

AULA 04 – 0

PROTOCOLO HTTP

Por Sediane Carmem Lunardi Hernandes

1

AGENDA



Introdução



Tipos de
conexões



Formato das
mensagens



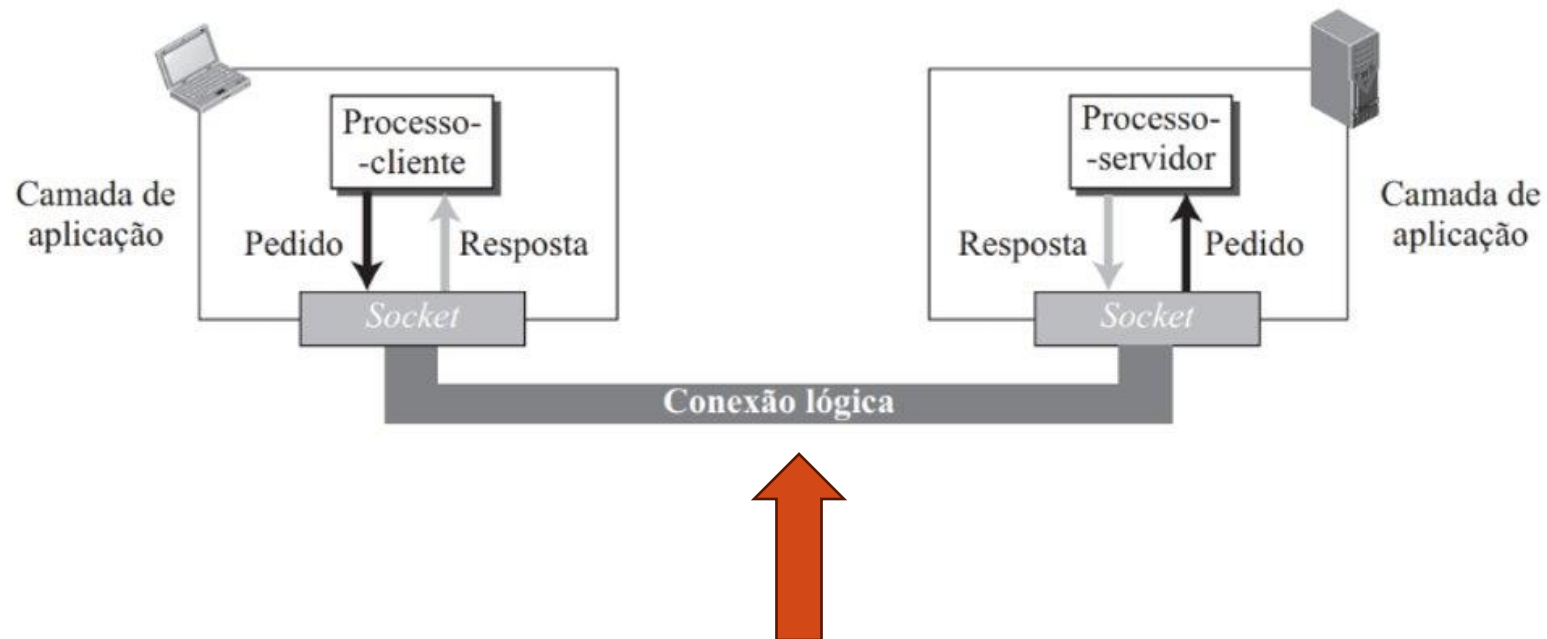
Diferentes versões
do protocolo HTTP

1. INTRODUÇÃO

- O Protocolo de Transferência de Hipertexto (HTTP – *HyperText Transfer Protocol*) é um protocolo usado para **definir COMO** os **programas cliente-servidor podem ser escritos para recuperar páginas Web**
 1. Um cliente HTTP envia uma solicitação
 - o cliente usa um número de porta temporário
 2. Um servidor HTTP retorna uma resposta
 - o servidor usa o número de porta 80

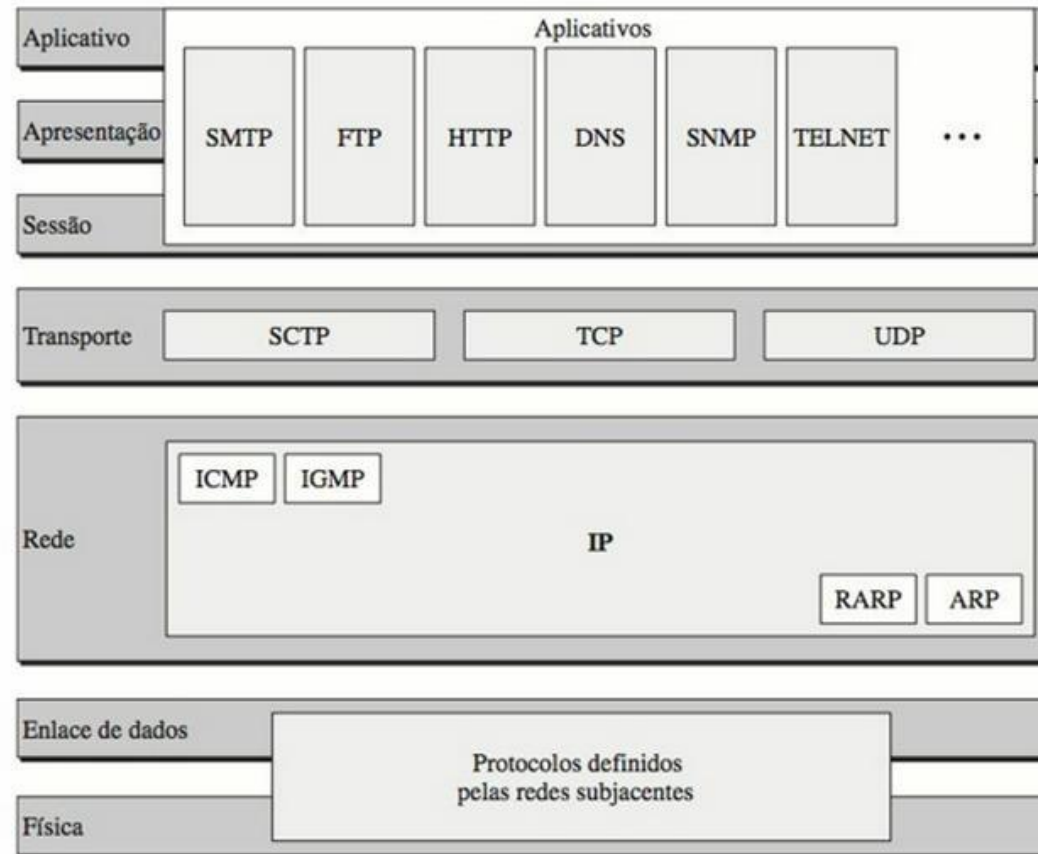
O HTTP usa os serviços do TCP que é um protocolo orientado à conexão e confiável. Isto significa que, antes que ocorra qualquer transação entre o cliente e o servidor, uma conexão precisa ser estabelecida entre eles. Após a transação ser efetuada, a conexão deve ser encerrada.

