



UNIVERSIDADE  
FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ

# A Internet e a arquitetura cliente-servidor

QXD0020 - Desenvolvimento de Software para Web

Prof. Bruno Góis Mateus ([brunomateus@ufc.br](mailto:brunomateus@ufc.br))

# Agenda

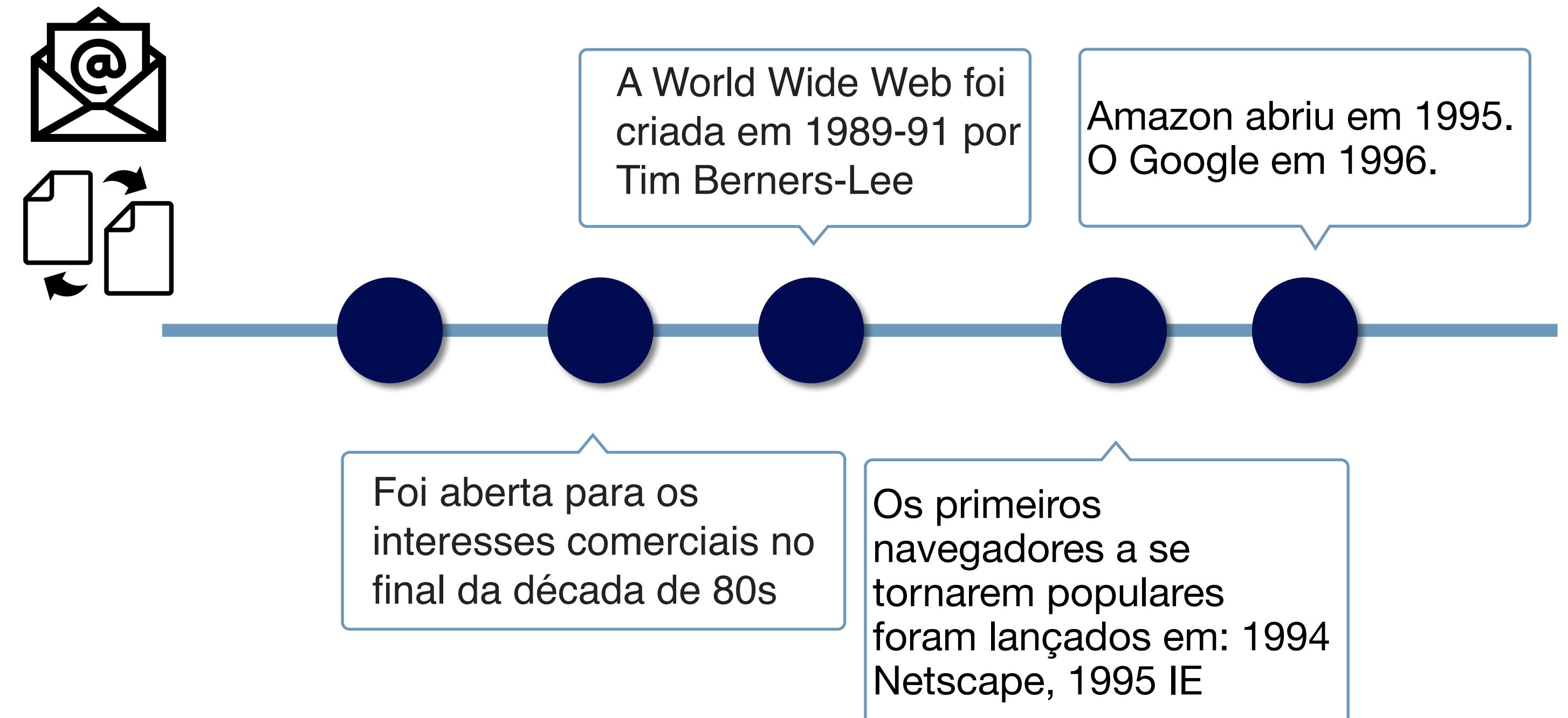
- A Internet
- Protocolos de comunicação
- A World Wide Web
- Protocolo HTTP
- Arquitetura Cliente Servidor

# A Internet



# A Internet

- Rede mundial de computadores
  - Dispositivos se comunicam por meio do **protocolo TCP/IP**
  - Começou com a rede do Departamento de Defesa dos Estados Unidos chamada **ARPANET (1960s-70s)**
- Os serviços iniciais eram:
  - Correspondência online(e-mail)
  - Transferência de arquivos



# A Internet

## Aspectos chaves

- As subredes são independentes
- Os computadores podem entrar e sair dinamicamente da rede
- Construída sob padrões abertos
- A ausência de um controle centralizado (em grande parte)
- Qualquer pessoa pode utilizar

# A Internet

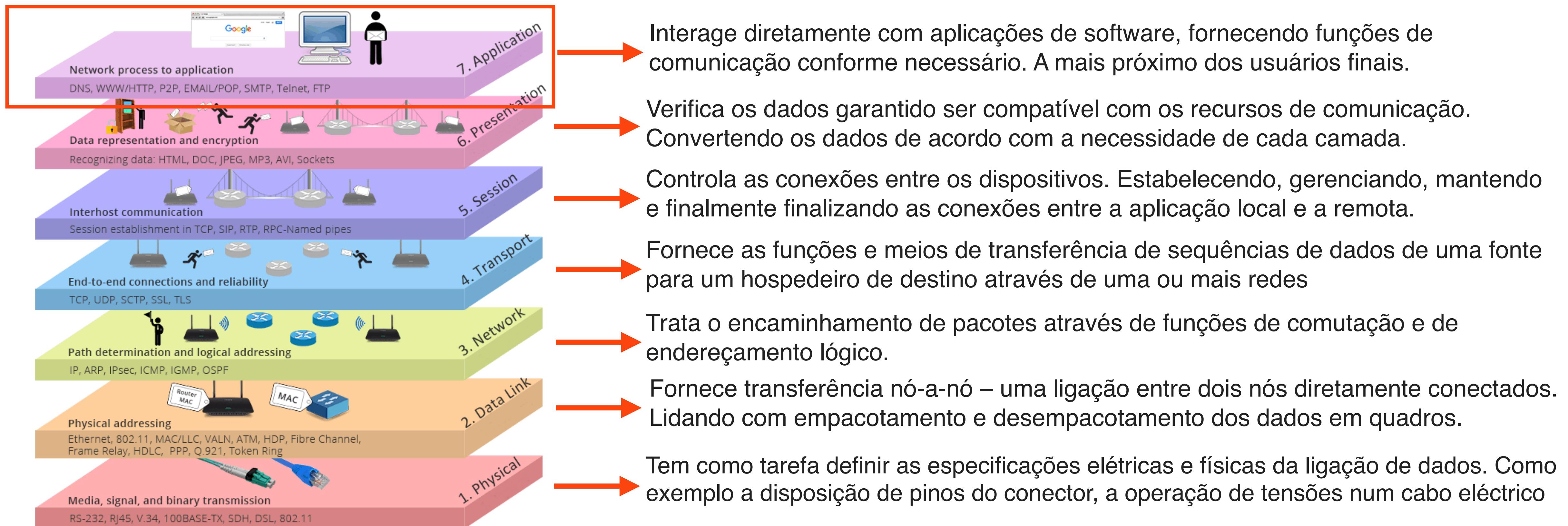
## Organizações

- Internet Engineering Task Force – IETF
  - Padrões do Protocolo IP
- Corporation for Assigned Names and Numbers - ICANN
  - Decide sobre o níveis mais altos de domínio
- World Wide Web Consortium - W3C
  - Padrões WEB



