Programação Web com Linguagens de Script

Thiago Rodrigues

Professor

- Thiago Rodrigues Medeiros

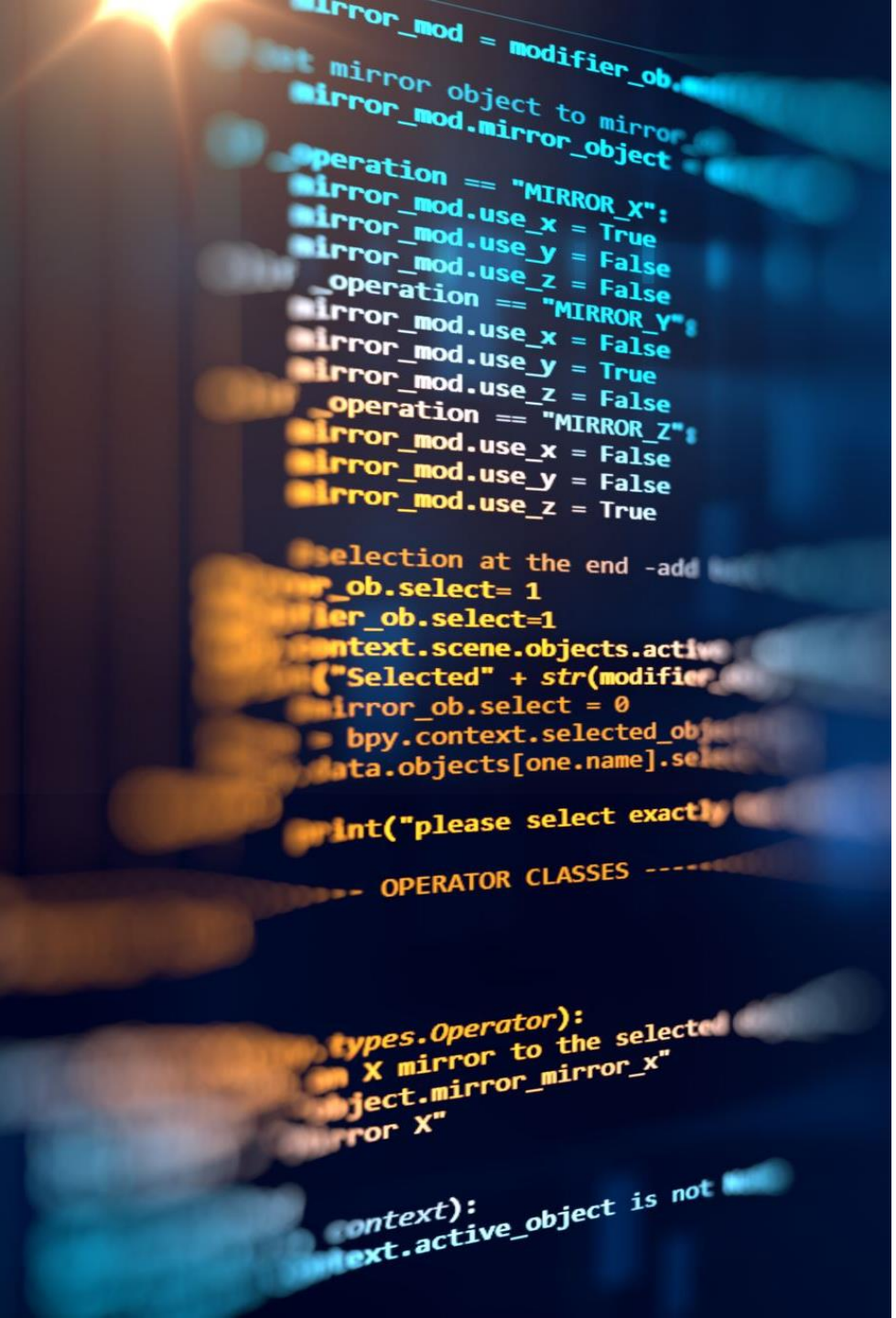
- Mini Bio: Professor, Desenvolvedor Back-End e Tech Lider

Agenda do Curso

- Relembrando HTTP
- Serviços Web
- API Rest
- Node.js
- Express Framework
- Desenvolvimento da API

Agenda

- O que é e como funciona o protocolo HTTP.
- Quais são as partes de um pedido HTTP.
- Quais são as partes de um a resposta HTTP.



O que é o protocolo HTTP?

- Hyper Text Transfer Protocol é o protocolo usado na World Wide Web para a distribuição e recuperação de informação.
- A troca de informações entre um browser e um servidor Web é toda feita através desse protocolo, que foi criado especificamente para a World Wide Web.
- A versão mais utilizada atualmente é a 1.1, definida pela especificação RFC 2616;
- Embora esta especificação devesse ser leitura obrigatória para todo desenvolvedor web, muitos nem sabem como ela se parece!!!
- 1990: HTTP 1.0; 1999: HTTP 1.1

O protocolo HTTP

- O HTTP define uma forma de **conversação** no estilo **pedido-resposta** entre um cliente (o **browser**) e um servidor (o **servidor Web**).

- Toda a conversação se dá no formato ASCII (texto puro) através de um conjunto de comandos simples baseados em palavras da língua inglesa.

Cliente HTTP

- Os clientes de uma conexão HTTP geralmente são os browsers.

- Atualmente alguns browsers se destacam no mercado:
  - Firefox,
  - Google Chrome,
  - Edge,
  - Opera
  - Brave
  - Safari

- Ambos os browsers são gratuitos.