1. INTRODUÇÃO (CONT.)

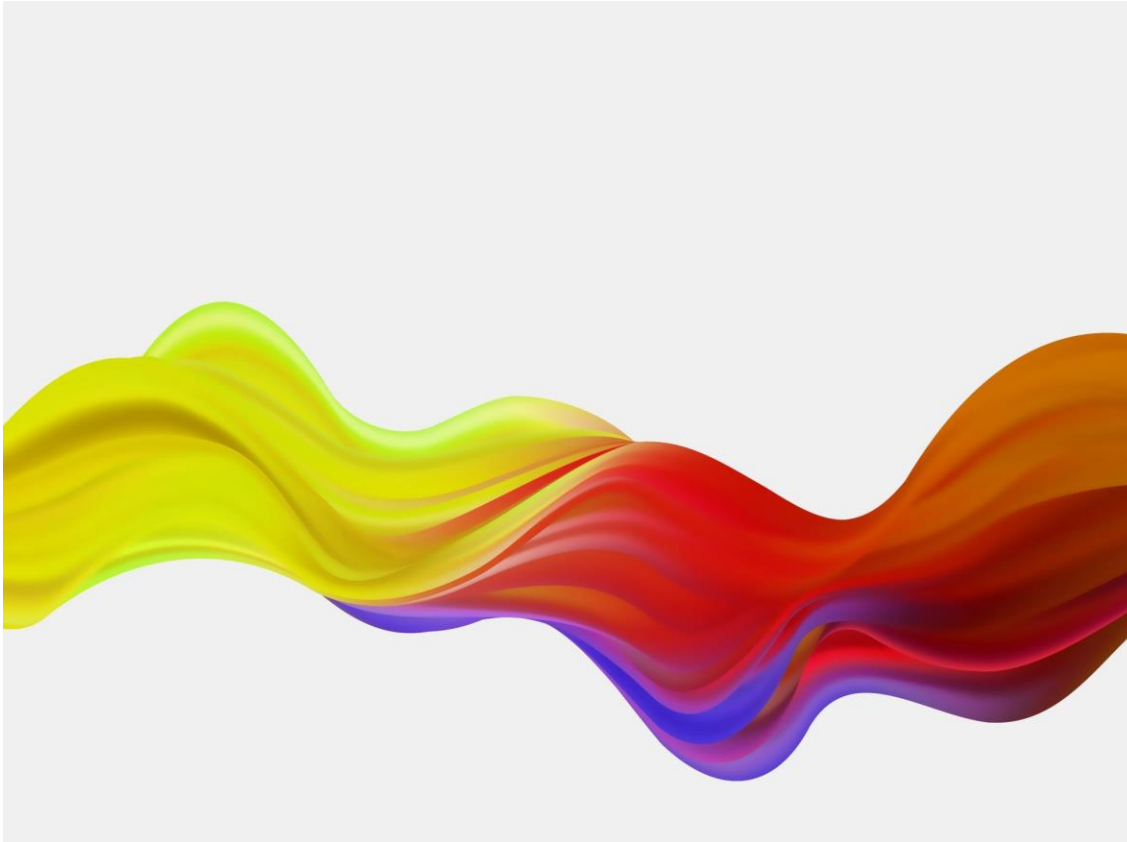


1. INTRODUÇÃO (CONT.)

Modelo OSI x TCP/IP



(FOROUZAN, 2010, p. 60)



SERVIDOR TCP

(SLIDES)

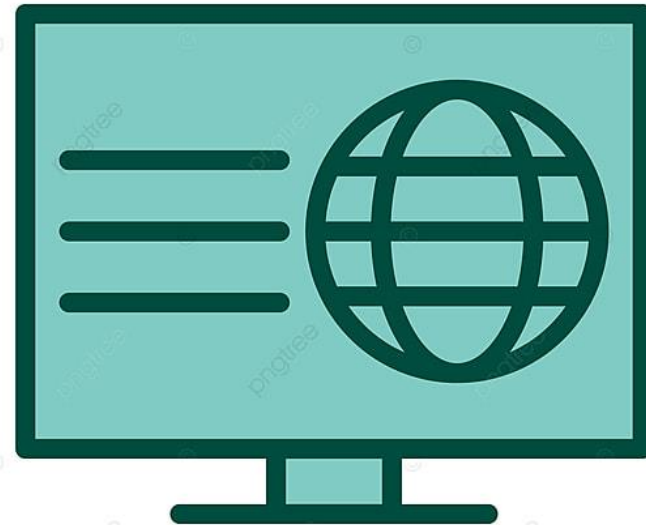
Por Sediane Carmem
Lunardi Hernandez

RELEMBRANDO...

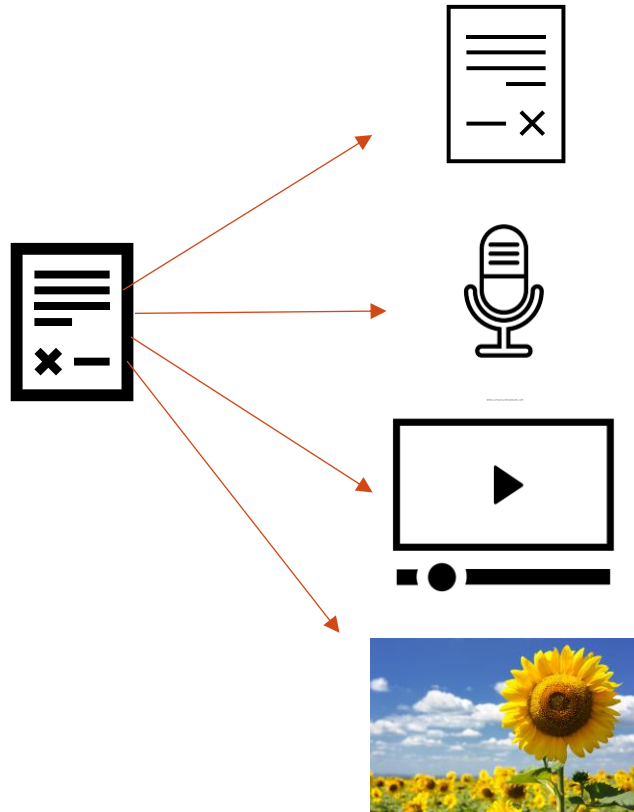
- O que é uma página Web e o que ela tem?
 - Uma página web é um documento que pode conter um texto, uma imagem, um arquivo de áudio ou um arquivo de vídeo que pode ser **vinculado** a outra(s) página(s) web



Vinculação permite que uma página Web faça referência a outra página armazenada em um servidor em algum outro lugar no mundo



RELEMBRANDO...



- **Vinculação** de páginas Web é obtida usando um conceito chamado **hipertexto**
 - Ideia inicial: uma máquina recuperaria outro documento armazenado no sistema quando um vínculo (link) aparecesse no documento
 - Implementação da **ideia** realizada pela Web
 - permitiu que o documento vinculado fosse recuperado quando o link fosse clicado pelo usuário
- Termo atual mudou de **hipertexto** para **hipermídia**

RELEMBRANDO...

