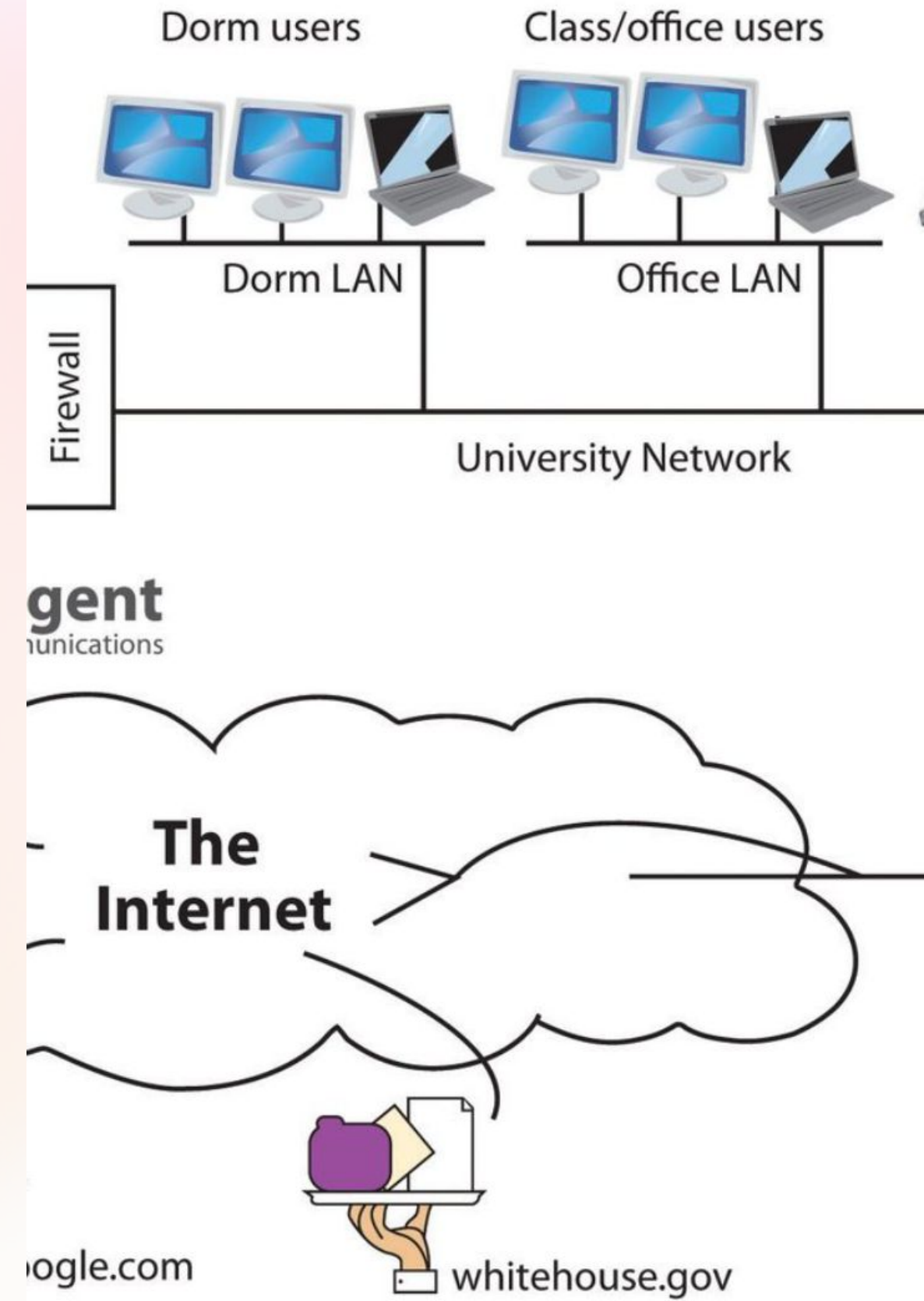