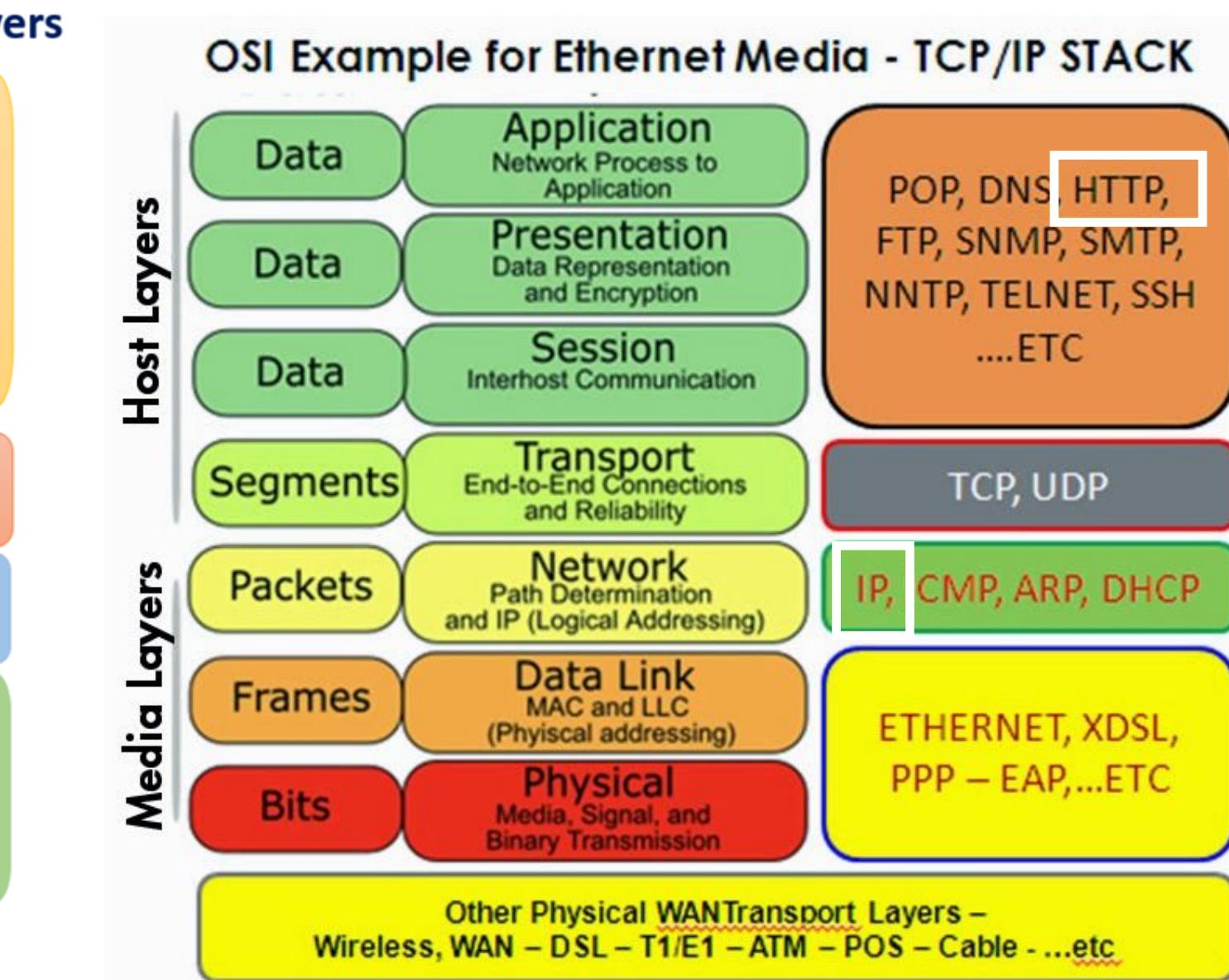
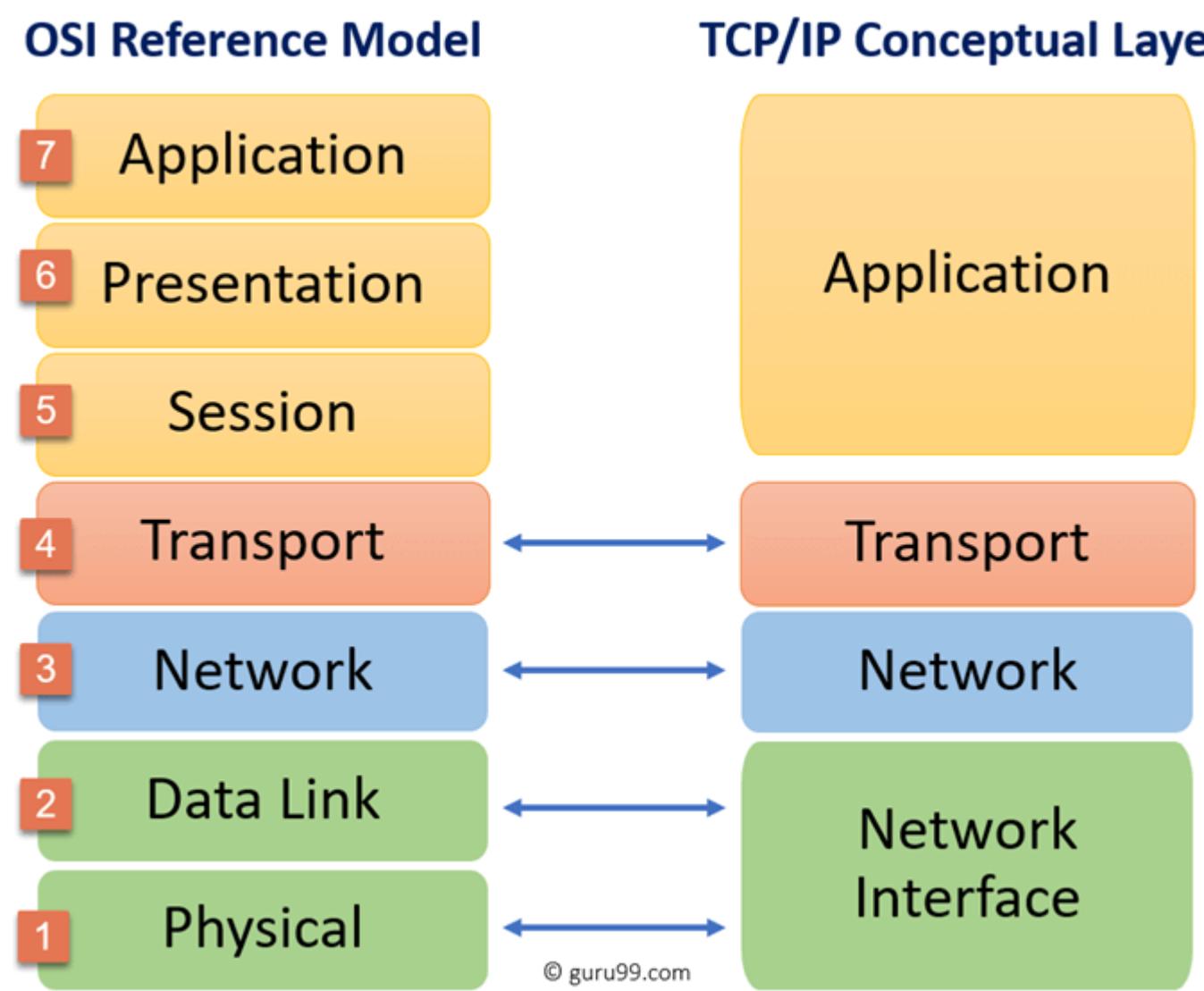
# A Internet