- Atualmente, a Web vai além da simples recuperação de documentos vinculados
 - Utilizada para
 - prover serviços de compras eletrônicas e jogos
 - ouvir programas de rádio
 - ver programas de televisão

1. INTRODUÇÃO

- Conceito de **hipertexto** incorporado em páginas Web pode exigir vários pedidos e respostas
 - Para isso o protocolo HTTP pode precisar de uma ou várias conexões TCP

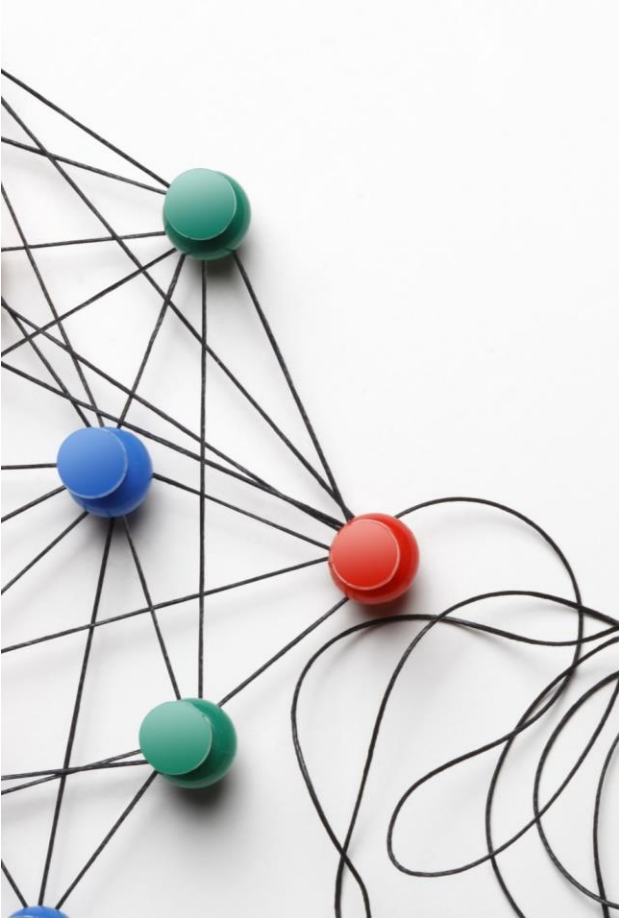


TIPOS DE CONEXÕES (HTTP SOBRE TCP)

- O conceito de hipertexto incorporado em páginas Web pode exigir vários pedidos e resposta
 - Se as páginas Web (objetos a serem recuperados) estiverem localizadas em servidores diferentes
 - criar uma nova conexão TCP para recuperar cada objeto
 - Senão
 - se alguns dos objetos estão localizados no mesmo servidor, tem-se duas opções
 1. recuperar cada objeto usando uma nova conexão TCP (conexão não persistente); ou
 2. criar uma ÚNICA conexão TCP (conexão persistente) e recuperar todos os objetos.
- Logo, as conexões podem ser persistentes e não persistentes



2. TIPOS DE CONEXÕES



2. TIPOS DE CONEXÕES (CONT.)

▪ Não persistentes:

- Uma conexão TCP é criada para cada pedido/resposta
- Os passos desta estratégia são listados a seguir:
 - 1. O cliente abre uma conexão TCP e envia uma solicitação.
 - 2. O servidor envia a resposta e fecha a conexão.
 - 3. O cliente lê os dados até encontrar um marcador de fim de arquivo; ele, então, fecha a conexão.

▪ Persistentes:

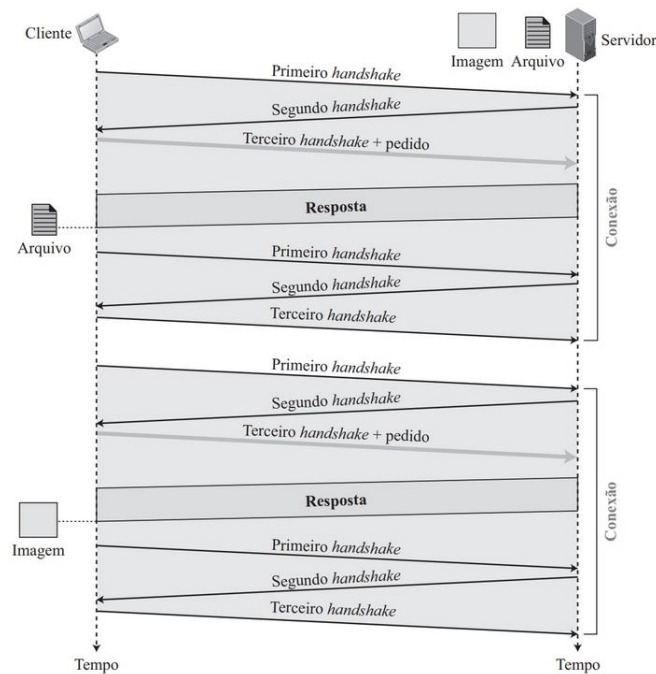
- Servidor deixa a conexão aberta para outras solicitações após o envio de uma resposta
 - o servidor pode fechar a conexão a pedido de um cliente ou se um tempo limite for atingido

O HTTP versão 1.1 especifica como padrão o uso de uma conexão persistente

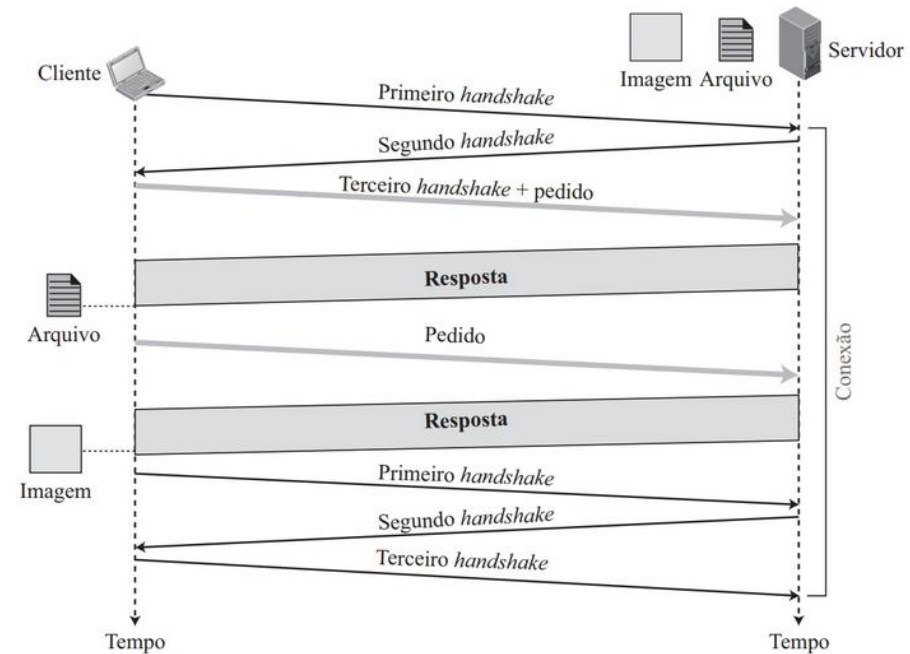
$N\text{-links} = N\text{-conexões} + 1 \text{ abertas e fechadas}$

