

Laboratório de Programação de Web Sites Dinâmicos

HTTP

Tassio Sirqueira – 2019/02

| HTTP | Principais Conceitos

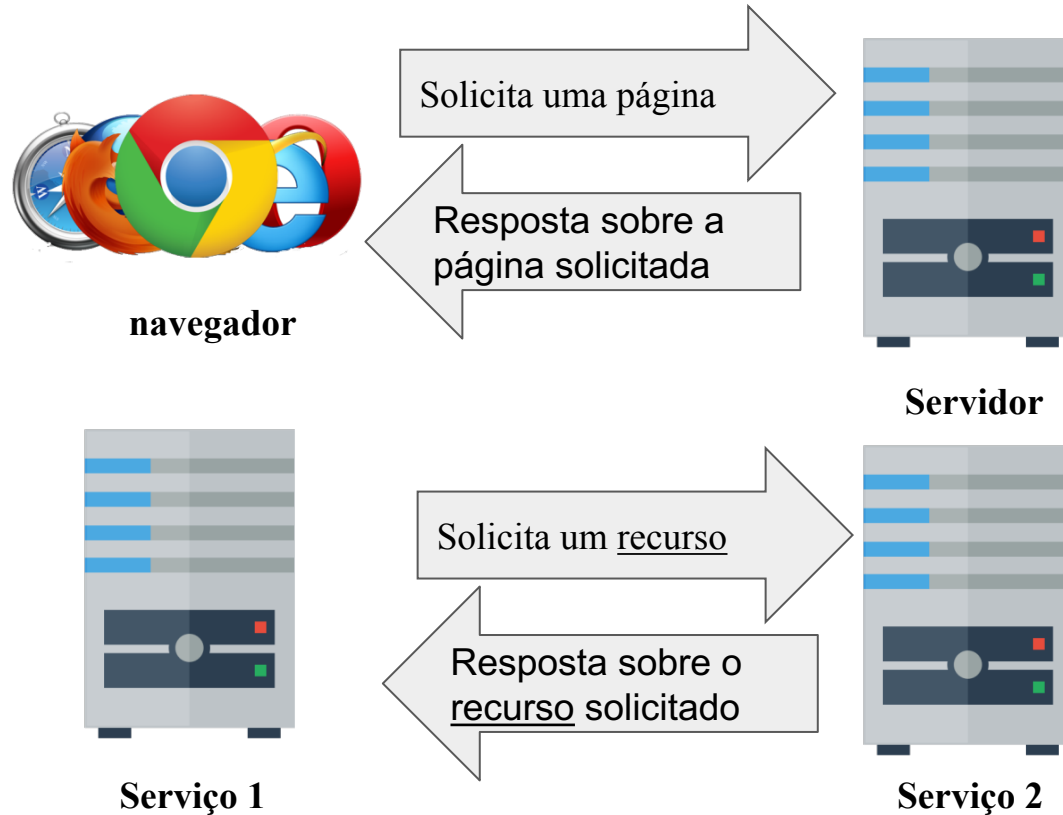
HTTP



A Internet moderna é inteiramente baseada no protocolo Hypertext Transfer Protocol (**HTTP**).

- ❑ Protocolo de comunicação da camada de aplicação.
- ❑ Não possui estado, portanto não tem ‘memória’ das requisições anteriores.
- ❑ Especificado nas RFCs (7230-7237).

| HTTP | Exemplo de comunicação HTTP



| HTTP | HTTP

- ❑ O **HTTP** se resume a requisições e respostas.
 - ❑ Assume muito pouco a respeito dos detalhes de implementação.
 - ❑ A versão mais utilizada é a HTTP 1.1, embora já exista a versão HTTP 2.0.



| HTTP | Recursos

- ❑ Na *web* qualquer endereço que pode ser acessado é um **recurso**.
- ❑ Representamos esses endereços por meio das **URLs** (Uniform Resource Locators)
 - ❑ `http://www.google.com`
 - ❑ `http://globoesporte.globo.com/`
- ❑ Uma URL pode possuir várias partes.

http://www.domain.com:1234/path/to/resource?a=b&x=y				
Protocolo	Servidor (IP, DNS..)	Porta	Caminho para o recurso	Parâmetros de Busca

