolos humanos?

Nós já vimos que a internet é complexa, composta de diversos componentes e pode ser vista por diferentes camadas de abstração:

- servidores
- computadores
- roteadores
- conexões de diversos tipos
- protocolos
- aplicações
- dispositivos

Nós já vimos que a internet é complexa, composta de diversos componentes e pode ser vista por diferentes camadas de abstração:

- servidores
- computadores
- roteadores
- conexões de diversos tipos
- protocolos
- aplicações
- dispositivos

## Desafio

Como organizar?

# Pilha de protocolos da internet

- aplicação

viabilizar aplicações de rede: FTP, SMTP, HTTP



# Pilha de protocolos da internet

- aplicação

viabilizar aplicações de rede: FTP, SMTP, HTTP

- transporte

transferência de dados entre processos: TCP, UDP



# Pilha de protocolos da internet

- aplicação

viabilizar aplicações de rede: FTP, SMTP, HTTP

- transporte

transferência de dados entre processos: TCP, UDP

- rede

roteamento de datagramas da origem até o destino:

IP



# Pilha de protocolos da internet

- **aplicação**  
viabilizar aplicações de rede: FTP, SMTP, HTTP
- **transporte**  
transferência de dados entre processos: TCP, UDP
- **rede**  
roteamento de datagramas da origem até o destino:  
IP
- **enlace**  
transferência de dados entre elementos vizinhos na rede: Ethernet, 802.11(WiFi), PPP





# Pilha de protocolos da internet

- **aplicação**  
viabilizar aplicações de rede: FTP, SMTP, HTTP
- **transporte**  
transferência de dados entre processos: TCP, UDP
- **rede**  
roteamento de datagramas da origem até o destino:  
IP
- **enlace**  
transferência de dados entre elementos vizinhos na rede: Ethernet, 802.11(WiFi), PPP
- **físico**  
movimenta 'bits' no meio físico



# Protocolos da camada de aplicação

## Definem

- **tipos de mensagem** – request, response, etc
- **sintaxe** – quais campos e como são delimitados
- **semântica** – o que cada campo significa
- **regras** – quando e como processos respondem ou enviam mensagens

# Protocolos da camada de aplicação

## Definem

- **tipos de mensagem** – request, response, etc
- **sintaxe** – quais campos e como são delimitados
- **semântica** – o que cada campo significa
- **regras** – quando e como processos respondem ou enviam mensagens

## Protocolo Abertos

- Facilitam a interoperabilidade
- São definidos por padrões
- Exemplos: HTTP, SMTP

# Protocolos da camada de aplicação

## Definem

- **tipos de mensagem** – request, response, etc
- **sintaxe** – quais campos e como são delimitados
- **semântica** – o que cada campo significa
- **regras** – quando e como processos respondem ou enviam mensagens

## Protocolo Abertos

- Facilitam a interoperabilidade
- São definidos por padrões
- Exemplos: HTTP, SMTP

## Protocolos Proprietários?

## Protocolos da camada de transporte da Internet - TCP

- **transporte confiável** entre o processo emissor e o receptor
- **controle de fluxo** – o emissor não sobrecarrega o receptor
- **controle de congestionamento** – desacelera o emissor quando a rede sobrecarrega
- **não oferece** – segurança, controle de tempo, garantia de desempenho (throughput) mínimo
- **orientado a conexão** – necessita de configuração entre os processos cliente e servidor

- transporte de dados não confiável
- não oferece – controle de fluxo, confiabilidade, controle de congestionamento, controle de tempo, garantia de desempenho (throughput) mínimo, configuração de conexão

- transporte de dados **não confiável**
- **não oferece** – controle de fluxo, confiabilidade, controle de congestionamento, controle de tempo, garantia de desempenho (throughput) mínimo, configuração de conexão

**Por que UDP existe?**

- **TCP e UDP** não oferecem encriptação, todas as informações transitam pela internet como texto



- **TCP e UDP** não oferecem encriptação, todas as informações transitam pela internet como texto
- **SSL** oferece:

- **TCP e UDP** não oferecem encriptação, todas as informações transitam pela internet como texto
- **SSL** oferece:
  - encriptação para conexões TCP