## Modelo de referência OSI

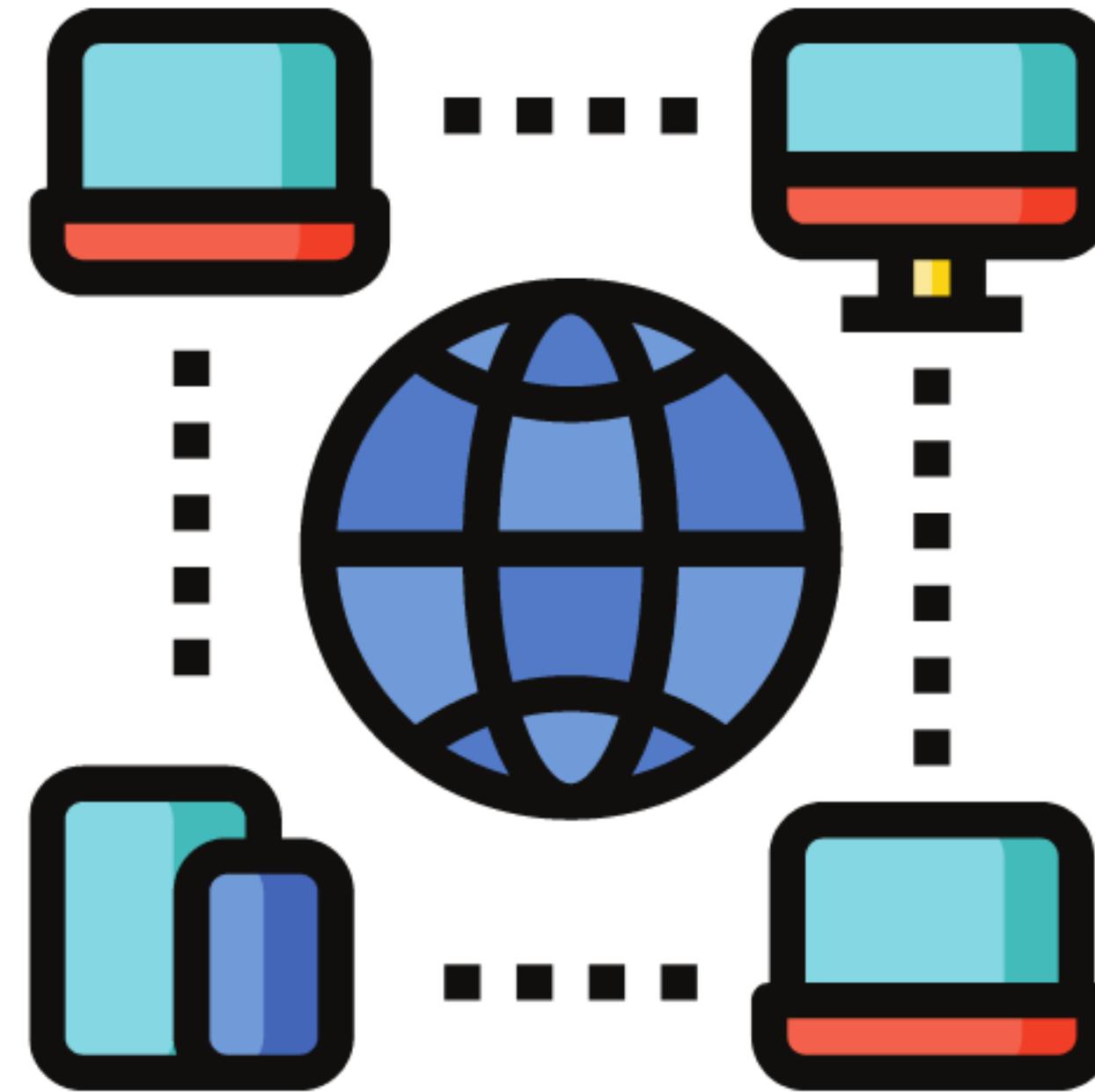


# A Internet

## Modelo OSI vs TCP/IP



# Protocolos de comunicação



# Protocolo

Define o formato e a ordem das mensagens trocadas entre duas ou mais entidades de comunicação, bem como as ações tomadas na transmissão e/ou recepção de uma mensagem ou outro evento.”

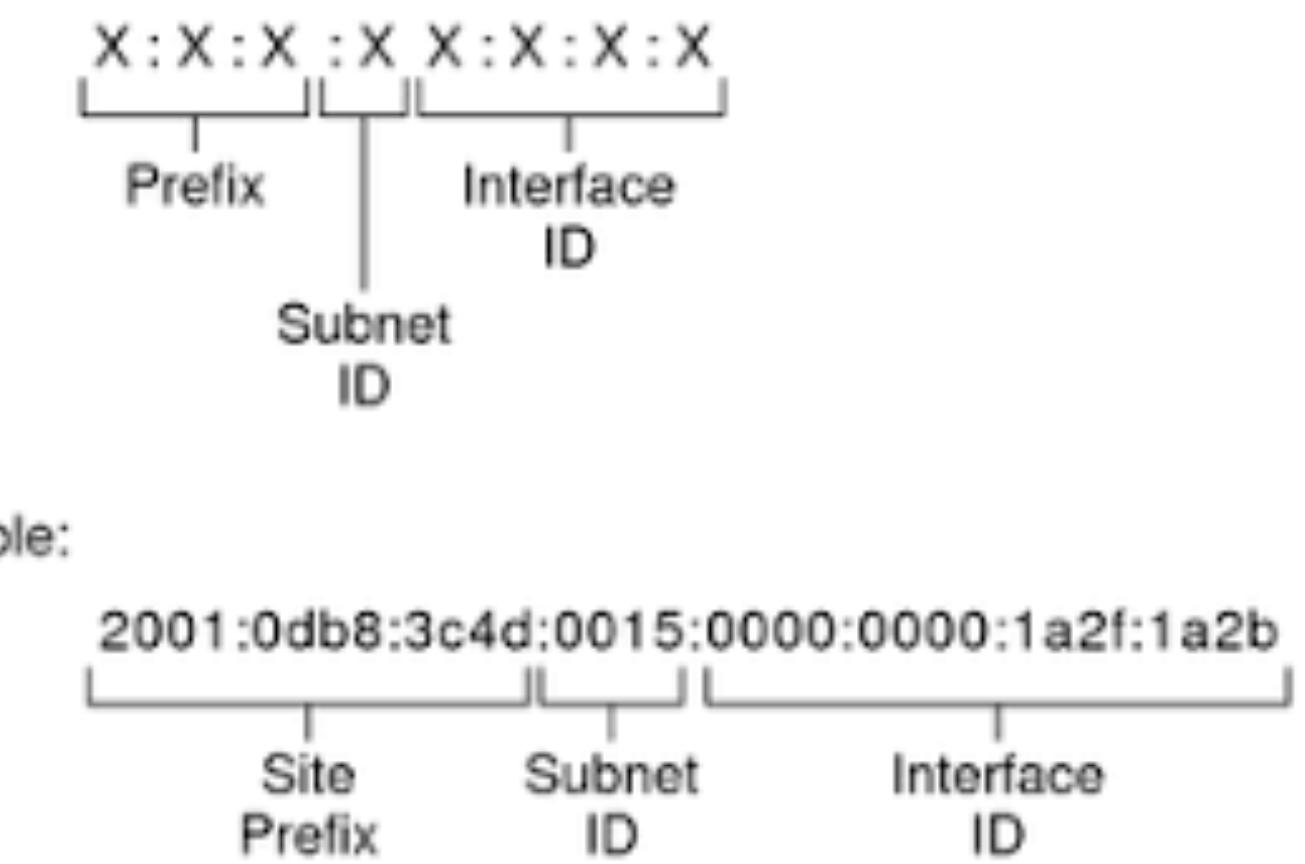
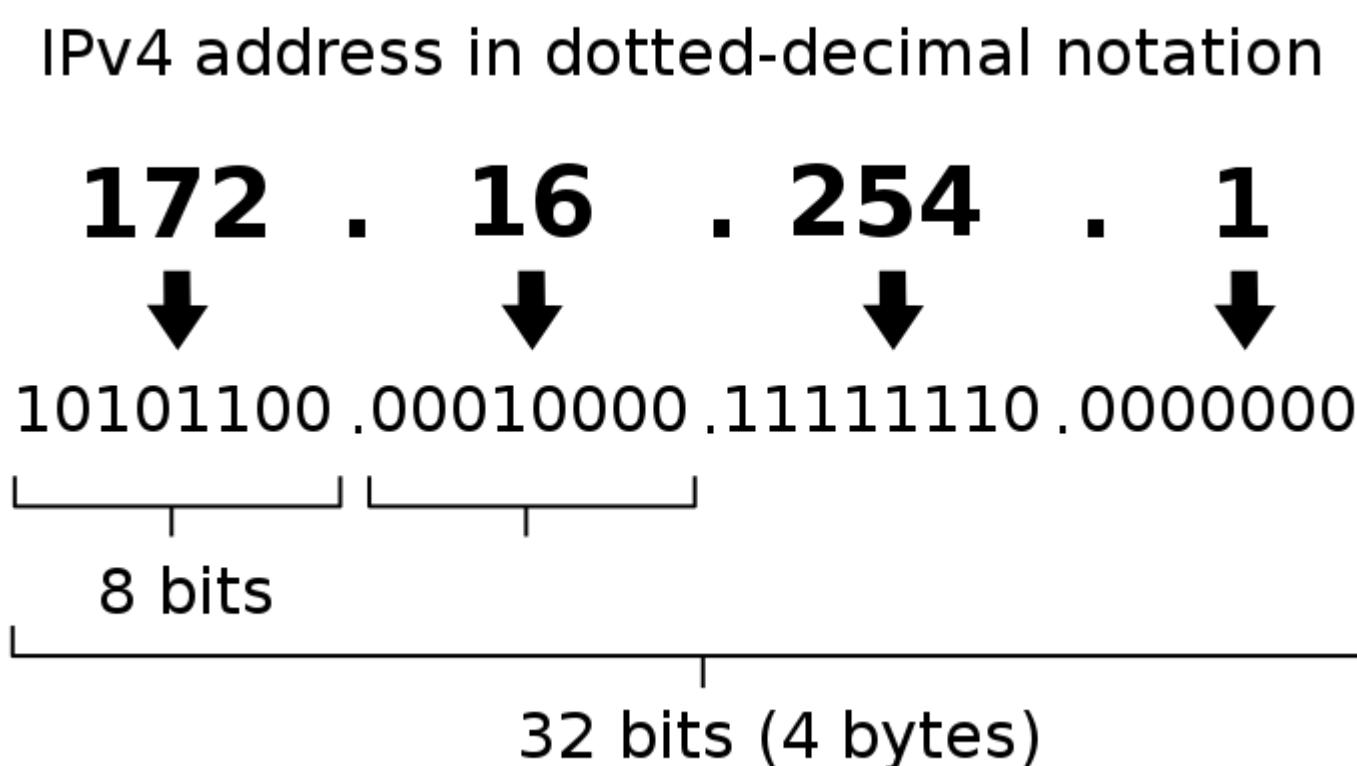
[Kurose, 2009]