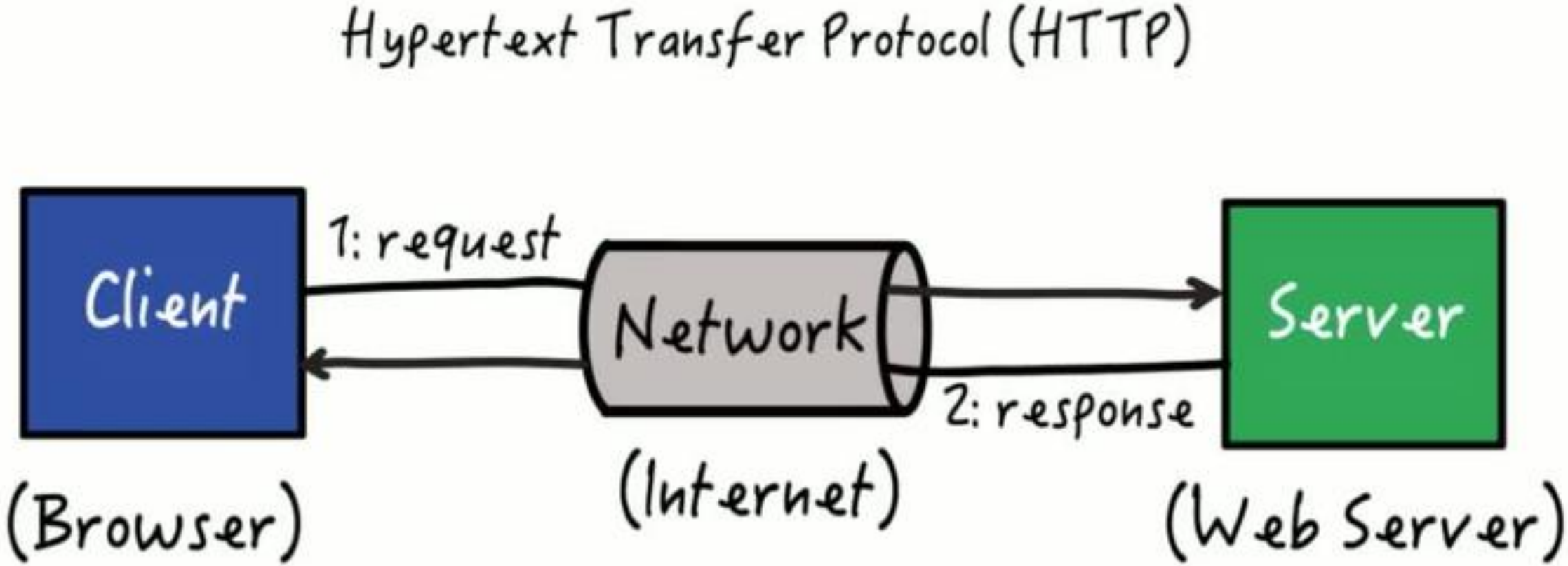
Servidores HTTP

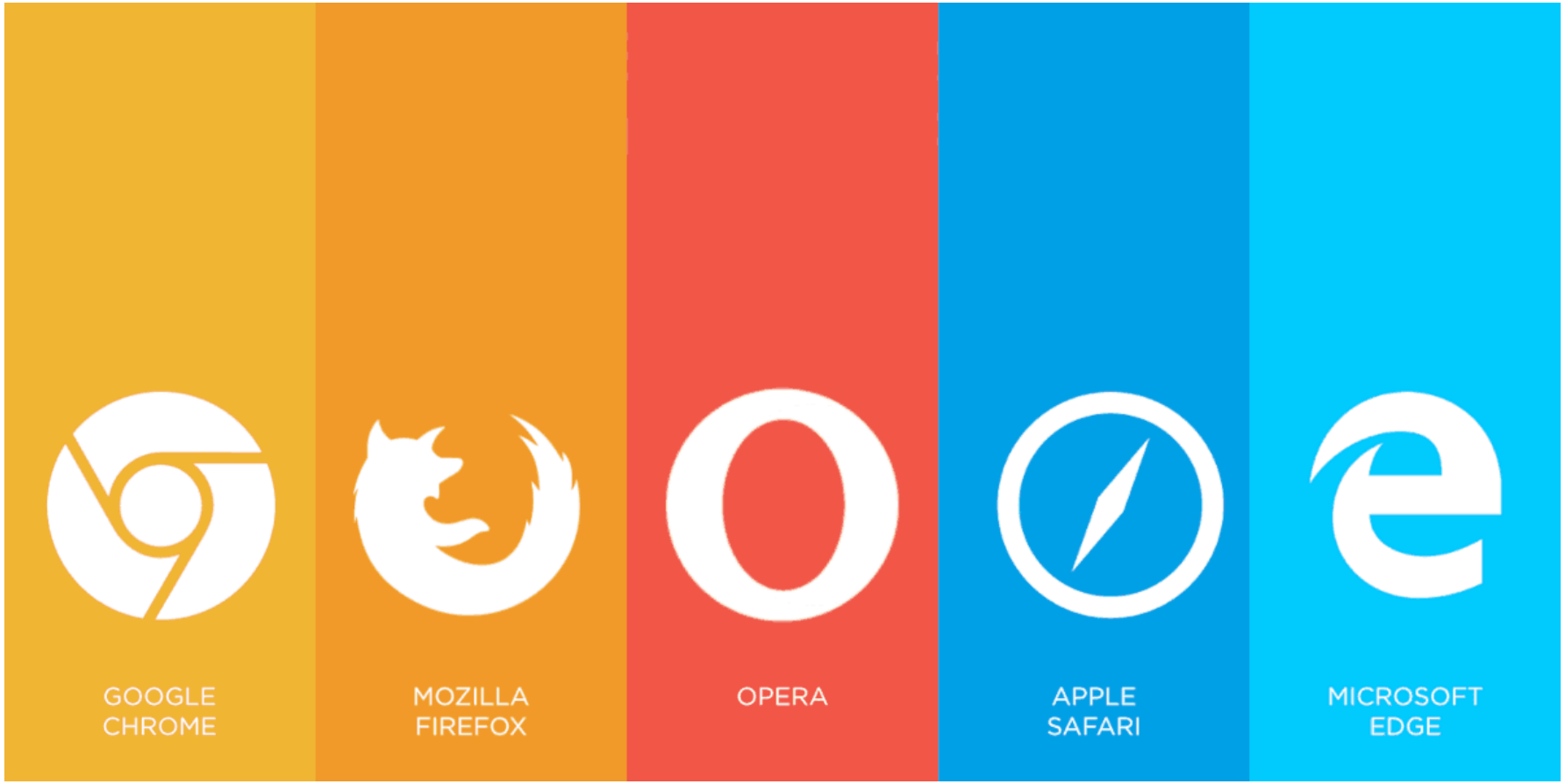
- Os servidores de uma conexão HTTP são máquinas que possuem softwares de servidores web.