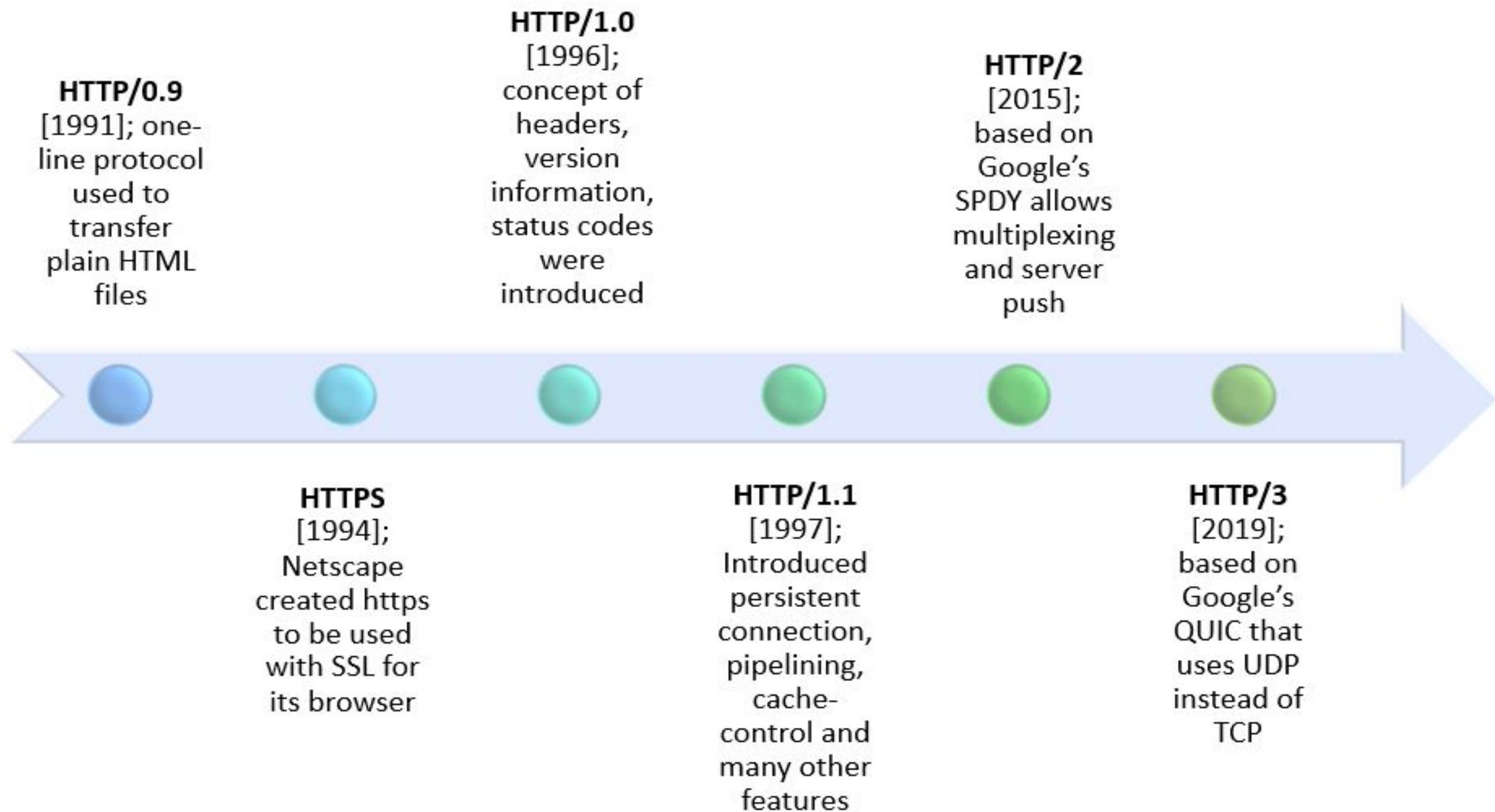