[Κυροσε, 2009]

- "Regras que governam" a sintaxe, semântica e sincronização da comunicação
- Podem ser implementados pelo hardware, software ou por uma combinação dos dois

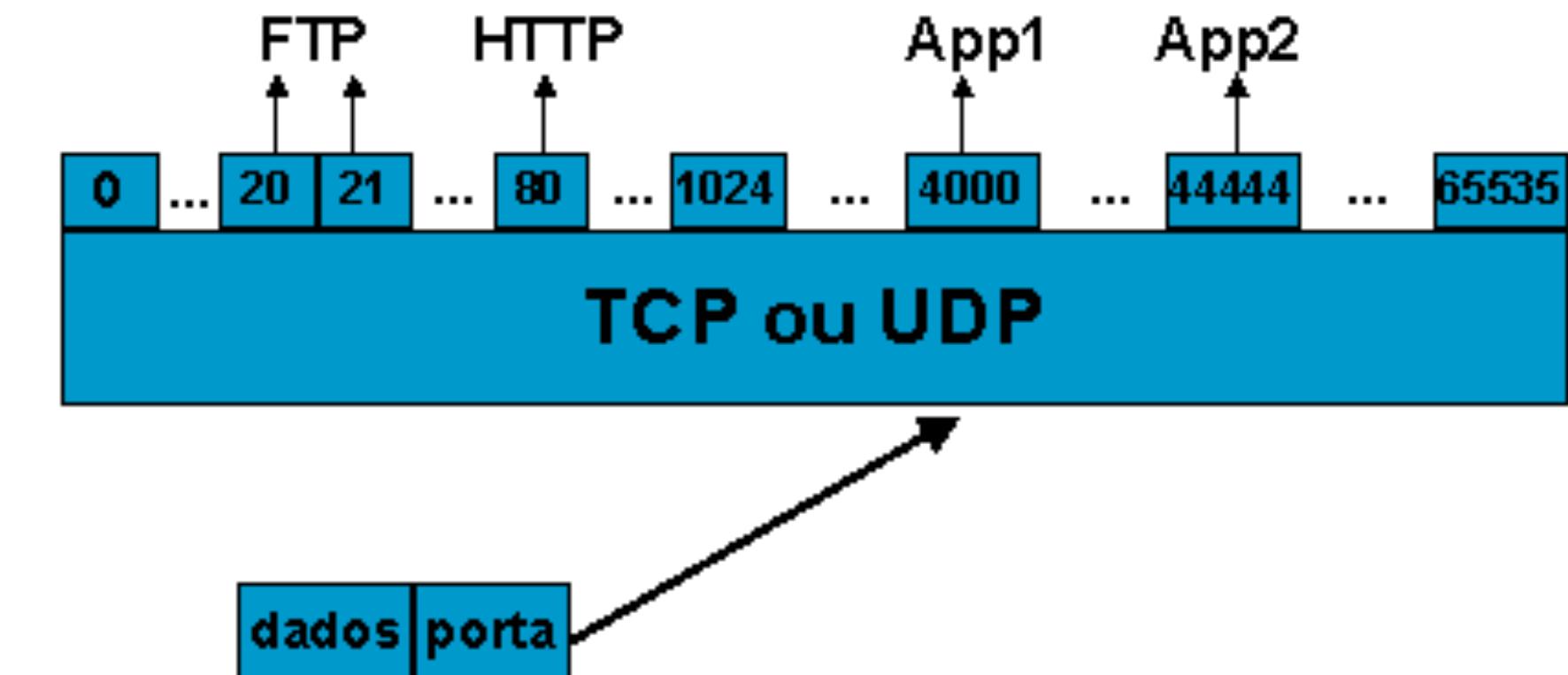
# Protocolo IP

- Um protocolo simples para tentar enviar dados entre computadores
  - Identifica unicamente um host da rede
  - Cada dispositivo possui um endereço IP de 32 bits, divididos em 4 números de 8 bits (0-255)
    - Atribuído a cada interface



# Portas

- Identificam os processos origem e destino
- Viabilizam a comunicação fim-a-fim
- Sistema operacional oferece interface para especificar e acessar portas
- Permitem comunicação com diversas aplicações na mesma máquina