- Os servidores Web de maior destaque atualmente no mercado são:
  - Apache HTTP Server;
  - Internet Information Server, da Microsoft;
  - Enterprise Server, da Netscape.
  - Apache Tomcat
  - Nginx

Protocolo HTTP

- O protocolo HTTP é baseado em **requisições** e **respostas** entre clientes e servidores;
- O cliente, navegador ou dispositivo que fará a requisição; também é conhecido como **user agent,** solicita um determinado recurso (resource), enviando um pacote de informações contendo alguns cabeçalhos (headers) a um URI ou, mais especificamente, URL;
- O servidor recebe estas informações e envia uma resposta, que pode ser um recurso ou simplesmente um outro cabeçalho.


Hypertext Transfer Protocol (HTTP)
Client
1: request
Network
Server
2: response
(Browser)
(Internet)
(Web Server)



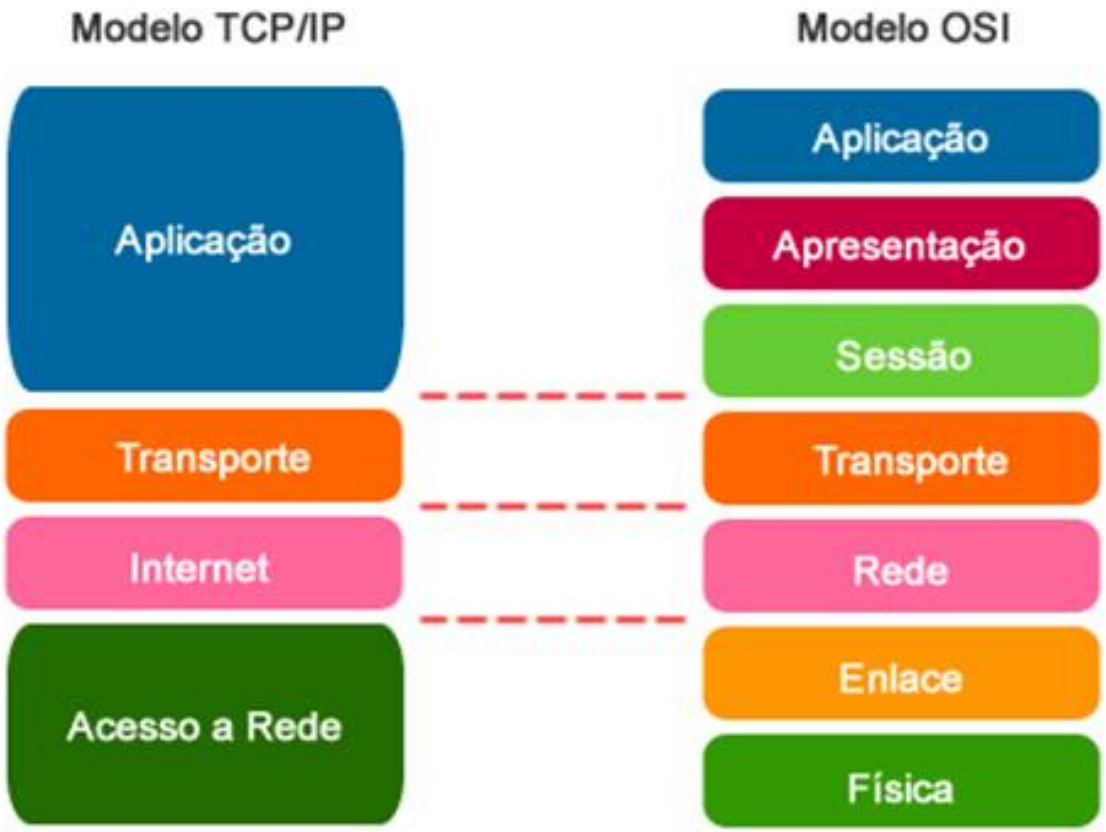
Quem são os User Agents?

Quem são os User Agents?