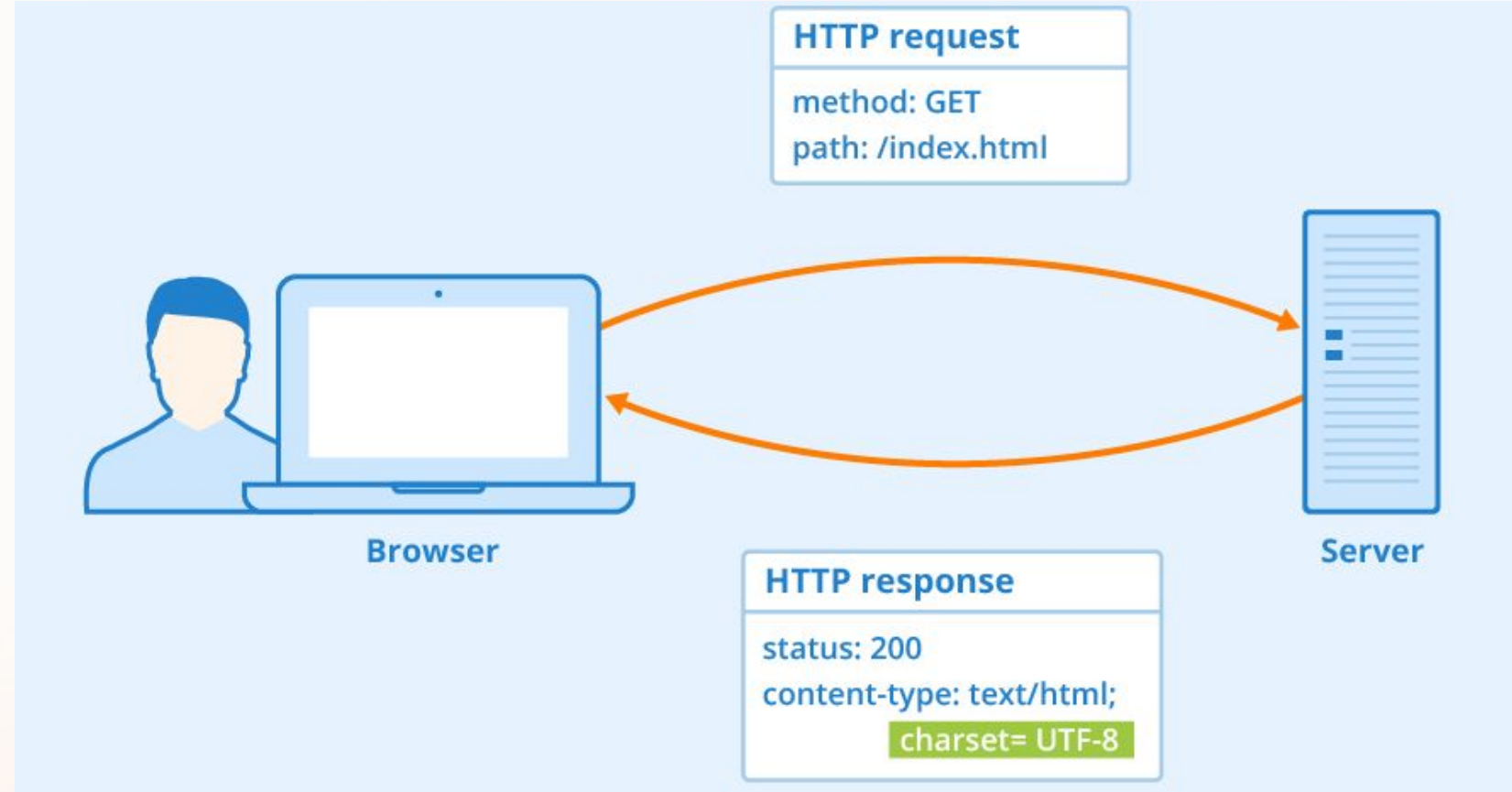
Protocolo HTTP: Funcionamento e Características

