





HTTP Back-end

<input checked="" type="checkbox"/> Completo	<input type="checkbox"/>
 Criador	 Kauê Matos

O que é Http?

O Hypertext Transfer Protocol, sigla HTTP é um protocolo de comunicação utilizado para sistemas de informação de **hipermídia**, distribuídos e colaborativos. Ele é a base para a comunicação de dados da World Wide Web. Hipertexto é o texto estruturado que utiliza ligações lógicas entre nós contendo texto.

- **HTTP - foi feito para estabelecer regras de comunicação em URLs disponíveis na internet.**

Dados preenchidos em um form (Formulário)

Todos os dados preenchidos em um formulário e enviado para banco de dados via HTTP, chegando no banco de dados ele irá verificar se aquele dado é verdadeiro ou não e após isso ele irá exibir na sua página dependendo do tipo de formulário que você preencheu.

HTTP envia dados em texto puro?

Sim o Http trafega texto puro...como assim? basicamente quando preenchemos um formulário de login para acessar um aplicativo ou site ou algo parecido o http leva essa informações para o banco de dados para verificar e nesse caminho esse texto que nós escrevemos que no caso e nossa senha e nosso nome de usuário vai sem criptografia para ser buscado no banco de dados...isso pode ocasionar em um grande problema no Protocolo HTTP já que ele pode trazer grandes problemas caso alguém consiga obter acesso a esses dados pessoais que foram enviados por um usuário.

HTTPS ou HTTP SSL

- **O que é SSL ?**

Esse “s” representa a tecnologia SSL (em inglês, Secure Sockets Layer), criada em 1994 pela Netscap. Esse tipo de certificado digital funciona como uma “camada” de segurança em sites. Com ele, o site torna-se um ambiente seguro, muito mais atrativo para o mercado.

- **HTTPS**

HTTPS é uma implementação do protocolo HTTP sobre uma camada adicional de segurança que utiliza o protocolo SSL/TLS. Essa camada adicional permite que os dados sejam transmitidos por meio de uma conexão criptografada e que se verifique a autenticidade do servidor e do cliente por meio de certificados digitais.



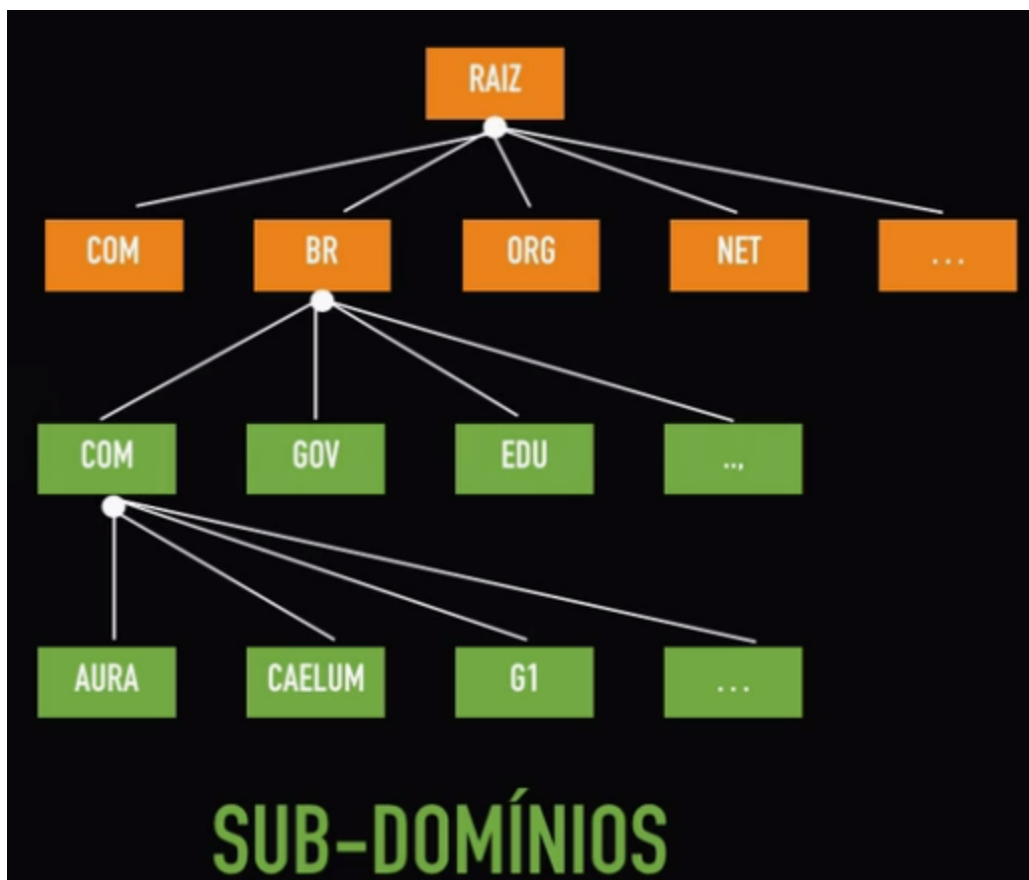
- **Importante* / Requisitos para o HTTPS**