Quem são os Servidores

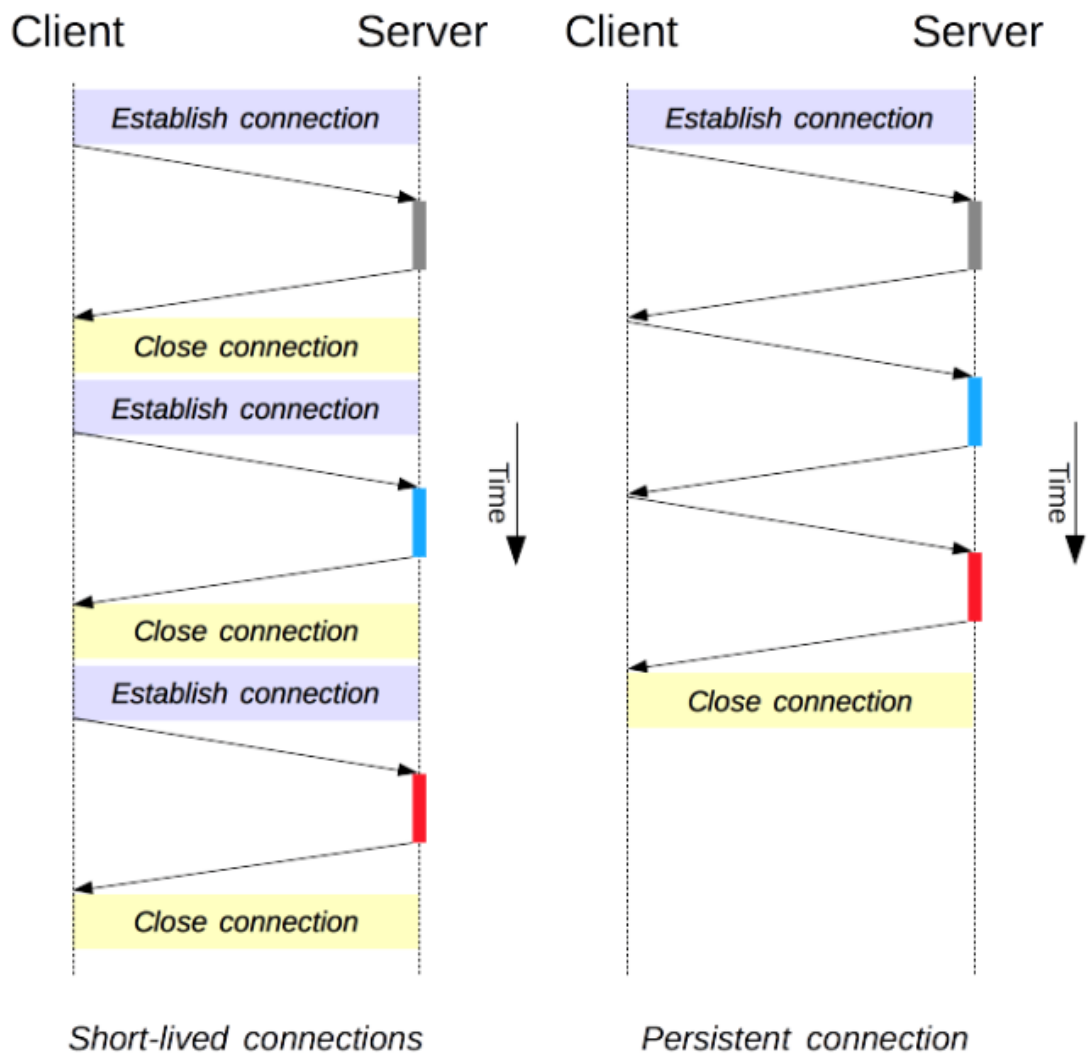
- Um servidor web é um computador que tem software e hardware em que vai permitir que os usuários solicitem páginas web;
- O computador servidor vai entregar essas páginas da web para os computadores que o solicitem;
- Ele também pode fazer outras transformações com as páginas da web, como cálculos e acessos a bancos de dados.

Camada OSI e TCP/IP



Características do protocolo HTTP

- **O HTTP usa o protocolo TCP como seu protocolo da camada de transporte**
- **Protocolo HTTP é um protocolo sem estado**
  - Se um determinado cliente solicitar o mesmo objeto duas vezes em um período de poucos segundos, o servidor não consegue responder dizendo que acabou de enviar o objeto.
- **O protocolo HTTP possui conexões persistentes e não persistentes.**
  - Cada par de requisição/resposta deve ser enviado por uma conexão TCP distinta ou todas as requisições e suas repostas devem ser enviadas por uma mesma conexão TCP?
- Conexões persistentes são usadas de forma padrão pelo HTTP, mas podem ser configuradas.


Client
Server
Establish connection
Close connection
Establish connection
Close connection
Establish connection
Close connection
Time
Short-lived connections
Client
Server
Establish connection
Close connection
Time
Persistent connection

Conexões não Persistentes

- A conexão TCP não persiste para cada par requisição/reposta

- Conexões TCP paralelas
  - Por padrão a maioria browsers abrem de 5 a 10 conexões paralelas
  - Reduzir o tempo de resposta

Conexões Persistentes

- Conexões TCP persistentes;
- Troca de informações de forma sequencial.
- Uma página HTML inteira (com todos os objetos) podem ser enviadas por uma única conexão TCP;
- A conexão fecha quando não está sendo utilizada a um certo tempo;
- O modo padrão do HTTP usa conexões persistentes com paralelismo.
- Possuem algumas desvantagens
  - Podem sobrecarregar o servidor (buffers TCP, variáveis no cliente e no servidor)

Pedido HTTP

- Um exemplo de pedido HTTP (é totalmente transparente para o usuário do browser):

```
GET /internet/index.html HTTP/1.0
User-agente: Mozilla /4.5 [en] (WinNT; I)
AcceptP:  text/plain, text/html, image/gif, image/x-xbitmap,
image/jpeg, image/pjpeg, image/png, */*
Accept-Charset: isso-8859-1, *, utf-8
Accept-Enconding: gzip
Accept-Language: en
```

Pedido HTTP

- Um pedido HTTP é composto de quatro partes básicas:
  - O método: ação a ser realizada.
  - A URI (Universal Resource Identifier): a informação requisitada.
  - A versão do protocolo HTTP: a mais utilizada a 1.1.
  - Informações adicionais, informações complementares às demais.