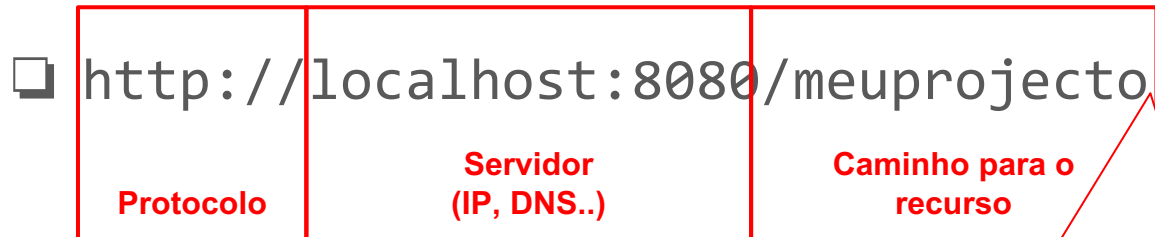
| HTTP | Recursos



The diagram shows a URL `https://www.google.com.br/search?q=o+que+é+http` enclosed in a red border. The URL is divided into four segments by red lines. Below each segment is a red label. A red line connects the 'Servidor' label to the 'Caminho para o recurso' label, and another red line connects the 'Caminho para o recurso' label to the 'Parâmetros de Busca' label.

<code>https://</code>	<code>www.google.com.br</code>	<code>/search</code>	<code>?q=o+que+é+http</code>
Protocolo	Servidor (IP, DNS..)	Caminho para o recurso	Parâmetros de Busca

A porta é opcional, se omitida é utilizada a porta padrão do TCP 80.



The diagram shows a URL `http://localhost:8080/meuprojecto` enclosed in a red border. The URL is divided into three segments by red lines. Below each segment is a red label. A red line connects the 'Servidor' label to the 'Caminho para o recurso' label.

<code>http://</code>	<code>localhost:8080</code>	<code>/meuprojecto</code>
Protocolo	Servidor (IP, DNS..)	Caminho para o recurso

O uso de parâmetros de busca é opcional.

| HTTP | Verbos

- ❑ Uma URL indica um recurso disponível no Servidor com o qual queremos nos comunicar.
- ❑ No entanto, a ação que deve ser realizada é definida pelo **Verbo**:
 - ❑ **GET** - Obtém um recurso existente de acordo com os dados da URL.
 - ❑ **POST** - Cria um novo recurso com os dados enviados na requisição.
 - ❑ **PUT** - Atualiza um recurso com base nos dados enviados na requisição.
 - ❑ **DELETE** - Remove um recurso existente.
- ❑ Observe que esses recursos “equivalem” às operações de cadastro, obtenção, atualização e remoção (CRUD) que as aplicações comumente oferecem.
- ❑ Outros verbos menos utilizados **HEAD, TRACE, OPTIONS...**

| HTTP | Status Code

- ❑ Com uma **URL** e um **Verbo**, um cliente pode iniciar uma comunicação com o servidor.
- ❑ Que vai retornar com um **Status Code** e uma resposta.
- ❑ Esse código de status é importante para interpretar a resposta obtida.
 - ❑ Esse status é um código numérico de 3 dígitos.
 - ❑ Entre 200 e 299 é um status de sucesso.
 - ❑ Entre 300 e 399 é um redirecionamento.
 - ❑ Entre 400 e 499 é um erro na mensagem enviada pelo cliente.
 - ❑ Entre 500 e 599 é um erro no servidor.



| HTTP | Status Code | 2xx: Successful

- ❑ Um retorno 2XX indica que a requisição foi bem sucedida.
- ❑ O retorno mais comum é o **200 OK**.
 - ❑ Por exemplo, ao realizar um GET para um arquivo no servidor o retorno é um 200 OK com o conteúdo do arquivo no corpo da resposta.
- ❑ Outros códigos de status comuns:
 - ❑ **201 Created** - O recurso enviado foi criado com sucesso no servidor.
 - ❑ **202 Accepted** - A requisição foi aceita mas o servidor ainda não tem uma resposta. Utilizada em serviços assíncronos.
 - ❑ **204 No Content** - A requisição foi bem sucedida mas não há resposta.



everything is going to be
● **200 OK**

| HTTP | Status Code | 3xx: Redirection

- ❑ Um retorno 3XX indica que uma ação adicional do cliente, para obter o recurso desejado em outro lugar.
- ❑ O retorno mais comum é o **301 Moved Permanently**.
 - ❑ Por exemplo, ao realizar um GET para um arquivo no servidor o retorno é um 300 Moved Permanently e é informada a nova localização do mesmo.
- ❑ Outros códigos de status comuns:
 - ❑ **303 See Other** - O recurso foi temporariamente movido para outra URL.
 - ❑ **304 Not Modified** - O cliente pode enviar dados da versão que possui do recurso desejado, caso ainda seja o mesmo no servidor o retorno é 304.

| HTTP | Status Code | 4xx: Client Error

- ❑ Um retorno 4XX indica que há um problema na requisição que impede que o servidor a processe adequadamente.