TIPOS DE CONEXÕES (HTTP SOBRE TCP)

Conexão não persistente



Conexão persistente



3. FORMATO DAS MENSAGENS

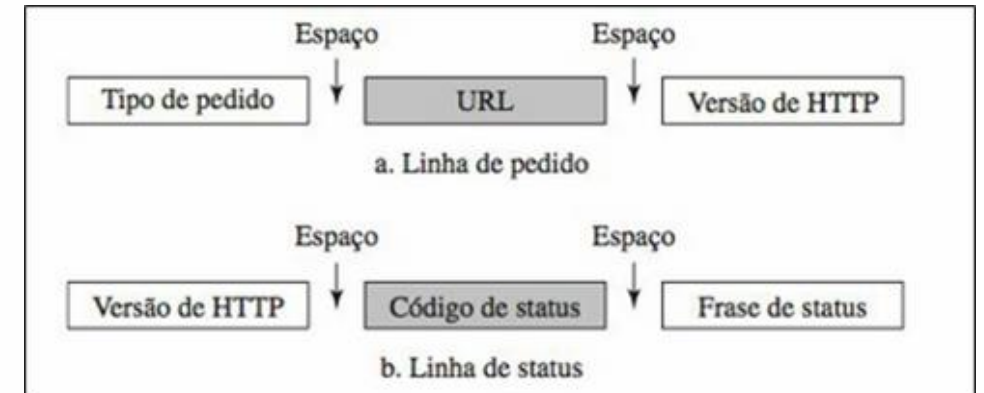
- O protocolo HTTP define o formato das mensagens de pedido e de resposta



Mensagem de pedido

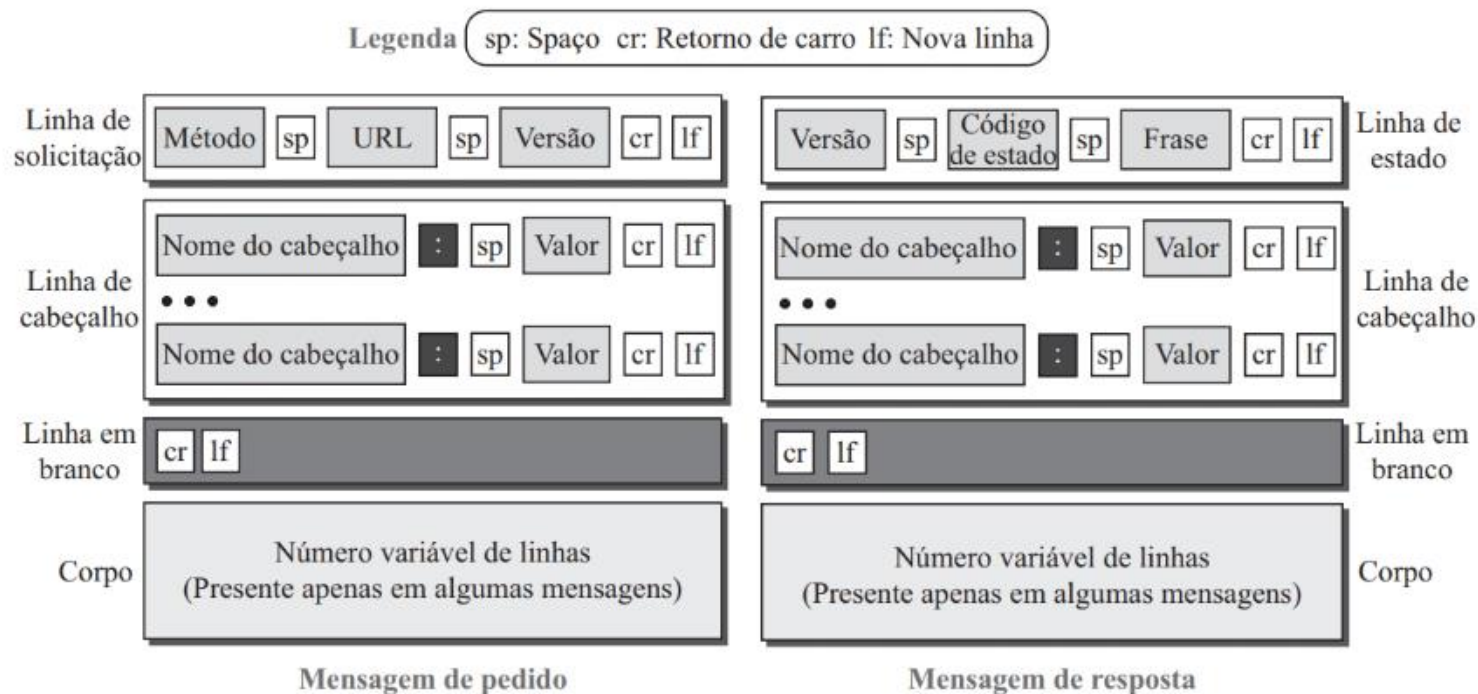


Mensagem de resposta



3. FORMATO DAS MENSAGENS (CONT.)

- Ou,





MENSAGEM DE PEDIDO

3. FORMATO DAS MENSAGENS (CONT.)

- **Mensagem de pedido:**

- 1a. Linha: **Linha de solicitação**

- Existem três campos nessa linha separados por um espaço e terminados por dois caracteres – retorno de carro (*carriage return*) e nova linha (*line feed*). Os campos são chamados **método**, **URL** e **versão**.

- **Primeiro campo:**

- **Método** - define os tipos de solicitação:

Método	Ação
GET	Solicita um documento ao servidor
HEAD	Solicita informações sobre um documento, mas não o documento em si
PUT	Envia um documento do cliente para o servidor
POST	Envia alguma informação do cliente para o servidor
TRACE	Ecoa a solicitação recebida
DELETE	Remove a página Web
CONNECT	Reservado
OPTIONS	Consulta opções disponíveis

3. FORMATO DAS MENSAGENS (CONT.)

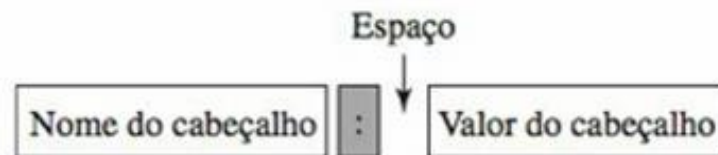
- **Mensagem de pedido (cont.):**

- 1a. Linha:...
- Primeiro campo: ...
- **Segundo campo:**
 - **URL** - define o endereço e o nome da página Web correspondente
- **Terceiro campo:**
 - **Versão** - mostra a versão do protocolo

3. FORMATO DAS MENSAGENS (CONT.)

▪ Mensagem de pedido (cont.):

- 2a. Linha: **Linhas de cabeçalho de solicitação**: zero ou mais
 - Cada linha de cabeçalho envia informações adicionais **do cliente para o servidor**.
 - Por exemplo, o cliente pode solicitar que o documento seja enviado em um formato especial
 - Cada linha de cabeçalho tem um **nome de cabeçalho**, um caractere de dois pontos (:), um espaço e um valor de cabeçalho
 - O campo valor define os valores associados a cada nome do cabeçalho.
 - A lista de valores pode ser encontrada nas RFCs correspondentes.