O protocolo HTTP é fundamental para o desenvolvimento de aplicações web. Ele permite a comunicação entre clientes e servidores, possibilitando o envio e recebimento de dados, como páginas da web, imagens e outros recursos.



Versões HTTP





HTTP: Solicitações e Respostas

1 Solicitações

Os clientes enviam solicitações HTTP para solicitar recursos, como abrir uma página da web ou enviar dados para o servidor.

2 Respostas

Os servidores respondem às solicitações com códigos de status, como 200 para "OK" ou 404 para "Not Found".

Requests

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;... )... Firefox/51.0
Accept: text/html,application/xhtml+xml,..., */*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345
```

```
-12656974
(more data)
```

start-
line

HTTP headers

empty
line

body

Responses

```
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
(more data)
```

Header HTTP

Request Header

O cabeçalho da solicitação inclui informações como o tipo de navegador, idiomas aceitos e cookies.

Response Header

O cabeçalho da resposta contém detalhes sobre o servidor, o tipo de conteúdo retornado e códigos de status.

Códigos de Status HTTP

200

Sucesso

A requisição foi bem sucedida.

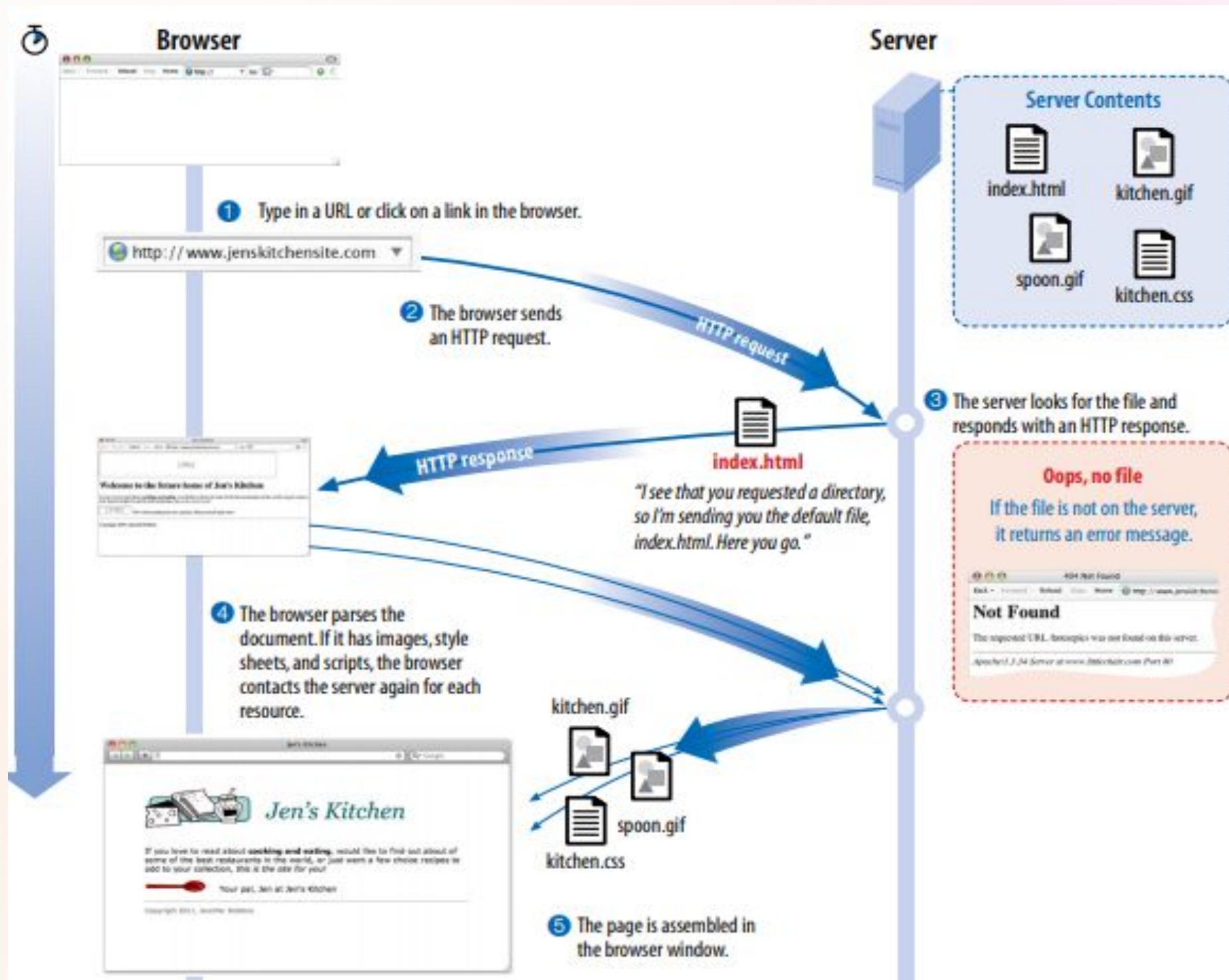
404

Página não encontrada

O servidor não encontrou a página solicitada.