- ❑ O retorno mais comum é o **404 Not Found**.

- ❑ Por exemplo, ao realizar um GET para um recurso não existente no servidor, o retorno é um 404 Not Found.

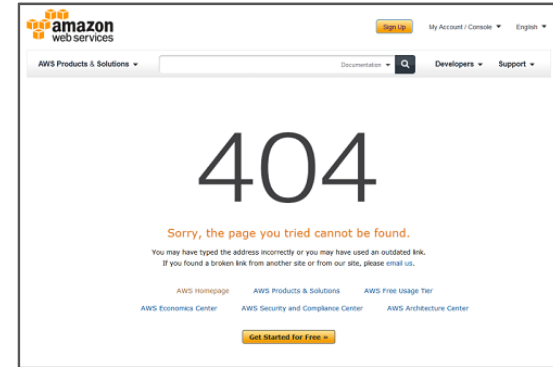
- ❑ Outros códigos de status comuns:

- ❑ **400 Bad Request** - A requisição está incompleta ou mal formatada.

- ❑ **401 Unauthorized** - O cliente não possui credenciais válidas.

- ❑ **403 Forbidden** - Acesso negado.

- ❑ **405 Method Not Allowed** - Aquele recurso não suporta o verbo HTTP enviado pelo cliente.



| HTTP | Status Code | 5xx: Server Error

- ❑ Um retorno 5XX indica que o servidor falhou enquanto processava a requisição.
- ❑ O retorno mais comum é o **500 Internal Server Error**.
 - ❑ Por exemplo, ao realizar um GET para um recurso e houve um erro ao acessar o banco de dados, o retorno é um 500 Internal Server Error.
- ❑ Outros códigos de status comuns:
 - ❑ **501 Not Implemented** - A requisição está sendo feita para um recurso que ainda não está pronto para receber requisições.
 - ❑ **503 Service Unavailable** - O servidor está temporariamente fora do ar.



| HTTP | Comunicação



Cliente



Servidor

| HTTP | Cabeçalhos

- ❑ Além das URLs, Verbos e Status Code, um importante componente da comunicação HTTP são os cabeçalhos.
- ❑ Eles são enviados junto com a requisição e oferecem informações adicionais:
 - ❑ Sobre o conteúdo da requisição/resposta como tipo de arquivo, tamanho, encoding...
 - ❑ Formatos de arquivo e idioma suportados na requisição/resposta.
 - ❑ Informações de autorização.
 - ❑ Dados do browser e do servidor utilizado...

| HTTP | Exemplo - Requisição

```
GET /doc/test.html HTTP/1.1
```

```
Host: www.test101.com
```

```
Accept: image/gif, image/jpeg, */*
```

```
Accept-Language: en-us
```

```
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0
```

```
Content-Length: 35
```

```
bookId=12345&author=Tan+Ah+Teck
```

Request Line

Request Headers

Request
Message
Header

A blank line separates header & body

Request Message Body

| HTTP | Exemplo - Resposta

HTTP/1.1 200 OK

Date: Sun, 08 Feb xxxx 01:11:12 GMT

Server: Apache/1.3.29 (Win32)

Last-Modified: Sat, 07 Feb xxxx

ETag: "0-23-4024c3a5"

Accept-Ranges: bytes

Content-Length: 35

Connection: close

Content-Type: text/html

<h1>My Home page</h1>

→ Status Line

} Response Headers

} Response
Message
Header

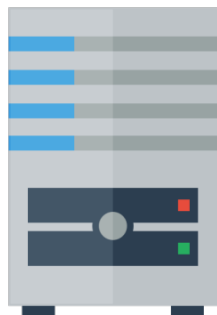
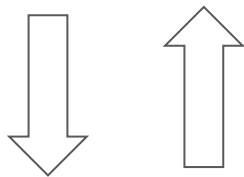
→ A blank line separates header & body

} Response Message Body

| HTTP | Exemplo - Resposta



navegador



Serviço 1

GET /doc/test.html HTTP/1.1

Host: www.test101.com

Accept: image/gif, image/jpeg, */*

Accept-Language: en-us

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0

Content-Length: 35

bookId=12345&author=Tan+Ah+Teck

Request Line

Request Headers

Request
Message
Header

A blank line separates header & body

Request Message Body

HTTP/1.1 200 OK

Date: Sun, 08 Feb xxxx 01:11:12 GMT

Server: Apache/1.3.29 (Win32)