É muito importante que tenhamos em mãos um **identidade confirmada** da sua empresa que está hospedada na internet ou seja um **certificado digital** que irá provar que a sua empresa é realmente uma empresa de vendas ou algo parecido...

Com isso você terá acesso ao HTTPS para ter mais segurança na sua página ao enviar dados de um formulário, por exemplo.

- **Acesso a dados criptografados com Certificado digital**

Sub - Domínios



- **DNS:** O DNS (Domain Name System – Sistema de nome de domínio) converte nomes de domínio legíveis por humanos (por exemplo, www.amazon.com) em endereços IP legíveis por máquina (por exemplo, 192.0.2.44).
- **Endereços:** uma Url começa com um protocolo por exemplo: (<http://>) seguido pelo domínio (www.alura.com.br).

- Após o isso também conseguimos especificar o caminho desse domínio no caso em que página você está por exemplo:(<http://www.alura/curso/front-end.com.br>)

O que são os cookies = Sessões ?

É uma chave de permissão que é gerada quando entramos em um site basicamente um **(Requerimento)** dentro dos **cookies** fica armazenado todas as informações de acesso de uma página que o usuário entrou, assim permitindo que ele tenha acesso a sua página normalmente sem precisar que fique colocando suas informações toda hora quando for acessar.

Depuração com HTTP

O que é Depuração? A depuração é feita quando precisamos acessar um site e estamos em busca de informações o site ele precisa ser carregado para que todas as coisas disponíveis no site apareçam então o **CSS(Estilo da página)** e carregado as imagens entre outras coisas são carregadas tudo por meio da depuração e pelo método **GET**(Que é uma ação de buscar o conteúdo),para a página hospedada na web.

- **Requisição:** cada coisa que vemos imagens vídeos e etc é feita uma requisição para que conseguimos ver essa imagem ou esse vídeo.

Classes de protocolo HTTP

2xx - Sucesso

3xx - ??

4xx - Erro do cliente

5xx - Erro do servidor

- 3XX:Redirecionamento

Status code do HTTP

STATUS CODE

200 - OK

301 - Moved Permanently

404 - Not Found

500 - Internal Server Error

Parâmetros | Métodos
GET,POST,DELETE,PUT

METHODS

GET - Receber dados (params na URL)

POST - Submeter dados(params no corpo da requisição)

DELETE - Remover um recurso

PUT - Atualizar um recurso

- **Método GET**

O **método GET** utiliza a própria URI (normalmente chamada de URL) para enviar dados ao servidor, quando enviamos um formulário pelo **método GET**, o navegador pega as informações do formulário e coloca junto com a URI de onde o formulário vai ser enviado e envia, separando o endereço da URI dos dados do formulário por um “?”

...

- **Método POST**

O método de requisição **POST** foi projetado para solicitar que o servidor web aceite os dados anexados no corpo da mensagem de requisição para

armazenamento. Ele é normalmente usado quando se faz o upload de um arquivo ou envia-se um formulário web completo.

- **Delete:Remove um Recurso**

- **PUT:Atualiza um recurso**

Resumo: Existem outros métodos de requisição para o HTTP | URL

O que é REST | Representational State Transfer ?

Os princípios REST são definidos por quatro controles de interface, incluindo identificação de recursos, gerenciamento de recursos por meio de representações, comunicações auto descritivas e hipermídia como mecanismo do estado do aplicativo.

HTTP2 - Dados binários,GZIP ativo e TLS

```
BP-de-Fabio:~ fabiopimentel$ curl -v www.caelum.com.br
Rebuilt URL to: www.caelum.com.br/
Trying 216.58.222.115...
TCP_NODELAY set
Connected to www.caelum.com.br (216.58.222.115) port 80 (#0)

GET / HTTP/1.1
Host: www.caelum.com.br
User-Agent: curl/7.51.0
Accept: */*

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding,User-Agent
Content-Language: pt-br
Content-Type: text/html; charset=UTF-8
X-DNS-Prefetch-Control: on
X-Cloud-Trace-Context: d58c6460837073c01e35848bac3d3848
Date: Mon, 20 Feb 2017 18:32:33 GMT
Server: Google Frontend
Content-Length: 94230

<!DOCTYPE html> <html class="no-js"lang="pt-br"> <head> <title>Caelum </title>
```