Pedido HTTP

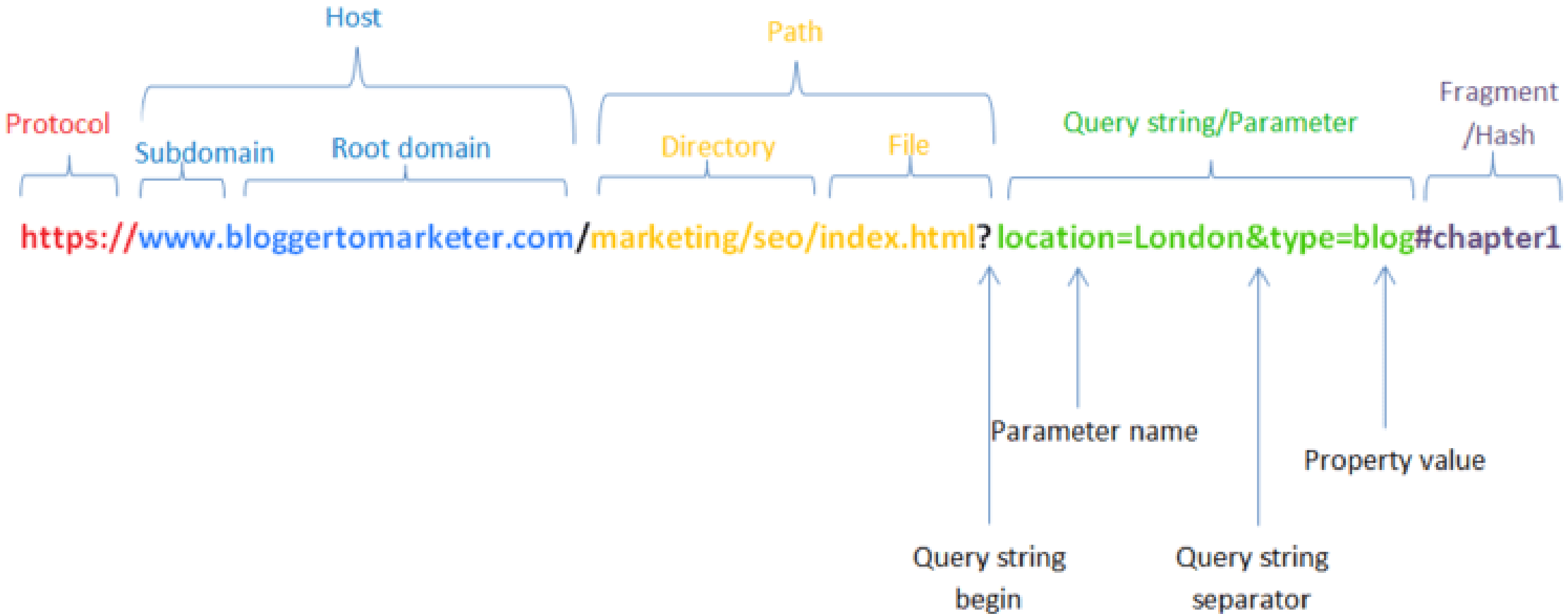
- **Método**
  - O método definido será aplicado no objeto (a informação requisitada) definido pela URI
- O método pode ser um entre vários tipos:

Método	**Descrição**
GET	Significa recuperar qualquer informação (na forma de uma entidade) identificada por um recurso.
HEAD	Idêntico ao GET, exceto que o servidor Não Deve retornar um corpo de mensagem de resposta
POST	Envia informações para o servidor Web. Método utilizado por scripts.
PUT	Envia uma cópia de um objeto/informação para ser armazenado num servidor Web.
DELETE	Apaga um objeto armazenado no servidor Web.
PATCH	Realizar alterações parciais em um recurso existente.

Pedido HTTP - URI

- O tipo de **URI** utilizada pelo protocolo HTTP é chamada de **URL** (Uniform Resourde Locator) e contém três partes:
  - A identificação do protocolo;
  - O endereço do computador servidor;
  - O documento requisitado (pode incluir subdiretórios).

- O Um bom exemplo de URL seria o documento index.html armazenado no diretório internet em um servidor de endereço www.unesa.com.br:
  - http://www.unsea.com.br/internet/index.html


Host
Path
Protocol
Subdomain
Root domain
Directory
File
Query string/Parameter
Fragment
/Hash
https://www.bloggertomarketer.com/marketing/seo/index.html?location=London&type=blog#chapter1
Parameter name
Property value
Query string
begin
Query string
separator

Pedido HTTP

- **Versão**
  - Ao enviar o pedido HTTP, o browser informa ao servidor qual versão do protocolo HTTP ele suporta.
  - Enquanto a versão HTTP 2.0 é mais atual, muitos browsers e servidores Web ainda utilizam a versão 1.1.

Pedido HTTP

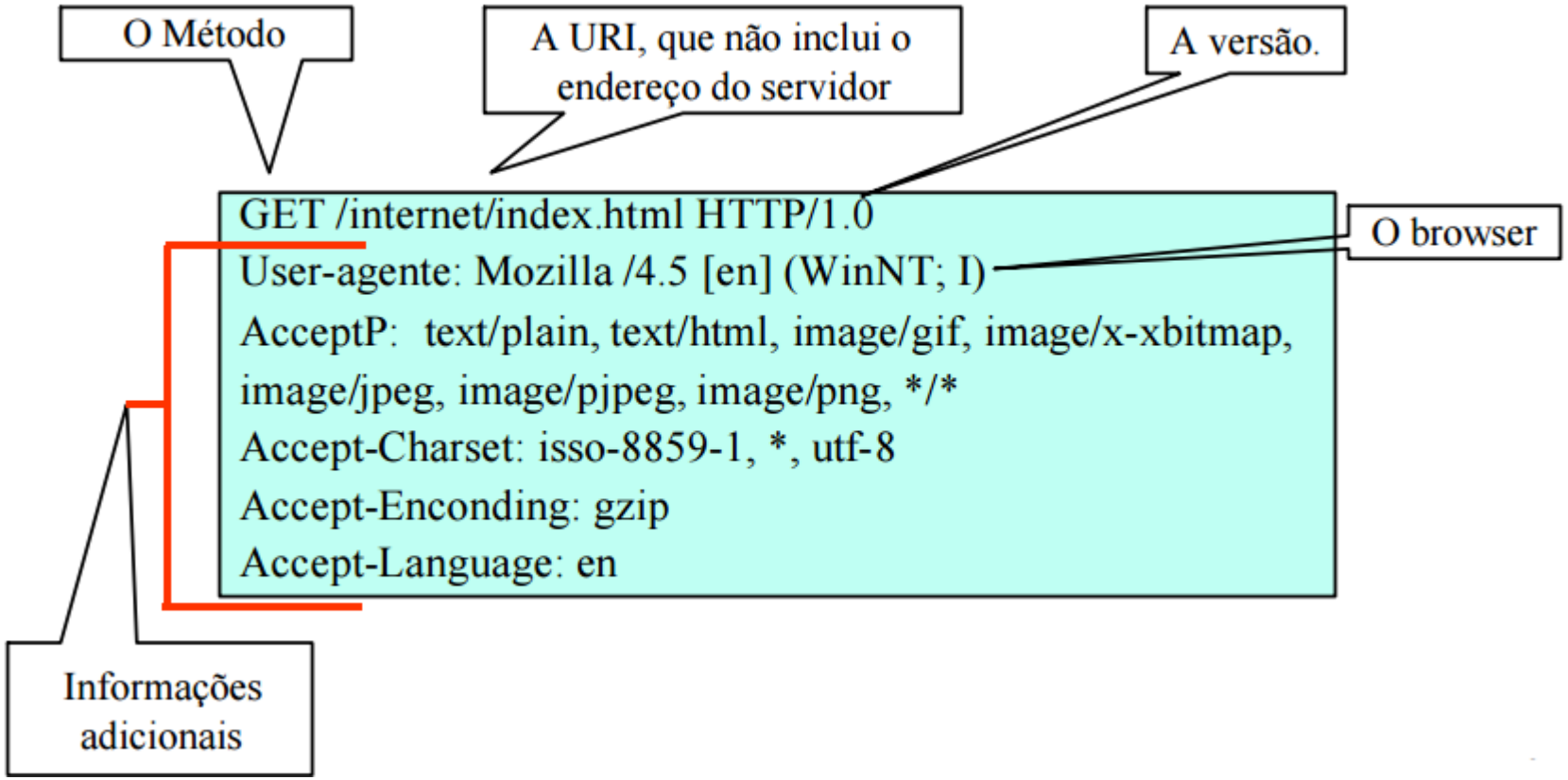
- **Informações Adicionais**
  - São maneiras do navegador informar ao servidor Web algumas preferências definidas na configuração do navegador como:
    - Tipo de documento aceitos.
    - Linguagem preferida para os documentos HTML que são retornados.
    - Set de caracteres suportados.
    - etc

