

# Protocolo HTTP

Alisson Chiquitto<sup>1</sup>

<sup>1</sup>Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Cianorte, 2014

- 1 Introdução
- 2 Requisições e Respostas
- 3 Cabeçalhos
  - Requisição (request)
  - Resposta (response)
- 4 Códigos de resposta
- 5 Usando o protocolo HTTP via Telnet
- 6 Glossário

# O que é o protocolo HTTP?

- *Hypertext Transfer Protocol (HTTP)* é um protocolo utilizado para enviar e receber informações na web.
- A versão mais utilizada atualmente é a 1.1, definida pela especificação **RFC 2616**;

# Requisições e Respostas

- O *HTTP* é baseado em requisições (*request*) e respostas (*response*) entre clientes e servidores;
- O **cliente** muitas vezes é o navegador (do usuário) ou outro dispositivo/programa que fará a requisição. Ele solicita um determinado **recurso** (*resource*), enviando um pacote de informações contendo alguns **cabeçalhos** (*headers*) a um **URI**;
- O **servidor** recebe a requisição e envia uma resposta, que pode ser um recurso ou simplesmente outro cabeçalho (*header*)

# Requisições e Respostas

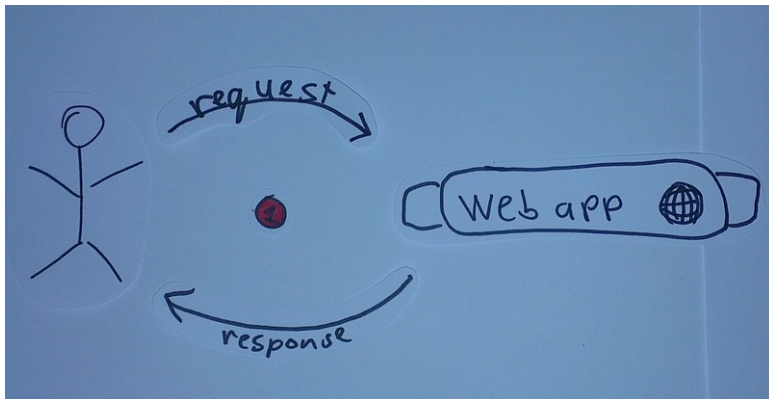


Figura: Protocolo HTTP simplificado

# Requisições e Respostas/Cabeçalhos

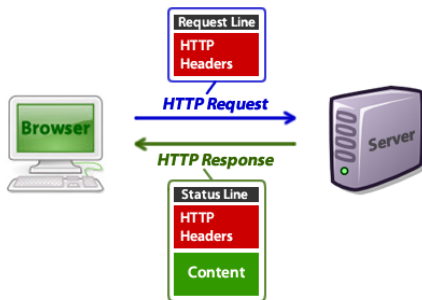


Figura: Protocolo HTTP e cabeçalhos

# Outline

- 1 Introdução
- 2 Requisições e Respostas
- 3 Cabeçalhos**
  - Requisição (request)
  - Resposta (response)
- 4 Códigos de resposta
- 5 Usando o protocolo HTTP via Telnet
- 6 Glossário

# Elementos do cabeçalho de requisição

Uma requisição *HTTP* é um conjunto de linhas enviado ao servidor pelo navegador. Nesta requisição existe 3 elementos:

- **Linha inicial** (*Request-Line*);
- **Linhas de cabeçalhos** (*Request-Header*);
- **Corpo da requisição** (*Request-Body*);



# Cabeçalho de requisição: Linha inicial

A linha inicial de uma requisição é composta por três partes separadas por espaços: o **método** (*Method*), a **identificação do URI** (*Request-URI*) e a **versão do HTTP** (*HTTP-Version*) utilizado.

```
GET /diretorio/arquivo.html HTTP/1.0
```

Figura: Linha inicial de uma requisição HTTP

# Cabeçalho de requisição: Linhas de cabeçalho

O cabeçalho da mensagem (*header*) é utilizado para transmitir informações adicionais entre o cliente e o servidor. Ele é especificado imediatamente após a linha inicial, seguido de dois pontos (:) e um valor.

```
Host: www.exemplo.com
User-Agent: Mozilla/5.0 (Windows
Accept: text/html,application/xhtml
Accept-Language: pt-br,pt;q=0.8,
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8
Keep-alive: 115
Cookie: nome=valor; nome2=valor2
Connection: keep-alive
Content-Length: 28
```

Figura: Cabeçalhos de uma requisição HTTP

# Cabeçalho de requisição: Corpo

Em uma mensagem de requisição, o corpo pode conter dados que serão enviados diretamente pelo usuário ou um arquivo que será enviado para o servidor. Este item é opcional.

```
usuario=exemplo&senha=123456
```

Figura: Corpo de uma requisição HTTP

# Cabeçalho de requisição HTTP

```
1 GET /diretorio/arquivo.html HTTP/1.0
2 Host: www.exemplo.com
3 User-Agent: Mozilla/5.0 (Windows; U;
4 Accept: text/html,application/xhtml+xml
5 Accept-Language: pt-br,pt;q=0.8,en-us
6 Accept-Encoding: gzip,deflate
7 Accept-Charset: ISO-8859-1,utf-8;q=0.
8 Keep-alive: 115
9 Cookie: nome=valor; nome2=valor2
10 Connection: keep-alive
11 Content-Length: 28
12
13 usuario=exemplo&senha=123456
```