MÉTODO E URL

STATUS CODE

RESPOSTA

HTTP2 comprimido em Gzip

```
GET / HTTP/1.1
Host: www.caelum.com.br
User-Agent: curl/7.51.0
Accept-Encoding: gzip

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Vary: Accept-Encoding,User-Agent
Content-Language: pt-br
Content-Type: text/html; charset=UTF-8
Date: Mon, 20 Feb 2017 18:32:33 GMT
Server: Google Frontend
Content-Encoding: gzip

H4sIAAAAAA/12Q3U4CMRBGX2XsdRfj/S4JQ0wXo15QGC+HdnDH9GdtZ+HG1/FZfDG7gEG8mnbyzcmXA/XN/ct0/bac
h16h0qk15Q4DQ11EsjsZTJNd7+IRpn3LMYAkecy8aRs+ztYZJ3sCmy1bCcLzW5T0q9hv168aRhu1rBKVzw3AnUtyckQ01
e20jw9myo0n408xZGV2WdJpq7oeUfxDYmj1JZEjLCMVx0Qo66NqZqQvx3lGhLKV66hGPiHYcSuy5oKR/7X4EncBYj5K
405YmYRN+wIPHKGRIXckmFbICGCP2TQRg0merBsv7+E0ZZX6UzBMFosBA5ZUj8xUYaPHh1vebJKIzX+AcdfBv03AQAA
```

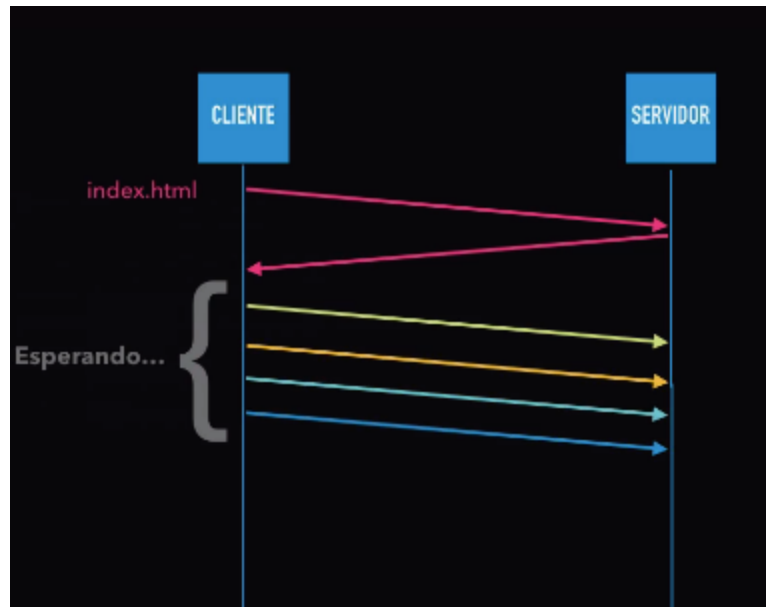
HTTP/2

GZIP

- Qual será o sentido de ter uma HTTP compilado

Basicamente quando navegamos pelos **smartphones** temos uma banda de largura um pouco maior, Envio de dados um mais elevado do que o normal esperado pelos servidores e também o consumo de conteúdo disponibilizado na tela do dispositivo, Por conta disso existe o **HTTP 2** onde ele possibilita que a alguns conteúdos **compilados** como dados pessoais ou dados da página que está sendo utilizado.

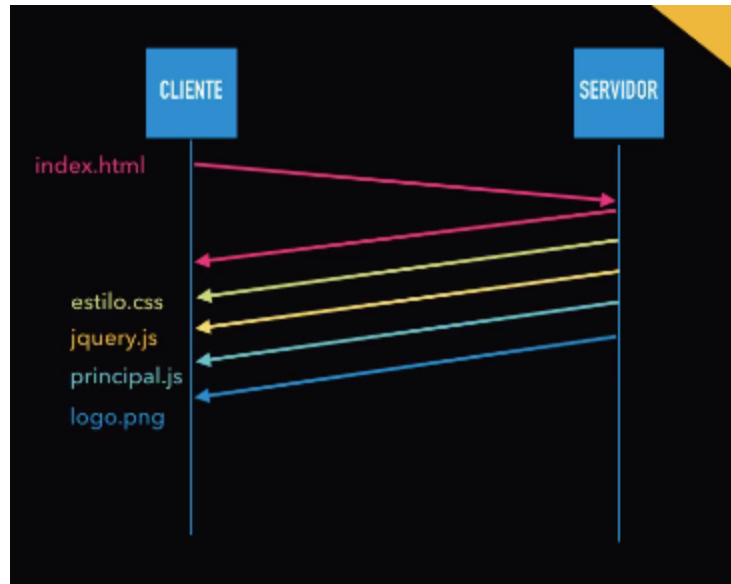
HTTP2 - SERVER PUSH....Requisições



HTTP 1

- A cada arquivo que é enviado para servidor é feita uma requisição para buscar os arquivos que no caso seria com método **GET**, após isso esse dados buscados são devolvidos para nossa página ou programa.