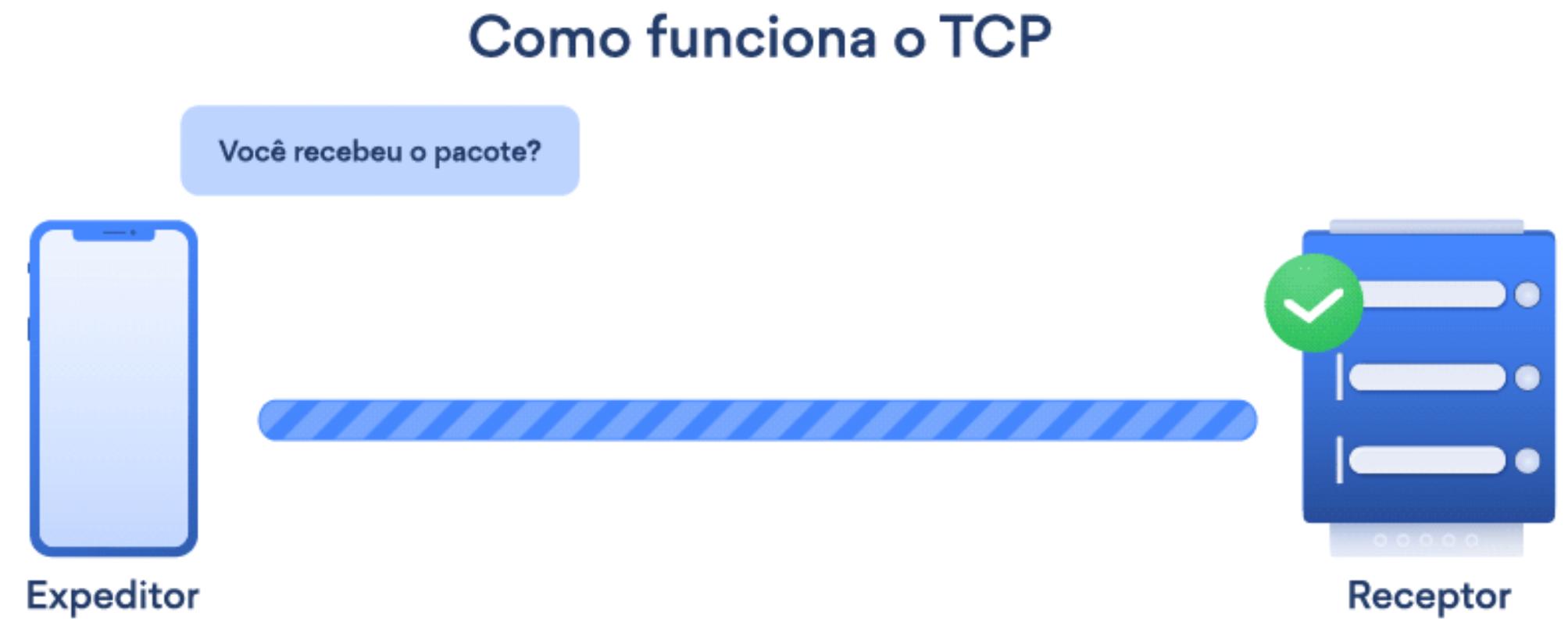
# O protocolo UDP

- Funciona como uma ponte para o protocolo IP
- Comunicação não orientada a conexão e não confiável
  - Entrega não garantida
  - Análogo ao sistema de correio
- Mais rápido que TCP
- Usado em:
  - Pequena quantidade de dados
  - Meios de comunicação seguros
  - Serviços que podem perder pequenas porções de informações



# O protocolo TCP

- Verifica a confiabilidade dos dados, garantindo que eles sejam enviados na ordem correta
- Verifica os possíveis erros nos pacotes de dados que fluem entre os usuários e dispositivos conectados à rede
- Utiliza o conceito de conexão para identificar os dois pontos envolvidos na comunicação
  - Análogo a uma chamada telefônica
  - Entrega garantida
- A comunicação é identificada por um par de endpoints.
  - Um endpoint é um par na forma Host, Port
  - Ex: (128.9.0.21, 1184) (128.10.2.3,25)
  - Permitem que uma determinada porta possa ser compartilhada por múltiplas conexões

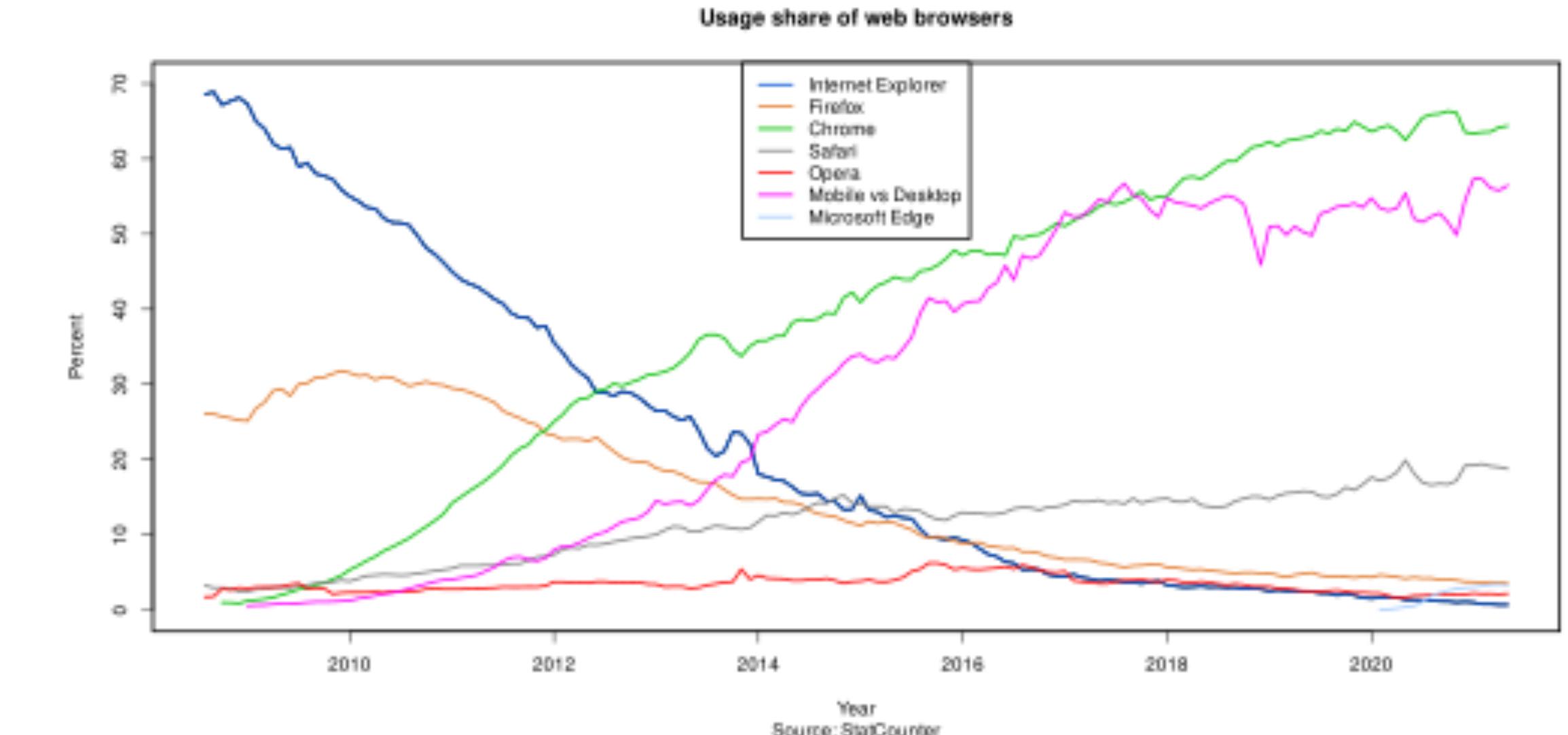


# A World Wide Web



# Servidores

- Servidores Web
  - Software que escutam as requisições por recursos
    - Apache
    - Micro\$oft Internet Server (IIS)
- Navegadores Web
  - Buscam e mostram os documentos disponibilizados por um servidor web
    - Firefox
    - Chrome
    - Safari
    - Opera
    - Spartan
    - Dolphin



# Domain Name System (DNS)

- Um conjunto de servidores que mapeiam os nomes para os endereços IP
  - www.ufc.br. -> 200.17.41.185
- A maioria dos sistemas operacionais mantém um cache local chamados de hosts file
  - Windows: C:/Windows/system32/drivers/etc/hosts
  - Mac: /private/etc/hosts
  - Linux: /etc/hosts

# Uniform Resource Identifier

- Identificador de Recursos Uniforme
  - É uma cadeia de caracteres usada para identificar ou denominar um recurso na Internet.
- Pode ser classificada como um localizador (URL) ou um nome (URN), ou ainda como ambos.
  - Exemplos:
    - URN
      - urn:isbn:0-486-27557-4
      - urn:issn:1535-3613
    - URL
      - <http://example.org/absolute/URI/path/to/resource.txt>
      - <ftp://example.org/resource.txt>

# Uniform Resource Locator

- Localizador de Recursos Universal
  - É uma URI que, além de identificar um recurso, provê meios de agir sobre, obter e representar este recurso, descrevendo seu mecanismo de acesso primário ou a localização na "rede".
- É o endereço de um recurso disponível em uma rede.
- Estrutura:
  - protocolo://máquina[:porta]/caminho/recurso