Referência completa códigos de status

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>



HTTP x HTML

? Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

GET
POST
PUT

Métodos de Requisição HTTP

1 GET

O método GET é usado para solicitar dados de um recurso específico no servidor.

2 POST

O método POST é usado para enviar e processar dados para serem incluídos no corpo de uma solicitação.

3 PUT, DELETE, HEAD, etc

Métodos menos comuns..

Referência completa métodos HTTP
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

DevTools is now available in Portuguese! **Always** **Switch Chrome's language** **Switch DevTools to Portuguese** **Don't show again** X

Elements Console Sources **Network** Performance Memory Application Security Lighthouse Recorder >> ⚙️ ⋮ X

⏏️ ⛔ 🔍 ☐ Preserve log ☐ Disable cache No throttling ⌵ 📶 ⬆️ ⬇️ ⚙️

Filter ☐ Invert ☐ Hide data URLs ☐ Hide extension URLs

All Doc JS Fetch/XHR CSS Font **Img** Media Manifest WS Wasm Other ☐ Blocked response cookies ☐ Blocked requests ☐ 3rd-party requests

10 ms 20 ms 30 ms 40 ms 50 ms 60 ms 70 ms 80 ms 90 ms 100 ms 110

Name	Status	Type	Initiator	Size	Time	Waterfall
localhost	200	document	Other	6.6 kB	15 ms	
style.css	200	stylesheet	(index):7	(memory cac...	0 ms	
gifLogo1.gif	200	gif	style.css	(memory cac...	0 ms	