Last-Modified: Sat, 07 Feb xxxx

ETag: "0-23-4024c3a5"

Accept-Ranges: bytes

Content-Length: 35

Connection: close

Content-Type: text/html

<h1>My Home page</h1>

Status Line

Response Headers

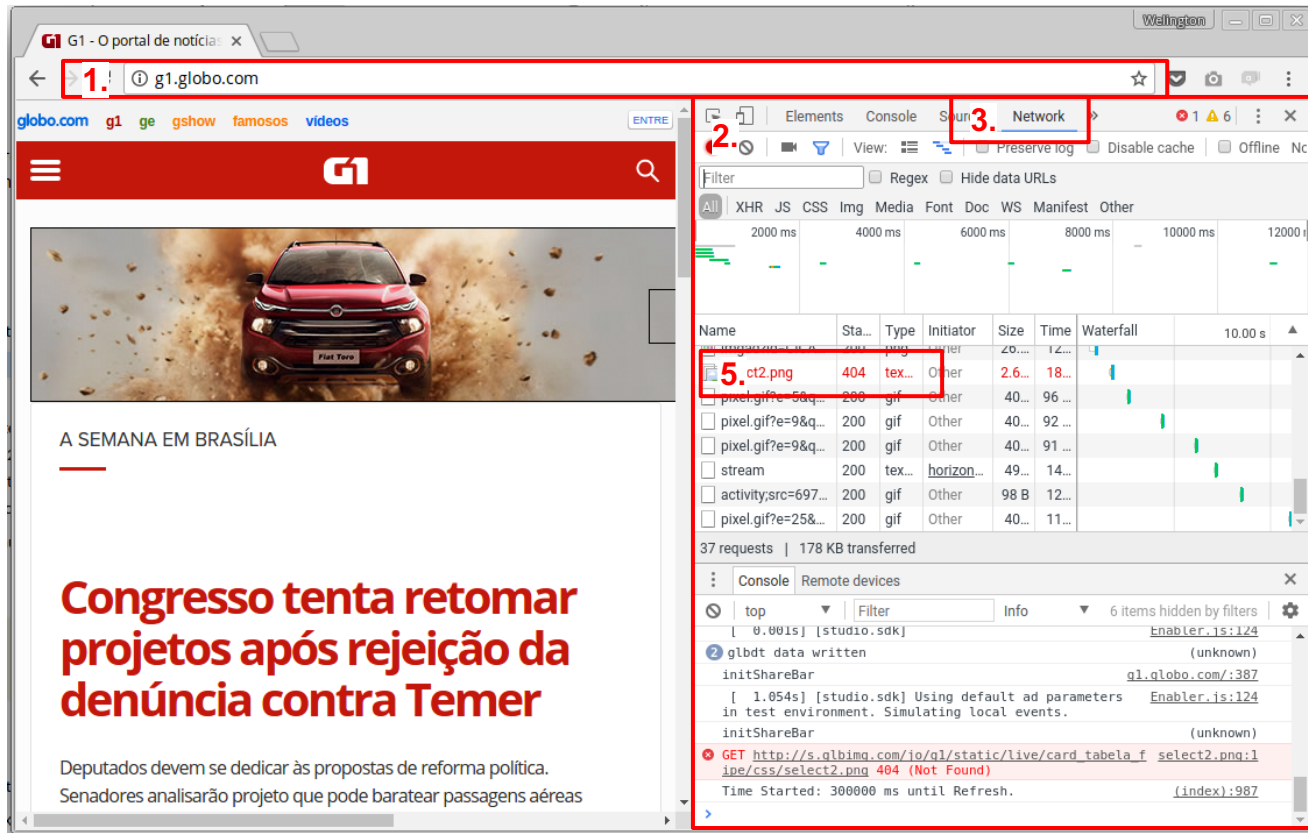
Response
Message
Header

A blank line separates header & body

Response Message Body

| HTTP | Verifique Você Mesmo.

1. Abra o navegador Chrome e navegue até uma página Web.
2. Para abrir as ferramentas de desenvolvedor aperte F12.
3. Selecione a aba **Rede** (Ou Network).
4. Pressione F5 e veja as requisições HTTP.
5. Selecione uma requisição e observe seus dados e da resposta obtida.



| HTTP | Referências adicionais

1. https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html
2. <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Headers>
3. <https://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177>
4. <https://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-2--net-31155>