- **TCP e UDP** não oferecem encriptação, todas as informações transitam pela internet como texto
- **SSL** oferece:
  - encriptação para conexões TCP
  - integridade dos dados

- **TCP e UDP** não oferecem encriptação, todas as informações transitam pela internet como texto
- **SSL** oferece:
  - encriptação para conexões TCP
  - integridade dos dados
  - autenticação de destinatário

- **TCP e UDP** não oferecem encriptação, todas as informações transitam pela internet como texto
- **SSL** oferece:
  - encriptação para conexões TCP
  - integridade dos dados
  - autenticação de destinatário
- **SSL** está na camada de aplicação

Desenvolver para web é construir programas:

- que rodam em sistemas diferentes
- se comunicam através de uma rede

Desenvolver para web é construir programas:

- que rodam em sistemas diferentes
- se comunicam através de uma rede

Desenvolver para web não é desenvolver programas para dispositivos do núcleo da internet como roteadores e switches:

- dispositivos do núcleo da internet não rodam programas
- aplicações em sistemas finais permitem desenvolvimento e distribuição rápidos

# Protocolos

---

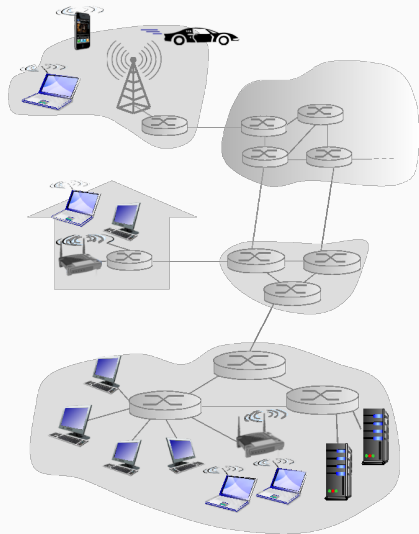
## Arquitecturas



# Arquitetura Cliente Servidor

## Servidor

- sempre 'no ar'
- endereço IP permanente



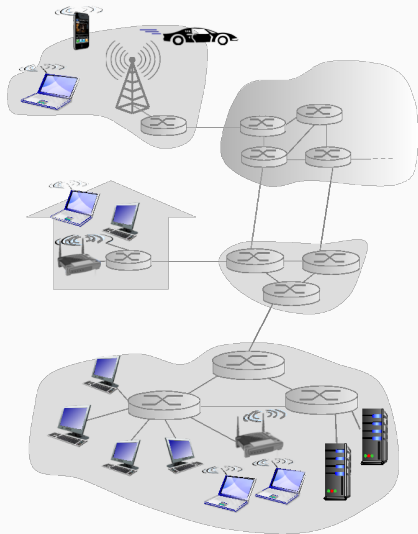
# Arquitetura Cliente Servidor

## Servidor

- sempre 'no ar'
- endereço IP permanente

## Clientes

- comunicam-se com o servidor
- podem ter conexões intermitentes
- podem ter endereços IP dinâmicos
- não se comunicam diretamente entre si



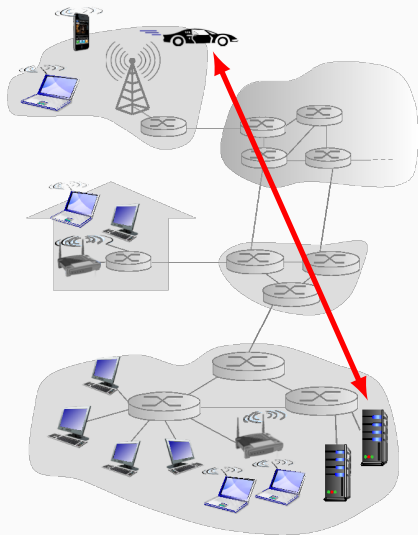
# Arquitetura Cliente Servidor

## Servidor

- sempre 'no ar'
- endereço IP permanente

## Clientes

- comunicam-se com o servidor
- podem ter conexões intermitentes
- podem ter endereços IP dinâmicos
- não se comunicam diretamente entre si