HTTP2

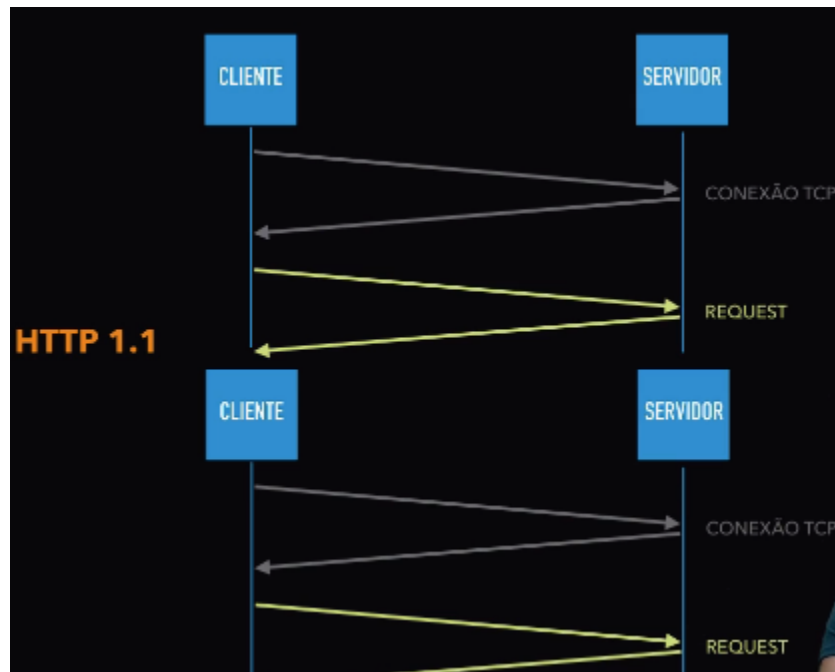


- No **HTTP2** as coisas são um pouco diferente já que pode ser retornado todo o conteúdo da sua página somente após ter feito uma única requisição na sua página ou APP.

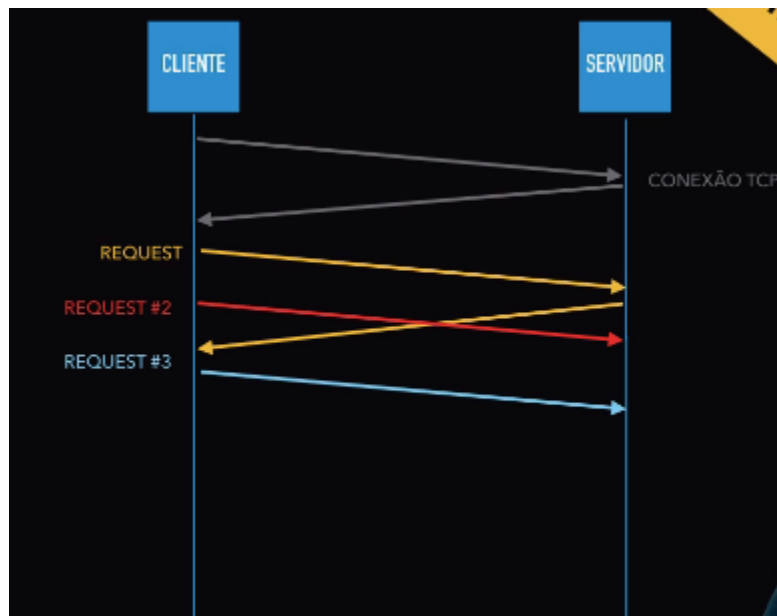
HTTP2 - Multiplexação

- **HTTP1**

Antes de conseguirmos fazer uma requisição no **HTTP1** existe uma conexão **TCP**(Protocolo de transporte) onde é feita antes da requisição.



- HTTP 2



- No HTTP2 não esperamos uma requisição terminar para executar outra e totalmente dinâmico após ter feito uma as outras começam a serem executadas logo em seguida.
- ESSE CONCEITO PODE SER CHAMADO DE MULTIPLEX