# Uniform Resource Locator



- Ao colocar uma URL no navegador, ele irá fazer:
  1. Perguntar ao servidor DNS pelo endereço IP do site `www.edu4java.com`
  2. Conectar ao endereço IP utilizando a porta 80
  3. Pedir ao servidor a página: GET `/es/androidgame/androidgame.7.html`
  4. Mostrar o resultado na tela do dispositivo

# O Protocolo HTTP



# O protocolo HTTP

- Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)
  - Usado desde 1990
  - Protocolo de comunicação utilizado para transferir dados por intranets e World Wide Web
  - Normalmente o servidor usa a porta 80
  - Responsável pelo tratamento de pedidos / respostas entre cliente e servidor Web
  - Forma padronizada de comunicação entre clientes e servidores da Web
  - Serve para transferência de texto e dados binários

# O protocolo HTTP

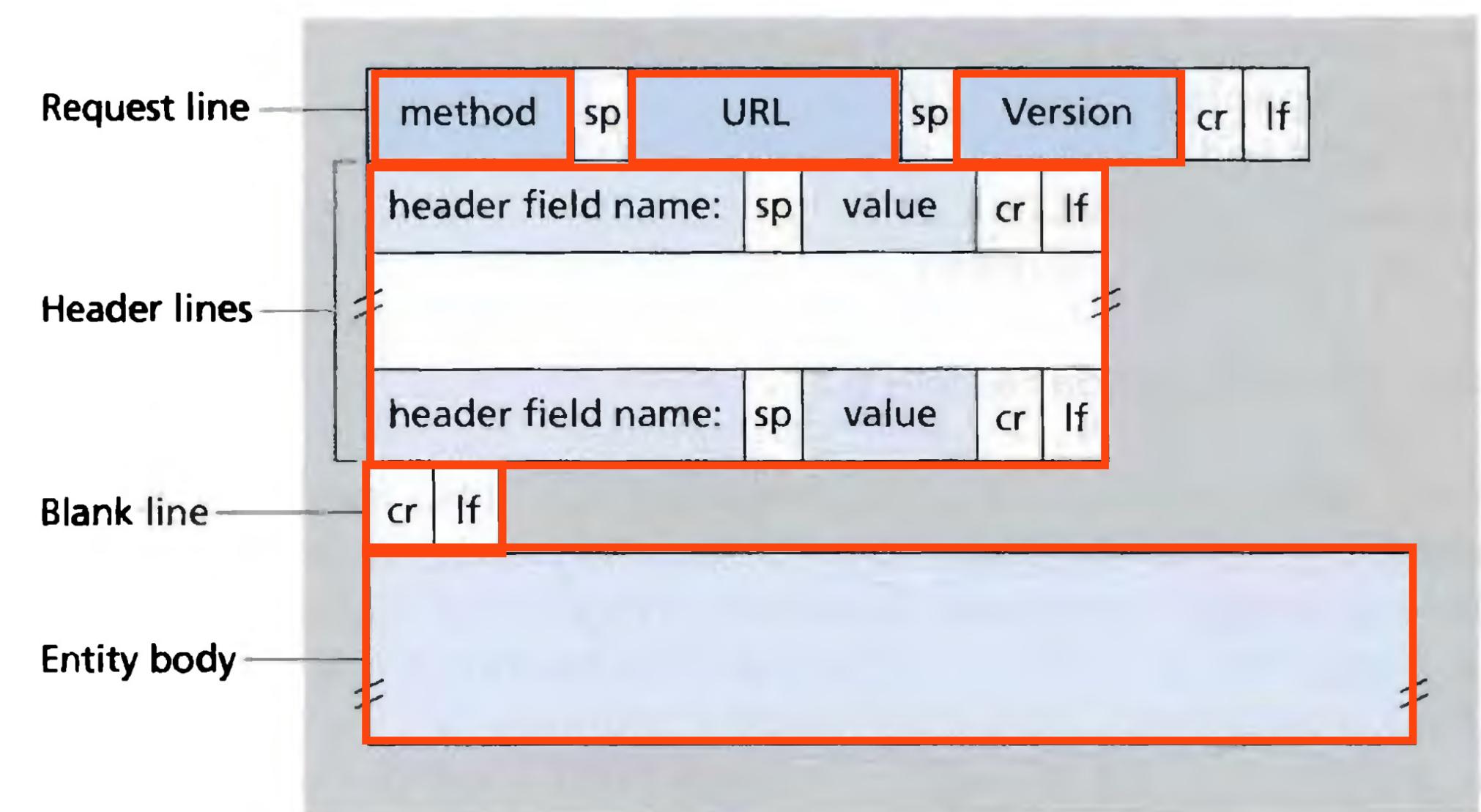
- Funcionamento:
  - O cliente envia uma mensagem de requisição de um recurso
  - O servidor envia uma mensagem de resposta ao cliente
- Composição das mensagens:
  - Cabeçalho
  - Corpo (opcional em certos casos)
- Stateless(Sem estados)
  - Não existe informações sobre o estado das conexões
  - Próxima ação independe da anterior
  - Realiza apenas uma ação por vez



# O protocolo HTTP

## Requisição

- Linha inicial
  - Método
  - Identificação do URI
  - Versão do HTTP utilizado
- Linhas de cabeçalhos
- Linha em branco obrigatória
- Corpo de mensagem opcional