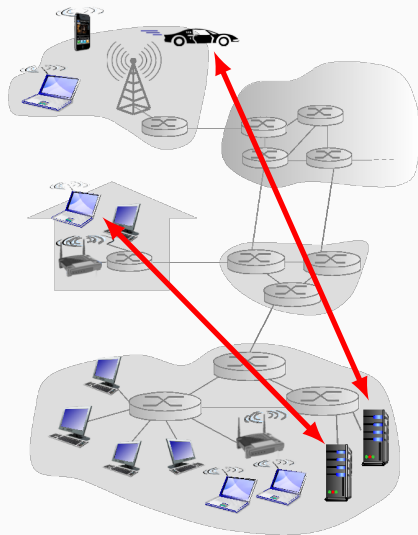
# Arquitetura Cliente Servidor

## Servidor

- sempre 'no ar'
- endereço IP permanente

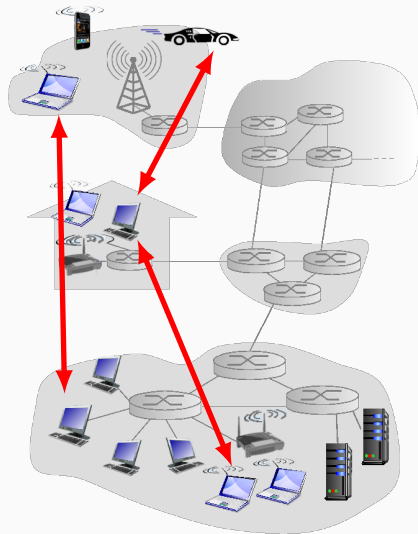
## Clientes

- comunicam-se com o servidor
- podem ter conexões intermitentes
- podem ter endereços IP dinâmicos
- não se comunicam diretamente entre si



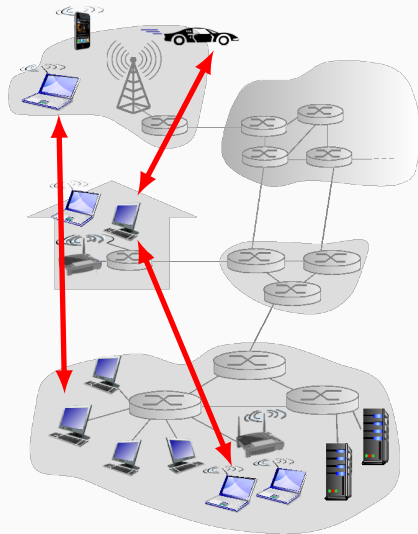
# Arquitetura P2P

- não existe servidor sempre ligado



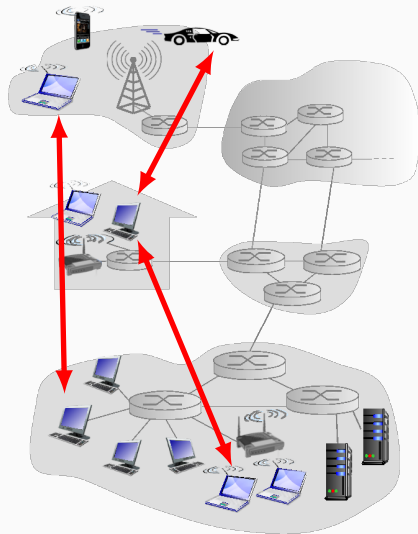
# Arquitetura P2P

- não existe servidor sempre ligado
- comunicação direta entre sistemas finais arbitrários



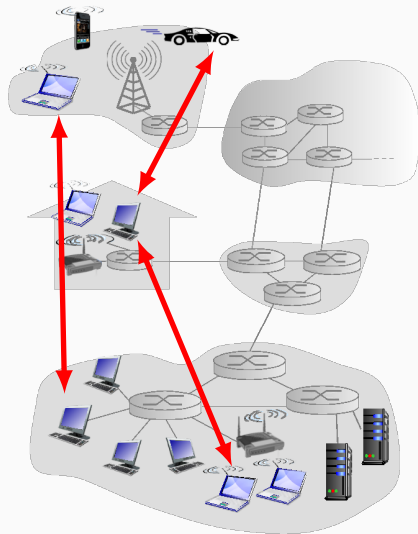
# Arquitetura P2P

- não existe servidor sempre ligado
- comunicação direta entre sistemas finais arbitrários
- usuários (peers) requisitam serviços de outros usuários e, ao mesmo tempo, fornecem serviços para outros usuários