3. FORMATO DAS MENSAGENS (CONT.)

<i>Cabeçalho</i>	<i>Descrição</i>
Accept	Mostra o formato de mídia que o cliente pode aceitar.
Accept-charset	Mostra o conjunto de caracteres que o cliente pode manipular.
Accept-encoding	Mostra o esquema de codificação que o cliente pode manipular.
Accept-language	Mostra a linguagem que o cliente pode aceitar.
Authorization	Mostra quais permissões o cliente tem.
From	Mostra o endereço de e-mail do usuário.
Host	Mostra o host e o número de porta do cliente.
If-modified-since	Envia o documento se for mais recente do que a data especificada.
If-match	Envia o documento somente se ele corresponder à tag dada.
If-non-match	Envia o documento somente se ele não corresponder à tag dada.
If-range	Envia apenas a parte do documento que está faltando.
If-unmodified-since	Envia o documento se não tiver sido alterado desde a data especificada.
Referrer	Especifica o URL do documento vinculado.
User-agent	Identifica o programa cliente.

- **Mensagem de pedido (cont.):**
 - 2a. Linha: **Linhas de cabeçalho de solicitação (cont.):**
 - **Cabeçalho de pedido:**

3. FORMATO DAS MENSAGENS (CONT.)

<i>Cabeçalho</i>	<i>Descrição</i>
Cache-control	Especifica informações sobre o uso de cache.
Connection	Mostra se a conexão deve ser fechada ou não.
Date	Mostra a data atual.
MIME-version	Mostra a versão de MIME usada.
Upgrade	Especifica o protocolo de comunicação preferencial.

- **Cabeçalho geral**

3. FORMATO DAS MENSAGENS (CONT.)

- **Mensagem de pedido (cont.):**

- 3a. Linha: **Linha em branco**
- 4a. Linha: **Corpo:** O corpo pode estar presente em uma mensagem de solicitação, e costuma conter o comentário a ser enviado ou o arquivo a ser publicado no site quando o método é PUT ou POST.



MENSAGEM DE RESPOSTA

3. FORMATOS DAS MENSAGENS (CONT.)

■ Mensagem de resposta:

- A mensagem de resposta consiste em:
 - 1ª linha: **Linha de estado**:
 - existem três campos nessa linha separados por espaços e terminados por um retorno de carry e uma nova de linha.
 - Campo 1: versão do protocolo HTTP
 - Campo 2: **código de estado** (estado do pedido – informativo (intervalo 100); sucesso da solicitação (intervalo 200); redirecionamento para outra URL (intervalo 300); erro do lado cliente (intervalo 400); erro no lado servidor (intervalo 500))
 - Campo 3: frase de estado (explica o código do estado em formato de texto)

CÓDIGO DE ESTADO

<i>Código</i>	<i>Frase</i>	<i>Descrição</i>
Informativos		
100	Continue	A parte inicial do pedido foi recebida e o cliente pode continuar com seu pedido.
101	Trocando	O servidor está consentindo em um pedido do cliente para trocar os protocolos definidos no cabeçalho de atualização.

<i>Código</i>	<i>Frase</i>	<i>Descrição</i>
Sucesso		
200	OK	O pedido é bem-sucedido.
201	Criado	Um novo URL é criado.
202	Aceito	O pedido é aceito, mas não é posto em prática imediatamente.
204	Nenhum conteúdo	Não há conteúdo no corpo.

CÓDIGO DE ESTADO

<i>Código</i>	<i>Frase</i>	<i>Descrição</i>
Redirecionamento		
301	Movido permanentemente	O URL solicitado não é mais usado pelo servidor.
302	Movido temporariamente	O URL solicitado foi movido temporariamente.
304	Não modificado	O documento não foi modificado.

<i>Código</i>	<i>Frase</i>	<i>Descrição</i>
Erro de cliente		
400	Pedido inválido	Há um erro de sintaxe no pedido.
401	Não autorizado	O pedido não tem autorização correta.
403	Proibido	O serviço é negado.
404	Não encontrado	O documento não é encontrado.
405	Método não permitido	O método não é suportado nesse URL.
406	Não aceitável	O formato solicitado não é aceitável.

CÓDIGO DE ESTADO

<i>Código</i>	<i>Frase</i>	<i>Descrição</i>
Erro de servidor		
500	Erro interno de servidor	Existe um erro, como uma falha, nas instalações do servidor.
501	Não implementado	A ação solicitada não pode ser executada.
503	Serviço indisponível	O serviço está temporariamente indisponível, mas poderá ser solicitado no futuro.

3. FORMATOS DAS MENSAGENS (CONT.)

- **Mensagem de resposta:**

- A mensagem de resposta consiste em:

- 1a linha: **Linha de estado:**

-

- 2a linha: **Linhas de cabeçalho (0 ou mais linhas):**

- envia informações adicionais do servidor para o cliente

- cada linha de cabeçalho tem um nome de cabeçalho, um caractere de dois pontos (:), um espaço, e um valor do cabeçalho

3. FORMATOS DAS MENSAGENS (CONT.)

- **Mensagem de resposta:**

- ...

- Linhas de cabeçalho (0 ou mais linhas):

- ...

<i>Cabeçalho</i>	<i>Descrição</i>
Accept-range	Mostra se o servidor aceita o intervalo solicitado pelo cliente.
Age	Mostra a idade do documento.
Public	Mostra a lista de métodos suportados.
Retry-after	Especifica a data após a qual o servidor está disponível.
Server	Mostra o nome do servidor e o número da versão.

3. FORMATOS DAS MENSAGENS (CONT.)

- **Mensagem de resposta:**

- ...

- Linhas de cabeçalho (0 ou mais linhas):

- ...