pngPlugin.png	200	png	style.css	(memory cac...	0 ms	
pngWrench.png	200	png	style.css	(memory cac...	0 ms	
pngFolderGo.png	200	png	style.css	(memory cac...	0 ms	
favicon.ico	200	x-icon	Other	(disk cache)	1 ms	

HTTP no Browser

F12 ou Botão direito -> Inspecionar

HTTPS e Segurança

Segurança na Transmissão

O HTTPS(**lançado em 1994**) usa criptografia para proteger os dados transmitidos entre cliente e servidor.

Identificação do Servidor

Os certificados SSL/TLS autenticam a identidade dos servidores, garantindo que os dados sejam enviados para o servidor correto.

Cookie HTTP

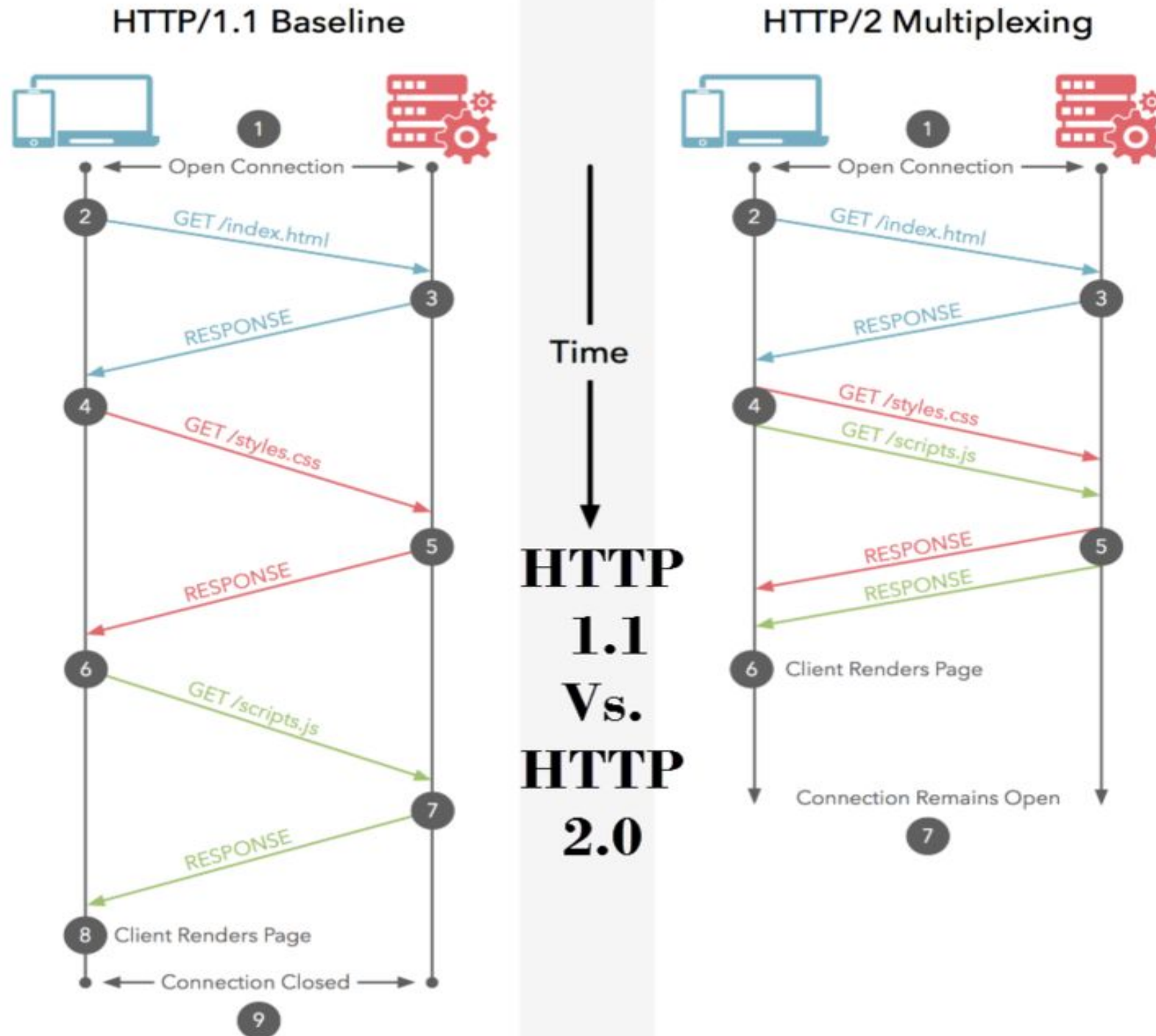
Envio e Armazenamento

Os cookies são usados para armazenar dados no navegador do cliente e enviar informações de volta para o servidor em requisições futuras.

Privacidade e Segurança

Os cookies podem ser configurados com restrições para proteger a privacidade do usuário e evitar ataques de segurança.

HTTP 1 vs HTTP2



HTTP/3 e o Futuro

1

Desempenho

O HTTP/3 visa melhorar o desempenho em ambientes de rede com alta latência e perda de pacotes.

2

Segurança

As melhorias de segurança no HTTP/3 oferecem proteção adicional contra ameaças online.

3

Implementação Gradual

A migração para o HTTP/3 está em andamento, impulsionando o desenvolvimento de aplicativos web mais rápidos e seguros.