Informações Adicionais

Cabeçalho	Funcionalidade
Accept	Especifica tipos de mídia/conteúdo aceitos como resposta. Podemos escolher JSON, XML, YAML etc. (text/html, application/json, */*)
Accept-Language	Especifica quais idiomas são aceitos como resposta. (pt-br, en-US, fr-CA)
Accept-Encodding	Especifica como a resposta pode ser codificada. O Cliente pode definir se deseja a resposta em texto plano ou codificada em algum algoritmo de compressão. (gzip, compress, deflate)
Accept-Charset	Especifica quais charsets são aceitos como resposta. (utf-8, iso-8859-1)

Pedido HTTP completo

- Exemplo de pedido HTTP completo:



Resposta HTTP

- O servidor Web ao receber o pedido, processa-o de modo a determinar o que deverá ser feito.
- Em relação ao pedido do slide anterior, o servidor Web deverá procurar o arquivo **index.html** no diretório internet e retorna-lo ao browser.

Resposta HTTP

- Um exemplo de resposta HTTP (é totalmente transparente para o usuário do browser):

```
HTTP/1.0 200 Document follows
Date: Thu, 20 Aug 1998 18:47:27 GMT
Server: NCSA/1.5.1
Content-type: text/html
Last-modified: Fri, 14 Aug 1998 20:14:23 GMT
Content-length:5807

<html>
<head><title> Navegando na Internet</title></head>
<body>
```

Resposta HTTP

- Uma resposta HTTP é formada por três elementos:
  - Linha de status indicando sucesso ou falha do pedido.
  - Descrição da informação contida na resposta (Metainformação/MIME).
  - A própria informação - que foi requisitada

Resposta HTTP

- **Status**
  - A linha de status traz as seguintes informações:
    - A versão do protocolo HTTP;
    - O código de status que define o resultado do pedido;
    - Uma pequena frase explicando o que significa o código.

Família de erros HTTP

- A família de erros HTTP é composta por códigos de status de três dígitos que são usados para comunicar o resultado de uma solicitação entre um cliente e um servidor na web.
  - 1xx: Respostas informativas
  - 2xx: Respostas de sucesso
  - 3xx: Respostas de redirecionamento
  - 4xx: Erros do cliente
  - 5xx: Erros do servidor

Códigos de Status

Código	Descrição
200	(Ok) - pedido bem sucedido. A informação requisitada será retornada.
201	(Created) - requisição foi bem-sucedida e resultou na criação de um novo recurso.
204	(No contente) - requisição foi bem-sucedida e o servidor processou a requisição, mas não está retornando nenhum conteúdo no corpo da resposta
400	(Bad Request) Servidor não conseguiu entender ou processar a solicitação do cliente devido a dados inválidos ou malformados na solicitação
401	(Unautorized) - a informação requisitada é de acesso restrito, sendo necessário se autenticar.
403	(Forbidden) - acesso proibido.
404	(Not found) a informação requisitada não foi encontrada ou teve permissão de acesso negada. A primeira opção é muito frequente na Internet e pode ocorrer por erro de digitação de uma URL.
422	(Unprocessable Entity) servidor entende o tipo de conteúdo da entidade da requisição e a sintaxe da requisição está correta, mas não pôde processar as instruções contidas
500	(Server Error) - erro no servidor Web. Comum quando da execução de scripts.

https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status

Resposta HTTP

- **Descrição da Informação**
  - Uma das partes mais importantes de uma resposta HTTP é a informação que define o tipo de documento que está sendo retornado ao browser, de modo que ele possa exibi-lo adequadamente.
  - Essa informação é codificada pelo tipo MIME (Multi Purpose Mail Extensions).
  - Em outras palavras, o servidor Web, ao receber o pedido, procura o arquivo e checa a sua extensão (.html), realizando uma consulta em uma grande tabela de tipos MIME que indica o código que deverá ser usado para cada extensão existente.
  - No caso de arquivos .html/htm, o tipo MIME é o text/html.