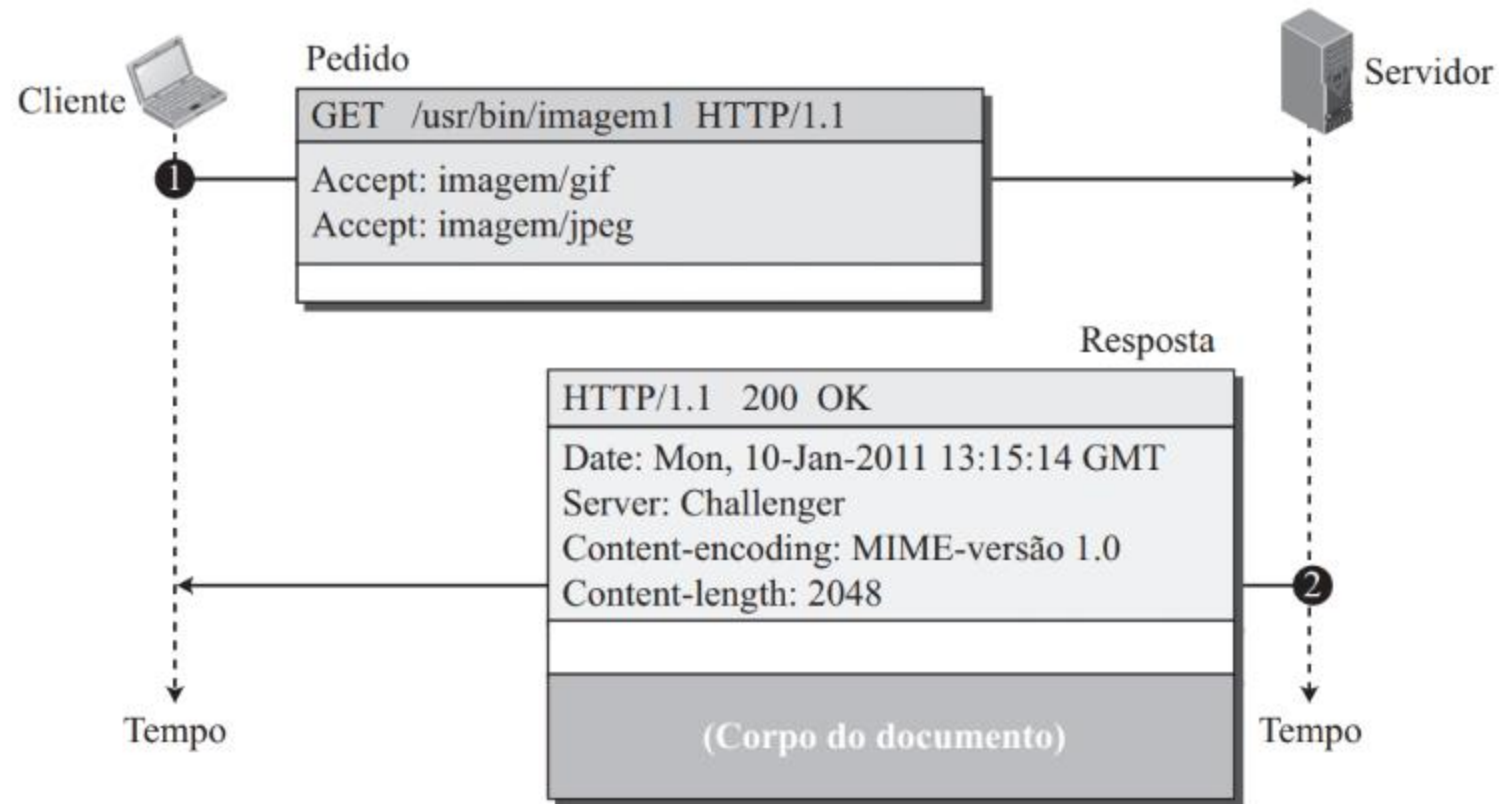
<i>Cabeçalho</i>	<i>Descrição</i>
Allow	Lista os métodos válidos que podem ser usados com um URL.
Content-encoding	Especifica o esquema de codificação.
Content-language	Especifica a linguagem.
Content-length	Mostra o comprimento do documento.
Content-range	Especifica o intervalo do documento.
Content-type	Especifica o tipo de mídia.
Etag	Fornece uma tag de entidade.
Expires	Fornece a data e a hora em que o conteúdo poderá mudar.
Last-modified	Fornece a data e a hora da última alteração.
Location	Especifica a localização do documento criado ou movido.

3. FORMATOS DAS MENSAGENS (CONT.)

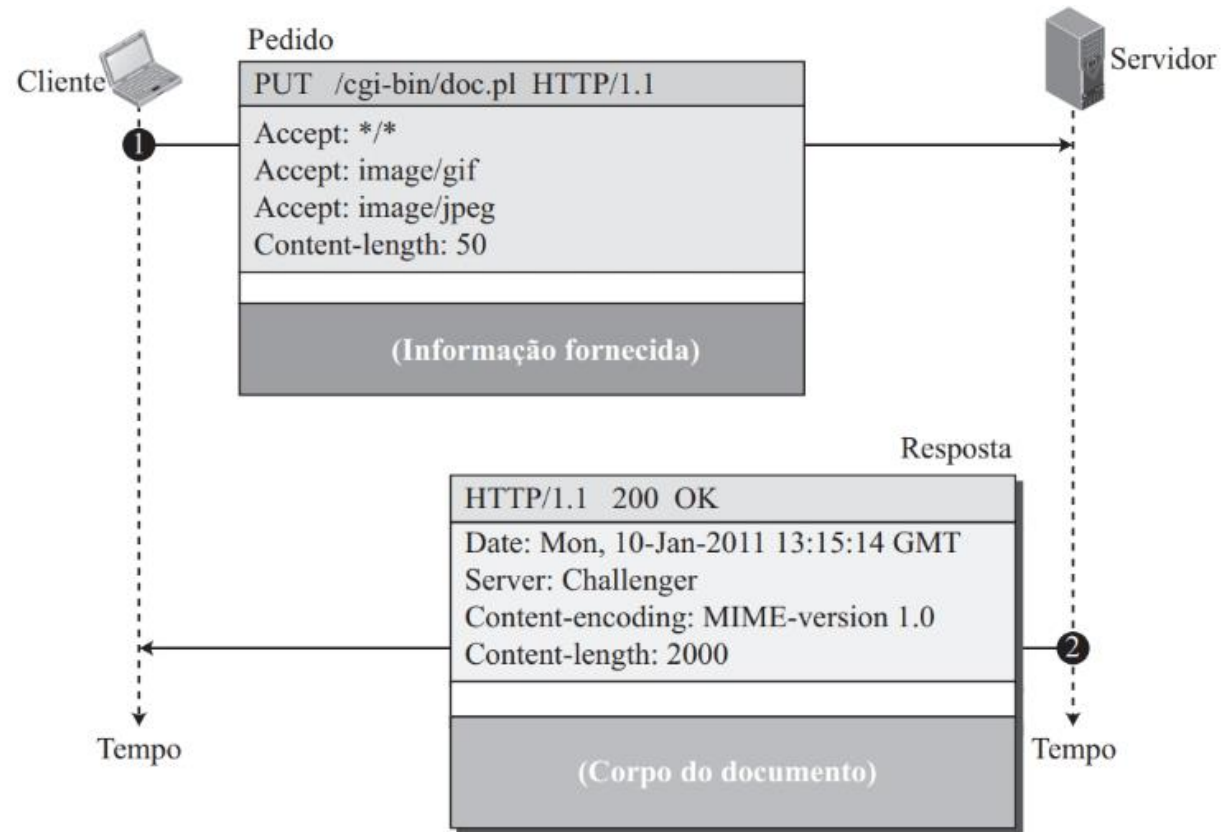
■ Mensagem de resposta:

- A mensagem de resposta consiste em:
 - 1a linha: **Linha de estado:** ...
 - 2a linha: **Linhas de cabeçalho (0 ou mais linhas):**...
 - 3a linha: **Linha em branco (uma)**
 - 4a linha: **Corpo (as vezes):**
 - Contém o documento a ser enviado pelo servidor para o cliente
 - está sempre presente, a não ser que a resposta seja uma mensagem de erro