# Arquitetura P2P

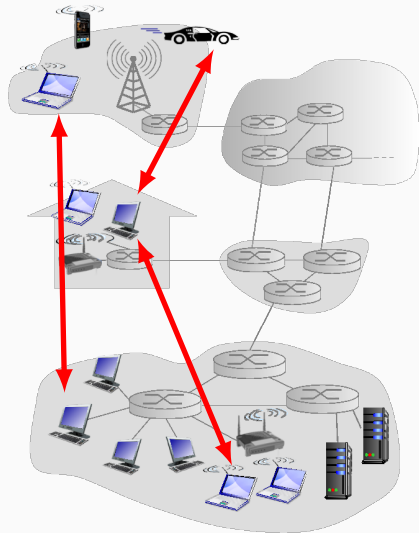
- não existe servidor sempre ligado
- comunicação direta entre sistemas finais arbitrários
- usuários (peers) requisitam serviços de outros usuários e, ao mesmo tempo, fornecem serviços para outros usuários
  - auto escalável – novos usuários trazem nova capacidade de fornecer e demandar serviços





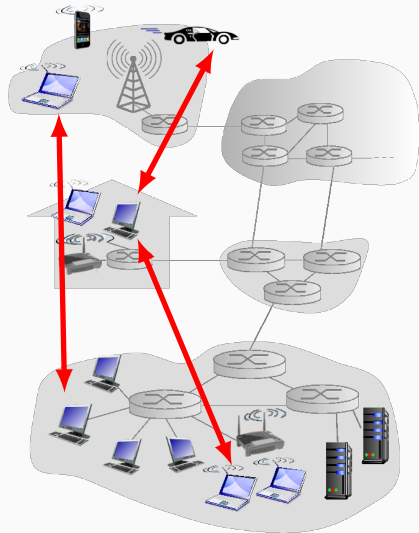
# Arquitetura P2P

- não existe servidor sempre ligado
- comunicação direta entre sistemas finais arbitrários
- usuários (peers) requisitam serviços de outros usuários e, ao mesmo tempo, fornecem serviços para outros usuários
  - auto escalável – novos usuários trazem nova capacidade de fornecer e demandar serviços
- usuários são conectados de forma intermitente e mudam seu endereço IP



# Arquitetura P2P

- não existe servidor sempre ligado
- comunicação direta entre sistemas finais arbitrários
- usuários (peers) requisitam serviços de outros usuários e, ao mesmo tempo, fornecem serviços para outros usuários
  - auto escalável – novos usuários trazem nova capacidade de fornecer e demandar serviços
- usuários são conectados de forma intermitente e mudam seu endereço IP
  - gerenciamento bastante complexo



**Processos** são programas rodando em um nó da rede.

- dentro do mesmo computador, os processos se comunicam usando comunicação inter-processos que é definida pelo SO
- processos em diferentes computadores se comunicam através de troca de mensagens

# Comunicação entre processos

**Processos** são programas rodando em um nó da rede.

- dentro do mesmo computador, os processos se comunicam usando comunicação inter-processos que é definida pelo SO
- processos em diferentes computadores se comunicam através de troca de mensagens

## clientes e servidores

**processo cliente:** processos que iniciam a comunicação

**processo servidor:** processo que espera pela conexão

- Aplicações P2P têm processos cliente e processos servidor

## Referenciando processos

- Apenas o endereço IP não é suficiente para identificar um processo
- Cada nó da rede tem seu próprio endereço IP
- Muitos processos podem rodar no mesmo nó
- O identificador de um processo é constituído um endereço IP e uma porta

# Referenciando processos

- Apenas o endereço IP não é suficiente para identificar um processo
- Cada nó da rede tem seu próprio endereço IP
- Muitos processos podem rodar no mesmo nó
- O identificador de um processo é constituído um endereço IP e uma porta

## Exemplos

- HTTP: 80
- email: 25

# Referenciando processos

- Apenas o endereço IP não é suficiente para identificar um processo
- Cada nó da rede tem seu próprio endereço IP
- Muitos processos podem rodar no mesmo nó
- O identificador de um processo é constituído um endereço IP e uma porta

## Exemplos

- HTTP: 80
- email: 25

## HTTP para mangeli.com.br

endereço IP

porta