# O protocolo HTTP

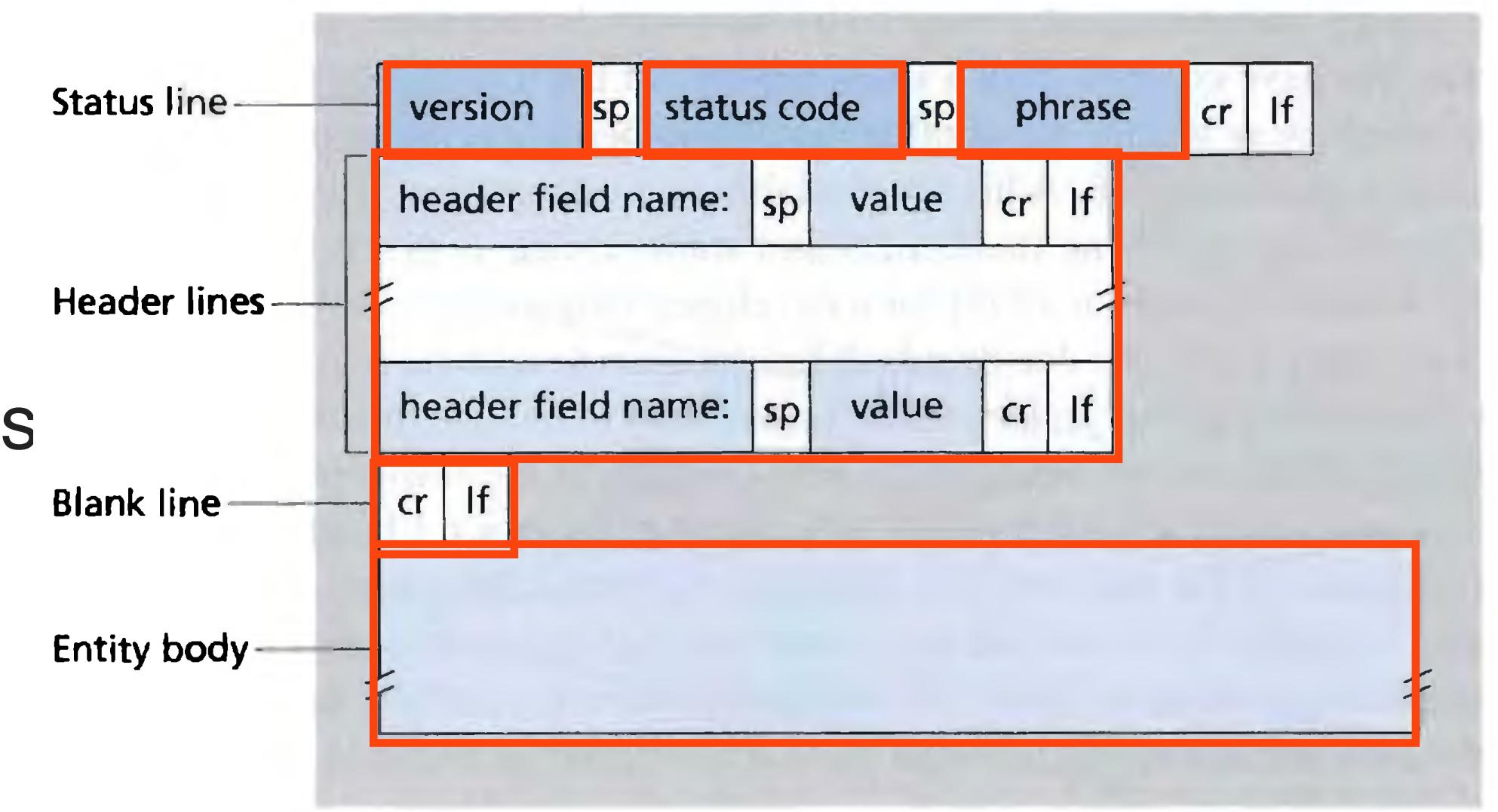
## Métodos

- **GET** – Solicita algum recurso
  - Dados são anexados à URL, ficando visíveis ao usuário
- **POST** – Envia dados referentes ao recurso especificado para serem processados
  - Dados são incluídos no corpo do comando
- **PUT** – Envia certo recurso
- **DELETE** – Exclui o recurso
- **HEAD** – Variação do GET em que o recurso não é retornado
- **TRACE** – Ecoa o pedido, de maneira que o cliente possa saber o que os servidores intermediários estão mudando em seu pedido
- **OPTIONS** – Recupera os métodos HTTP que o servidor aceita
- **CONNECT** – Converte a conexão de requisição em um túnel TCP/IP transparente, geralmente para facilitar a comunicação encriptada através de um proxy HTTP sem encriptação
- **PATCH** – Usado para aplicar modificações parciais em um recurso

# O protocolo HTTP

## Resposta

- Linha inicial ou linha de status
  - Versão do protocolo HTTP
  - Código de status da resposta
    - Fornece o resultado da requisição
    - Frase de justificativa que descreve o código do status
- Linhas de cabeçalhos
- Linha em branco obrigatória
- Corpo de mensagem opcional



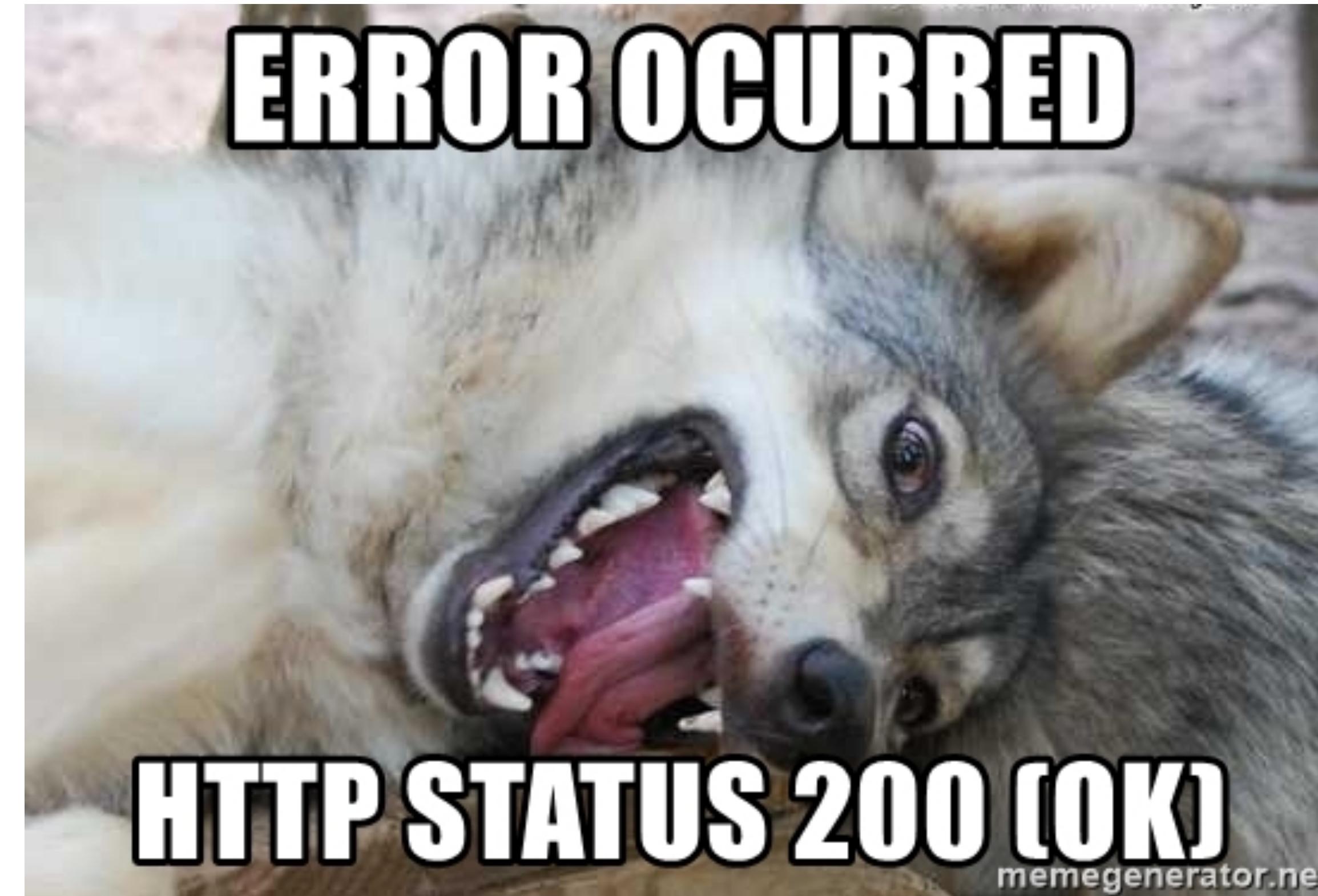
# O protocolo HTTP

## Código de Status

- É formado por três dígitos e o primeiro dígito representa a classe. São classificadas em cinco classes:
  - **1xx: Informational (Informação):**
    - Utilizada para enviar informações para o cliente de que sua requisição foi recebida e está sendo processada
  - **2xx: Success (Sucesso):**
    - Indica que a requisição do cliente foi bem sucedida
  - **3xx: Redirection (Redirecionamento):**
    - Informa a ação adicional que deve ser tomada para completar a requisição
  - **4xx: Client Error (Erro no cliente):**
    - Avisa que o cliente fez uma requisição que não pode ser atendida
  - **5xx: Server Error (Erro no servidor):**
    - Ocorreu um erro no servidor ao cumprir uma requisição válida
- O protocolo HTTP define somente alguns códigos em cada classe, mas cada servidor pode definir seus próprios códigos