3. FORMATOS DAS MENSAGENS (CONT.)



3. FORMATOS DAS MENSAGENS (CONT.)



PEDIDO CONDICIONAL

- Um cliente pode adicionar uma condição em seu pedido
 - O servidor enviará a página Web solicitada se a condição for satisfeita, ou informará o cliente caso contrário
 - uma das condições mais comuns impostas pelo cliente refere-se à data e hora em que a página Web foi modificada
 - o cliente pode enviar a linha de cabeçalho *If-Modified-Since* com o pedido para notificar ao servidor que precisa da página apenas se ela foi modificada após um certo ponto

```
GET http://www.commonServer.com/information/arquivo1 HTTP/1.1
```

```
If-Modified-Since: Thu, Sept 04 00:00:00 GMT
```

Linha de solicitação **Pedido**

Linha de cabeçalho

Linha em branco

```
HTTP/1.1 304 Not Modified
```

```
Date: Sat, Sept 06 08 16:22:46 GMT
```

```
Server: commonServer.com
```

```
(Corpo vazio)
```

Linha de estado

Primeira linha de cabeçalho

Segunda linha de cabeçalho

Linha em branco

Corpo vazio

Resposta

4. VERSÕES DO PROTOCOLO HTTP

- O protocolo HTTP possui as seguintes versões:



4. VERSÕES DO PROTOCOLO HTTP (CONT.)

■ HTTP/0.9

- A primeira versão foi proposta por Tim Berners-Lee
 - simplicidade era o ponto principal
 - transferência de dados era realizada no formato de texto **ASCII**
 - o único método de requisição: GET
- O surgimento da *World Wide Web* está interligado ao surgimento do protocolo HTTP

■ HTTP/1.0

- A comunicação e transferência de arquivos precisavam ir além da simples transferência de textos
 - era preciso também considerar dados mais ricos, metadados da requisição / resposta e negociação de conteúdo

■ HTTP/1.1

- Versão considerada um marco
 - Define o padrão da Internet
- Inseriu melhorias críticas de performance
 - mecanismo adicional de cache, transferências de codificação fragmentadas, pipelining de solicitação, codificações de transferência

4. VERSÕES DO PROTOCOLO HTTP (CONT.)

■ HTTP/2

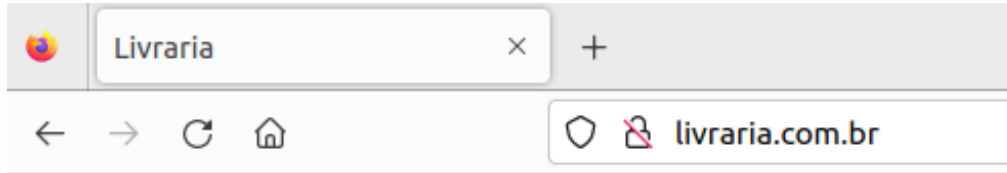
- O HTTP/2 é a evolução do protocolo HTTP
 - se concentra em fazer uso mais eficiente dos recursos de rede
 - não altera os fundamentos do HTTP, ou seja, sua semântica
 - ainda há solicitações, respostas, cabeçalhos, entre outros
- O HTTP/2 é um protocolo binário, não mais baseado em texto como o HTTP 1.1
- Um quadro (frame) é a menor unidade de comunicação dentro de uma conexão HTTP/2
 - Possui:
 - um cabeçalho, e;
 - uma sequência de octetos de comprimento variável estruturados de acordo com o tipo de quadro
- Um fluxo é um fluxo bidirecional de quadros na conexão HTTP/2
 - conceito correspondente no HTTP 1.1 é uma troca de mensagens de solicitação/resposta
- O HTTP/2 é capaz de executar vários fluxos de dados na mesma conexão TCP
 - evita o clássico cabeçote HTTP 1.1 de bloquear solicitações lentas e evitar reinstanciar conexões TCP para cada solicitação/resposta

MÃOS A OBRA

- Pesquisa:

1. Você deve fazer uma pesquisa nas RFCs das diferentes versões do protocolo HTTP/1.1 (RFC 2616) e HTTP/2 (RFC 7540) e montar um quadro comparativo apresentando as diferenças entre as versões. Entregar em formato de relatório de pesquisa.

ANALISANDO



← Acesso ao domínio livraria.com.br

Página da minha livraria!

Log de mensagens trocadas entre servidor NGINX e cliente HTTP (browser)

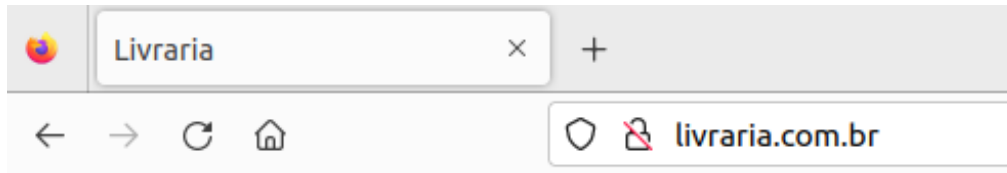
```
127.0.0.1      TCP        74 33678 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3692002432 TSecr=0 WS=128
127.0.0.1      TCP        74 80 → 33678 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3692002432 TSecr=3692002432 WS=128
127.0.0.1      TCP        66 33678 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3692002432 TSecr=3692002432
127.0.0.1      HTTP       515 GET / HTTP/1.1
127.0.0.1      TCP        66 80 → 33678 [ACK] Seq=1 Ack=450 Win=65152 Len=0 TSval=3692002432 TSecr=3692002432
127.0.0.1      HTTP       254 HTTP/1.1 304 Not Modified
127.0.0.1      TCP        66 33678 → 80 [ACK] Seq=450 Ack=189 Win=65408 Len=0 TSval=3692002433 TSecr=3692002433
```

▼ Hypertext Transfer Protocol

▶ GET / HTTP/1.1\r\n

```
Host: cafeteria.com.br\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
If-Modified-Since: Tue, 29 Aug 2023 18:02:48 GMT\r\n
If-None-Match: "64ee32c8-10"\r\n
\r\n
[Full request URI: http://cafeteria.com.br/]
[HTTP request 1/1]
[Response in frame: 144]
```


ANALISANDO



← Acesso ao domínio livraria.com.br

Página da minha livraria!

Log de mensagens trocadas entre servidor NGINX e cliente HTTP (browser)

```
127.0.0.1      TCP        74 33678 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3692002432 TSecr=0 WS=128
127.0.0.1      TCP        74 80 → 33678 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3692002432 TSecr=3692002432 WS=128
127.0.0.1      TCP        66 33678 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3692002432 TSecr=3692002432
127.0.0.1      HTTP       515 GET / HTTP/1.1
127.0.0.1      TCP        66 80 → 33678 [ACK] Seq=1 Ack=450 Win=65152 Len=0 TSval=3692002432 TSecr=3692002432
127.0.0.1      HTTP       254 HTTP/1.1 304 Not Modified
127.0.0.1      TCP        66 33678 → 80 [ACK] Seq=450 Ack=189 Win=65408 Len=0 TSval=3692002433 TSecr=3692002433
```

▼ Hypertext Transfer Protocol

▶ HTTP/1.1 304 Not Modified\r\n

Server: nginx/1.18.0 (Ubuntu)\r\n

Date: Thu, 05 Oct 2023 00:17:32 GMT\r\n

Last-Modified: Tue, 29 Aug 2023 18:02:48 GMT\r\n

Connection: keep-alive\r\n

ETag: "64ee32c8-10"\r\n

\r\n

[HTTP response 1/1]

[Time since request: 0.000155790 seconds]

[\[Request in frame: 142\]](#)

[Request URI: http://cafeteria.com.br/]

MAIS UM EXEMPLO...