MIME Code

Cabeçalho	Descrição
text/plain	Arquivo no formato texto (ASCII);
text/html	Documento no formato HTML, o padrão para documentos Web;
application/zip **image/gif**	Arquivo compactado;
image/gif	Imagem codificada no formato GIF
image/jpeg	Imagem codificada no formato JPEG
application/json	Arquivo json hash

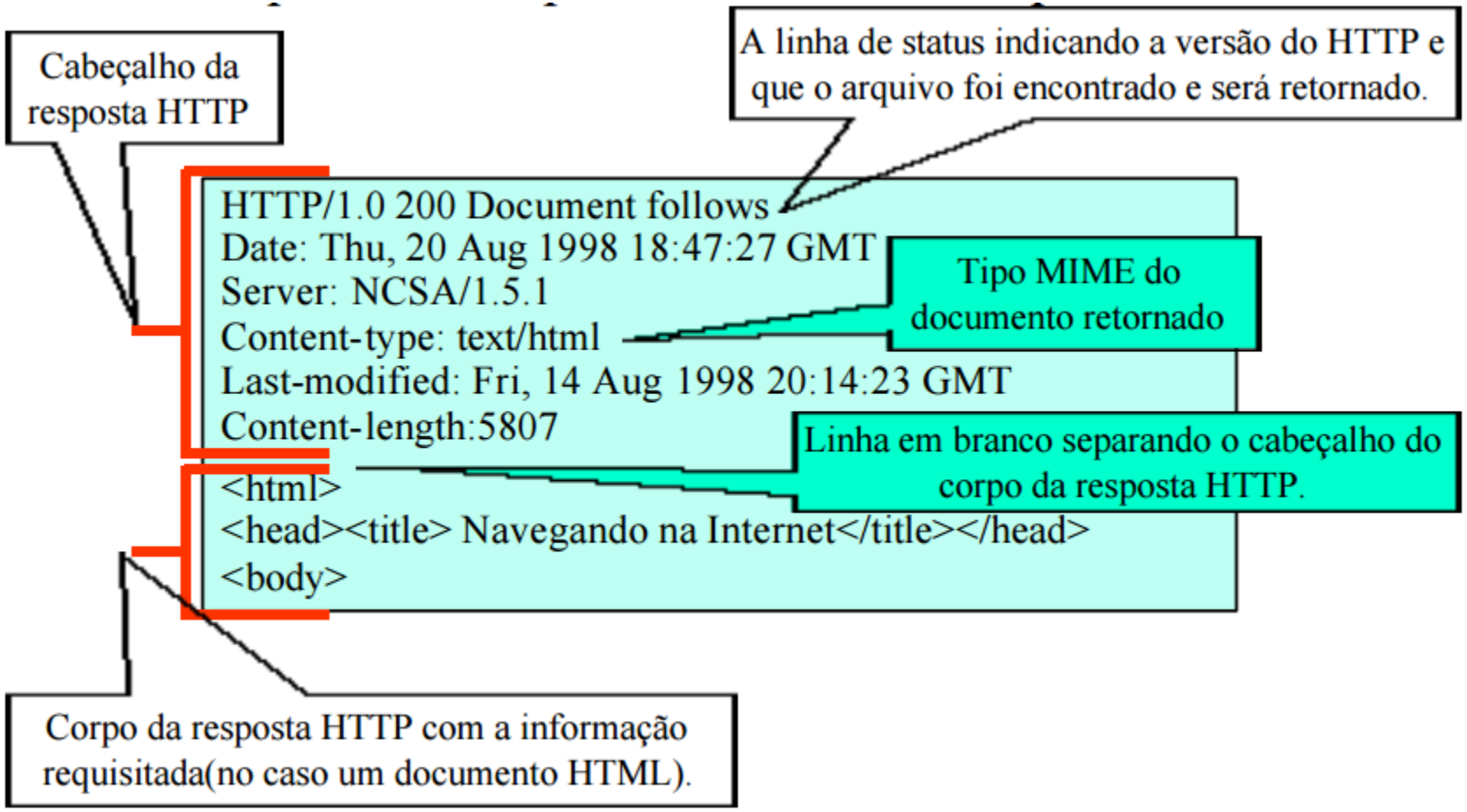
Resposta HTTP

- Descrição da Informação
  - Tamanho em bytes
  - Última data de atualização
  - ...

Resposta HTTP

- A última parte de uma resposta HTTP é sempre a informação que foi requisitada, que pode ser um documento HTML, uma imagem GIF, um arquivo javascript ou um arquivo de som.
- Essa informação, independente do tipo, é sempre em formato ASCII (texto puro).