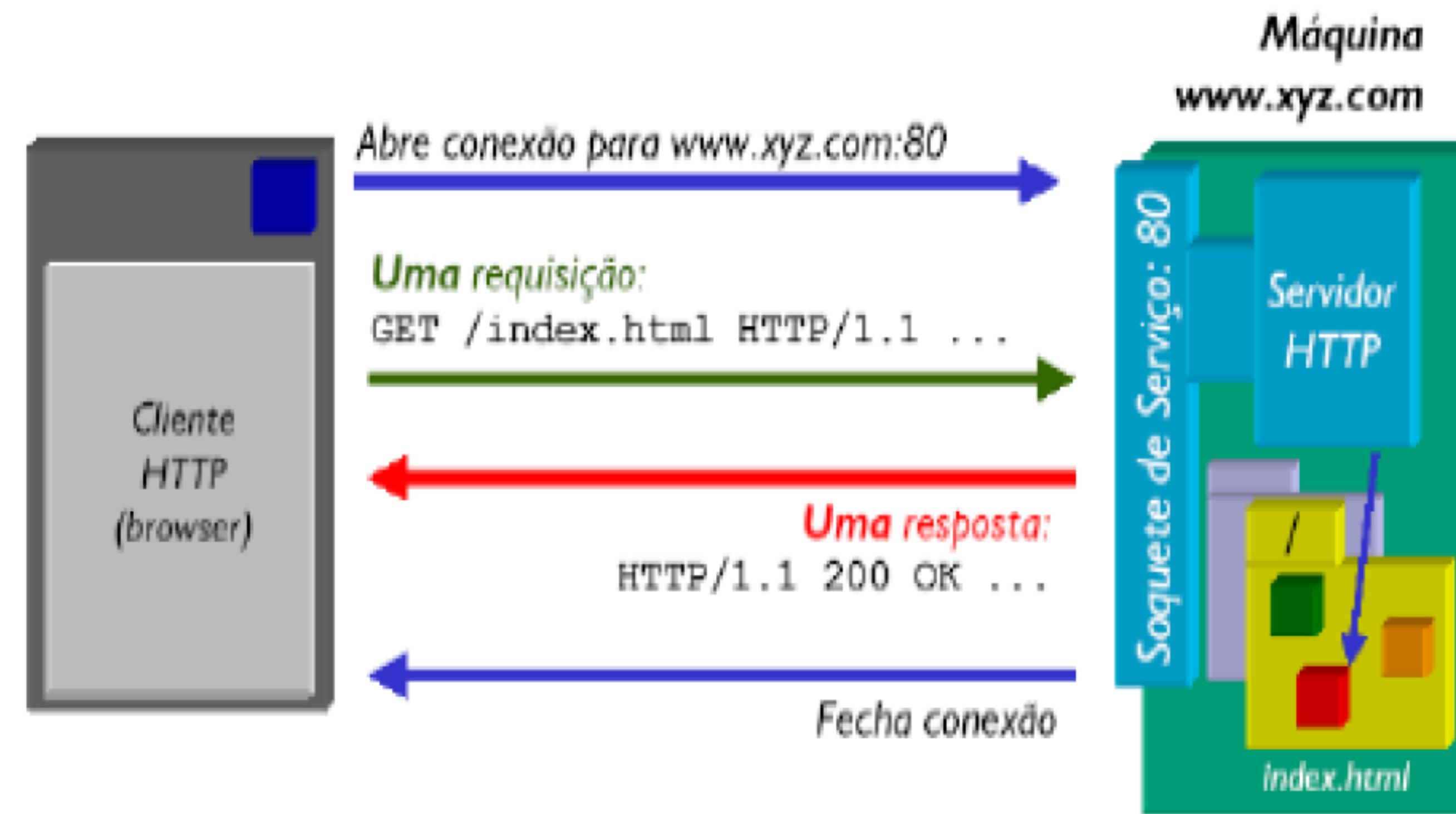
# O protocolo HTTP

## Código de Status

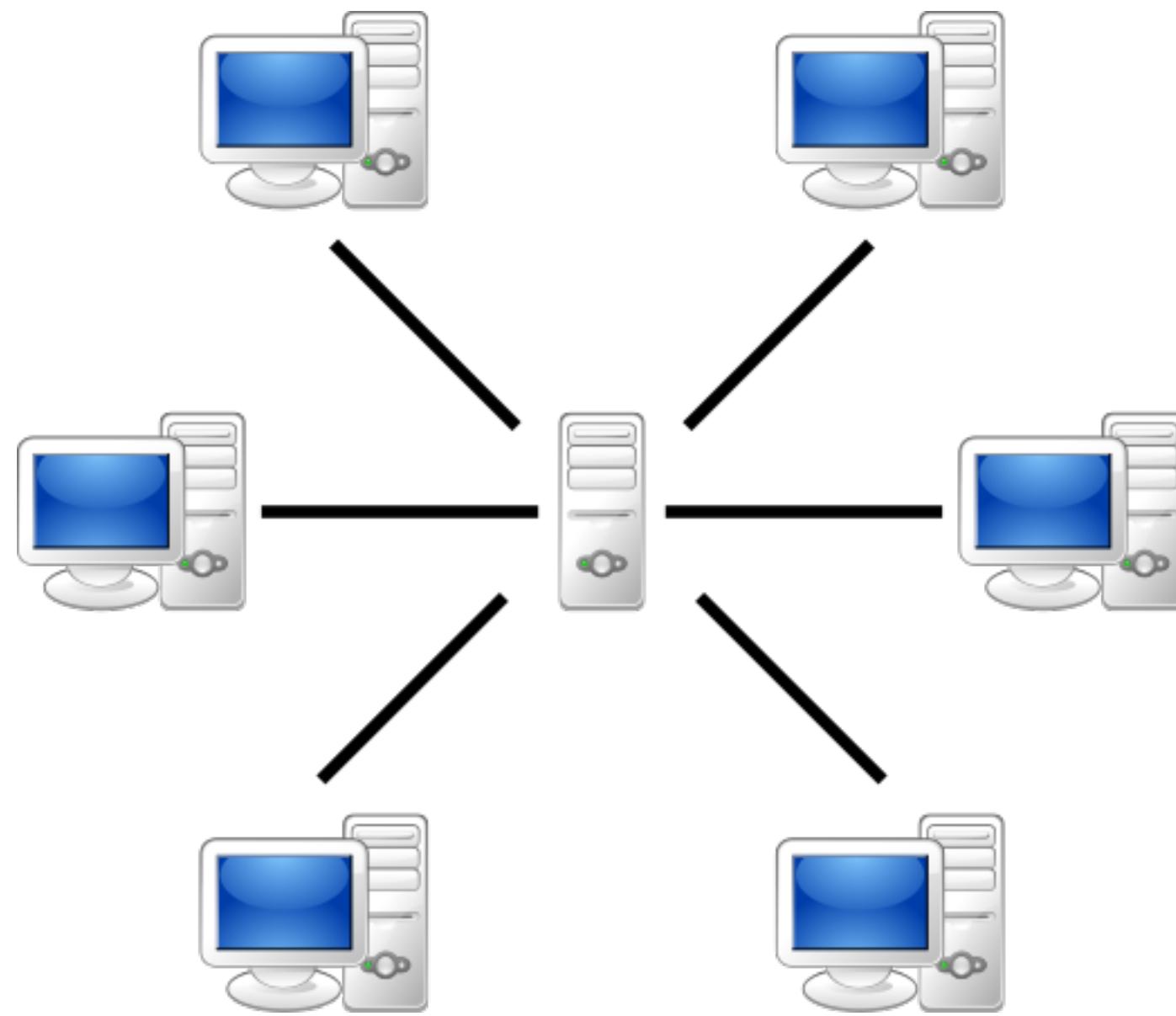


# O protocolo HTTP

## Comunicação



# Arquitetura Cliente Servidor



# Arquitetura Cliente Servidor

- Servidor
  - Oferece um ou mais serviços
  - Responsável por receber as requisições, executar o serviço e retornar o resultado
  - Um servidor espera por requisições em uma porta conhecida, reservada para o serviço
- Cliente
  - Requisita um serviço
  - Geralmente possui uma interface com o usuário
  - Um cliente aloca uma porta arbitrária disponível e não reservada
- Socket
  - Conjunto de funções para permitir a utilização do sistema de comunicação por processos no sistema operacional
  - Endereço IP + Porta

# Arquitetura Cliente Servidor

- No contexto web temos o seguinte:
  - Cliente Web
    - Envia **requisições HTTP** ao Servidor Web
    - Processa respostas HTTP recebidas
  - Servidor Web
    - Interpreta **requisições HTTP** do cliente
    - Gera uma resposta para o cliente
    - Devolve **resposta HTTP** ao cliente

Por hoje é só