Figura: Corpo de uma requisição HTTP

- 1 Introdução
- 2 Requisições e Respostas
- 3 Cabeçalhos**
  - Requisição (request)
  - Resposta (response)
- 4 Códigos de resposta
- 5 Usando o protocolo HTTP via Telnet
- 6 Glossário

# Elementos do cabeçalho de resposta

Uma mensagem de resposta do servidor é composta pelos seguintes campos:

- **Linha inicial** (*Status-Line*);
- **Linhas de cabeçalhos** (*Response-Header*);
- **Corpo da requisição** (*Response-Body*);

# Cabeçalho de resposta: Linha inicial

A linha inicial de uma resposta HTTP contém 3 partes separadas por espaços: a versão do protocolo *HTTP* (*HTTP-Version*), um código de status (*Status-Code*) da resposta, que fornece o resultado da requisição, e uma frase de justificativa (*Reason-Phrase*) que descreve o código do status.



```
HTTP/1.1 200 OK
```

Figura: Linha inicial de uma resposta HTTP

# Cabeçalho de resposta: Linhas de cabeçalho

São utilizados pelo servidor ao retornar o recurso no qual foi requisitado pelo cliente, para transmitir informações que descrevem as configurações do servidor e do recurso identificado pelo URI de requisição, e que não pertence à linha de status (*responseheader*).

```
Server: nginx/1.4.4  
Date: Thu, 06 Feb 2014 17:25:04 GMT  
Content-Type: text/html  
Connection: close
```

Figura: Cabeçalhos de uma resposta HTTP



# Cabeçalho de resposta: Corpo

Em uma mensagem de resposta, o corpo da mensagem é o recurso que foi requisitado pelo cliente, ou ainda uma mensagem de erro, caso este recurso não seja possível.

```
<!DOCTYPE html>  
<html>  
<head></head>  
<body></body>  
</html>
```

Figura: Corpo de uma requisição HTTP

# Cabeçalho de resposta HTTP

```
35 HTTP/1.1 200 OK
36 Server: nginx/1.4.4
37 Date: Thu, 06 Feb 2014 17:25:04 GMT
38 Content-Type: text/html
39 Connection: close
40
41 <!DOCTYPE html>
42 <html>
43 <head></head>
44 <body></body>
45 </html>
```

Figura: Corpo de uma resposta HTTP

# Códigos de resposta: Classe 2xx e 3xx

## 200 OK

A requisição foi bem sucedida.

## 301 Moved Permanently

O recurso foi movido permanentemente para outra URI.

## 302 Found

O recurso foi movido temporariamente para outra URI.

## 304 Not Modified

O recurso não foi alterado.

# Códigos de resposta: Classe 4xx e 5xx

## 401 Unauthorized

A URI especificada exige autenticação do cliente. O cliente pode tentar fazer novas requisições.

## 403 Forbidden

O servidor entende a requisição, mas se recusa em atendê-la.  
O cliente não deve tentar fazer uma nova requisição.

## 404 Not Found

O servidor não encontrou nenhuma URI correspondente.

## 405 Method Not Allowed

O método especificado na requisição não é válido na URI. A resposta deve incluir um cabeçalho Allow com uma lista dos métodos aceitos.

## 410 Gone

O recurso solicitado está indisponível mas seu endereço atual não é conhecido.

# Códigos de resposta: Classe 5xx

## 500 Internal Server Error

O servidor não foi capaz de concluir a requisição devido a um erro inesperado.

## 503 Service Unavailable

O servidor não é capaz de processar a requisição pois está temporariamente indisponível.

# Usando o protocolo HTTP via Telnet

- 1 Em um editor de texto (*notepad*, *sublime text*) digite todo o cabeçalho de requisição;
- 2 No terminal (ou *prompt comando* no *windows*), digite: **telnet IP\_SERVIDOR PORTA**. O *IP\_SERVIDOR* é o IP do servidor do site que você quer conectar. A porta padrão do protocolo *HTTP* é 80;
- 3 Com a conexão aberta via *telnet*, digite todo o cabeçalho de requisição (ou copie o cabeçalho digitado no editor de texto e cole no terminal. Use o botão direito do *mouse* para facilitar a tarefa);
- 4 Para informar ao telnet que o cabeçalho de requisição está pronto, insira duas novas linhas em branco no terminal (aperte *Enter* duas vezes);
- 5 Aguarde a resposta *HTTP*;

## Protocolo

Na ciência da computação, um protocolo é uma convenção que controla e possibilita uma conexão, comunicação, transferência de dados entre dois sistemas computacionais. De maneira simples, um protocolo pode ser definido como "as regras que governam" a sintaxe, semântica e sincronização da comunicação. Os protocolos podem ser implementados pelo hardware, software ou por uma combinação dos dois.

 RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1

<http://tools.ietf.org/html/rfc2616>

 Uniform resource identifier

*Wikipedia.*

[http://en.wikipedia.org/wiki/Uniform\\_Resource\\_Identifier](http://en.wikipedia.org/wiki/Uniform_Resource_Identifier)

 Protocolo

*Protocolo.*

[http://pt.wikipedia.org/wiki/Protocolo\\_\(ci%C3%Aancia\\_da\\_computa%C3%A7%C3%A3o\)](http://pt.wikipedia.org/wiki/Protocolo_(ci%C3%Aancia_da_computa%C3%A7%C3%A3o))

 Make HTTP requests

*Hurl.it.*

<http://www.hurl.it/>