Resposta HTTP

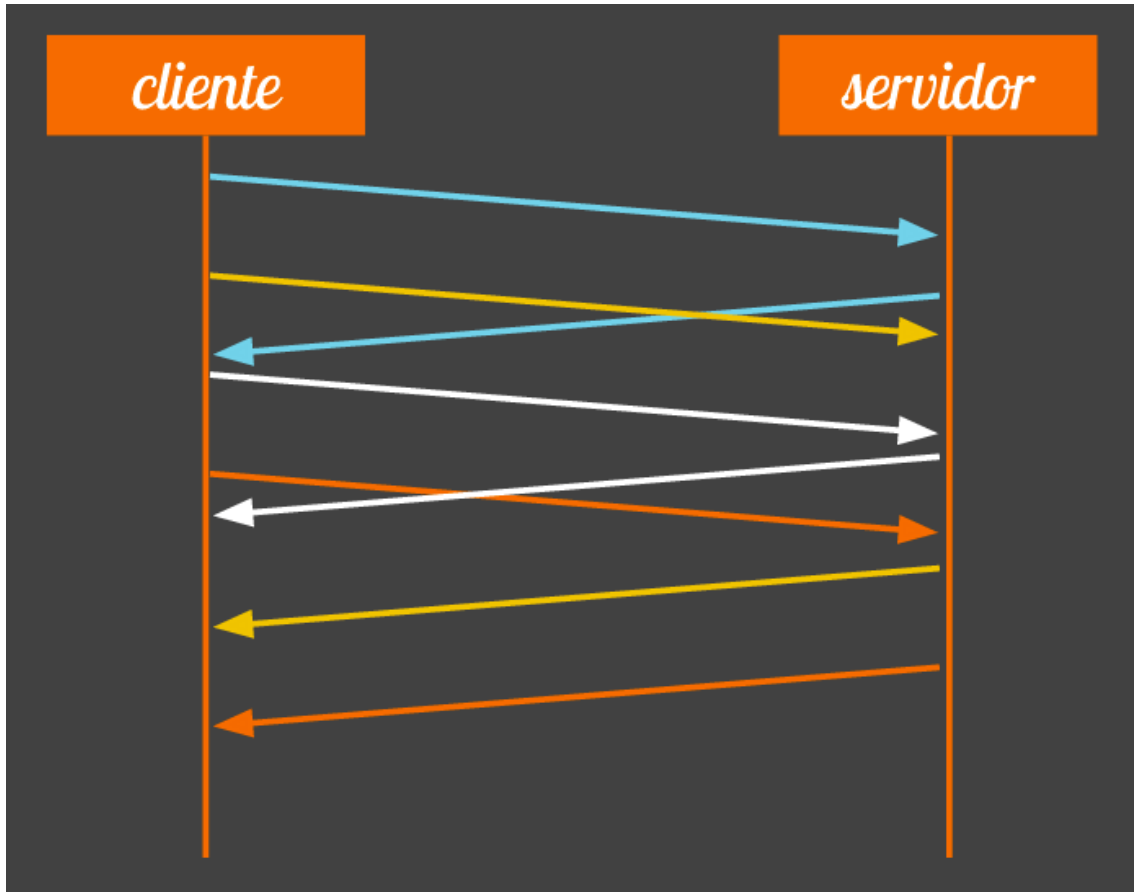


HTTP 2.0 ?

Http 2.0

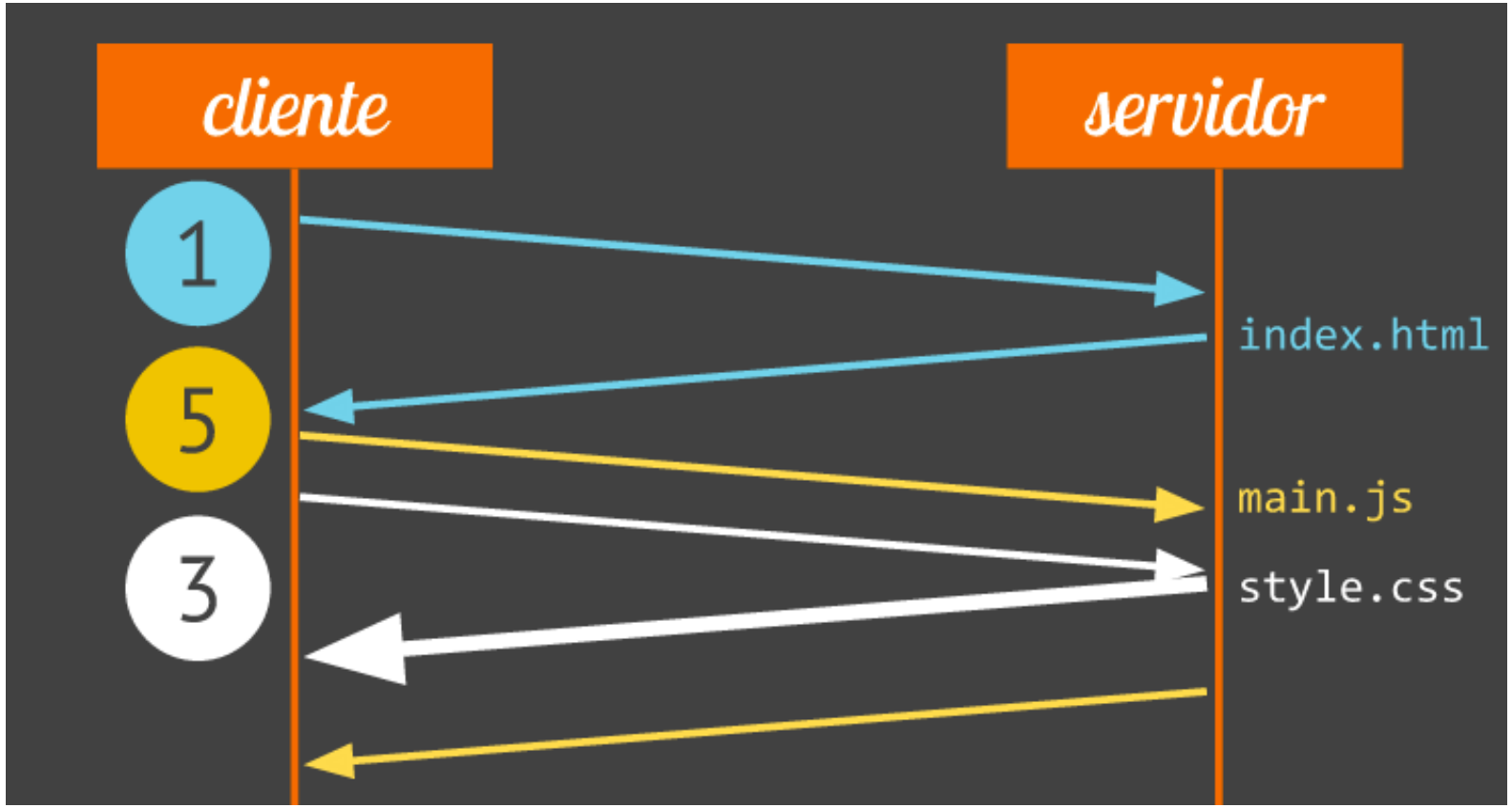
- **Vantagens**
  - Multiplexação de mensagens
  - Server Push
  - Compressão de cabeçalho (HPACK)
  - Priorização de requisições
  - Obrigação na utilização de SSL (HTTPS)

Multiplexação de Mensagens



No HTTP 2.0, as requisições e respostas são paralelas automaticamente em uma única conexão. É o chamado multiplexing que deixa que façamos vários requests ao mesmo tempo e recebamos as respostas de volta conforme forem ficando prontas, tudo paralelo e assíncrono.

Priorização de Requests



Server Push