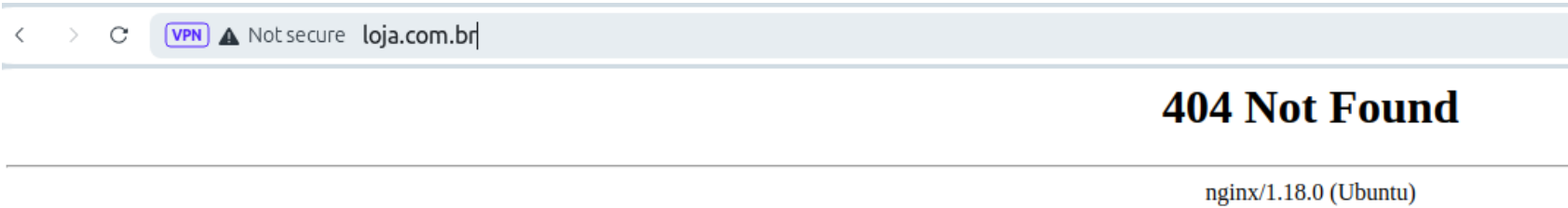


← Acesso ao domínio cafeteria.com.br

Log de mensagens trocadas entre servidor NGINX e cliente HTTP (browser)

404	593.623310011	127.0.0.1	127.0.0.1	TCP	74 80 → 46200 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3692538742 TSecr=3692538742 WS=128
405	593.623331303	127.0.0.1	127.0.0.1	TCP	66 46200 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3692538742 TSecr=3692538742
406	593.623525311	127.0.0.1	127.0.0.1	HTTP	515 GET / HTTP/1.1
407	593.623544676	127.0.0.1	127.0.0.1	TCP	66 80 → 46200 [ACK] Seq=1 Ack=450 Win=65152 Len=0 TSval=3692538742 TSecr=3692538742
408	593.623690131	127.0.0.1	127.0.0.1	HTTP	254 HTTP/1.1 304 Not Modified
409	593.623712014	127.0.0.1	127.0.0.1	TCP	66 46200 → 80 [ACK] Seq=450 Ack=189 Win=65408 Len=0 TSval=3692538742 TSecr=3692538742
410	603.672109238	127.0.0.1	127.0.0.1	TCP	66 [TCP Keep-Alive] 46200 → 80 [ACK] Seq=449 Ack=189 Win=65536 Len=0 TSval=3692548791 TSecr=3692538742
411	603.672134608	127.0.0.1	127.0.0.1	TCP	66 [TCP Keep-Alive ACK] 80 → 46200 [ACK] Seq=189 Ack=450 Win=65536 Len=0 TSval=3692548791 TSecr=3692538742
412	613.912256999	127.0.0.1	127.0.0.1	TCP	66 [TCP Keep-Alive] 46200 → 80 [ACK] Seq=449 Ack=189 Win=65536 Len=0 TSval=3692559031 TSecr=3692548791
413	613.912281307	127.0.0.1	127.0.0.1	TCP	66 [TCP Keep-Alive ACK] 80 → 46200 [ACK] Seq=189 Ack=450 Win=65536 Len=0 TSval=3692559031 TSecr=3692538742

ANALISANDO — PAGE NOT FOUND



```
127.0.0.1      TCP      74 40576 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3692875467 TSecr=0 WS=128
127.0.0.1      TCP      74 80 → 40576 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3692875467 TSecr=3692875467 WS=128
127.0.0.1      TCP      66 40576 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3692875467 TSecr=3692875467
127.0.0.1      HTTP     578 GET / HTTP/1.1
127.0.0.1      TCP      66 80 → 40564 [ACK] Seq=1 Ack=513 Win=65024 Len=0 TSval=3692875469 TSecr=3692875469
127.0.0.1      HTTP     458 HTTP/1.1 404 Not Found (text/html)
```

▼ Hypertext Transfer Protocol

▶ GET / HTTP/1.1\r\n

Host: loja.com.br\r\n

Connection: keep-alive\r\n

Upgrade-Insecure-Requests: 1\r\n

User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 OPR/102.0.0.0\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n

Accept-Encoding: gzip, deflate\r\n

Accept-Language: en-US,en;q=0.9\r\n

If-None-Match: W/"6509a715-eb"\r\n

If-Modified-Since: Tue, 19 Sep 2023 13:50:13 GMT\r\n

\r\n

[Full request URI: <http://loja.com.br/>]

[HTTP request 1/1]

[Response in frame: 595]

ANALISANDO...

< > ↻ VPN ▲ Not secure loja.com.br

404 Not Found

nginx/1.18.0 (Ubuntu)

127.0.0.1	TCP	74 40576 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=3692875467 TSecr=0 WS=128
127.0.0.1	TCP	74 80 → 40576 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=3692875467 TSecr=3692875467 WS=128
127.0.0.1	TCP	66 40576 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3692875467 TSecr=3692875467
127.0.0.1	HTTP	578 GET / HTTP/1.1
127.0.0.1	TCP	66 80 → 40564 [ACK] Seq=1 Ack=513 Win=65024 Len=0 TSval=3692875469 TSecr=3692875469
127.0.0.1	HTTP	458 HTTP/1.1 404 Not Found (text/html)

- ▼ Hypertext Transfer Protocol
 - ▶ HTTP/1.1 404 Not Found\r\nServer: nginx/1.18.0 (Ubuntu)\r\nDate: Thu, 05 Oct 2023 00:32:05 GMT\r\nContent-Type: text/html\r\nTransfer-Encoding: chunked\r\nConnection: keep-alive\r\nContent-Encoding: gzip\r\n\r\n[HTTP response 1/1]\n[Time since request: 0.000185758 seconds]\n[Request in frame: 593]\n[Request URI: http://loja.com.br/]
 - ▶ HTTP chunked responseContent-encoded entity body (gzip): 186 bytes -> 564 bytesFile Data: 564 bytes
 - ▶ Line-based text data: text/html (13 lines)

REFERÊNCIAS

- FOROUZAN, Behrouz A.; MOSHARRAF, Firouz. **Redes de computadores**. Porto Alegre: Grupo A, 2013. E-book. ISBN 9788580551693. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788580551693/>. Acesso em: 24 jul. 2023.
- FOROUZAN, Behrouz A. **Protocolo TCP/IP**. Porto Alegre: Grupo A, 2010. E-book. ISBN 9788563308689. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788563308689/>. Acesso em: 05 out. 2023.
- **HTTP/2 guide**. Acesso em 25 de julho de 2023. Disponível por www em <https://httpd.apache.org/docs/2.4/howto/http2.html>.
- RFC 7540. **Hypertext Transfer Protocol Version 2 (HTTP/2)**. Acesso em 25 de julho de 2023. Disponível por www em <https://datatracker.ietf.org/doc/html/rfc7540>.
- <https://http2-explained.haxx.se/en/part5>