

Apresentação HTTP

- Introdução
- Conceitos
- Exemplos de Métodos
- Encapsulamento
- Servidores e Versões HTTP
- Exemplos com Apache
- Laboratório
- Atividade extra

Intro

- Protocolo concebido em 1989 no CERN.
- Projetado para habilitar a comunicação de aplicações na rede Internet (WWW).

Version	Year introduced	Current status	Usage in August 2024	Support in August 2024
HTTP/0.9	1991	Obsolete	0	100%
HTTP/1.0	1996	Obsolete	0	100%
HTTP/1.1	1997	Standard	33.8%	100%
HTTP/2	2015	Standard	35.3%	66.2%
HTTP/3	2022	Standard	30.9%	30.9%

Wikipedia

- W3C é o grupo de trabalho que determina as padronizações HTTP, bem como da WWW.

Conceitos

- Protocolo em nível de aplicação.
- Troca de mensagens individuais Request-Response.
- Protocolo *stateless* (logo a necessidade de *cookies*).
- Requisição: método, caminho, versão, cabeçalho corpo(opcional).
- Resposta: versão, status, msg status, cabeçalho e corpo(opcional).
- Conceitos de URL (e.g.: <http://www.rzuolo.com>) e URI (e.g.: <urn:oid:1.3.6.1.4.1.1466.20036>).
- Métodos: GET, PUT, POST, DELETE, etc.

Conceitos

- Códigos de resposta

1xx informational

2xx successful

3xx redirection

4xx client error

5xx server error

HTTP status ranges in a nutshell:

1xx: hold on

2xx: here you go

3xx: go away

4xx: you messed up

5xx: I messed up

ProgrammerHumor.io



404
Not Found

Exemplos de Métodos

```
#> telnet www.rzuolo.com 80
```

```
GET /2025/ HTTP/1.0
```

```
HTTP/1.1 200 OK
```

```
Server: nginx/1.21.0
```

```
Date: Sun, 30 Mar 2025 09:08:49 GMT
```

```
Content-Type: text/html
```

```
Content-Length: 1187
```

```
Connection: close
```

```
Last-Modified: Thu, 27 Feb 2025 16:18:37 GMT
```

```
ETag: "67c0905d-4a3"
```

```
Accept-Ranges: bytes
```

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
body
```

```
{
```

```
    color:green;
```

```
    bgcolor:black;
```

```
text-align:left;
```

```
font-size:16pt;
```

```
...
```



```
curl -X GET https://dadjokes.online/
```

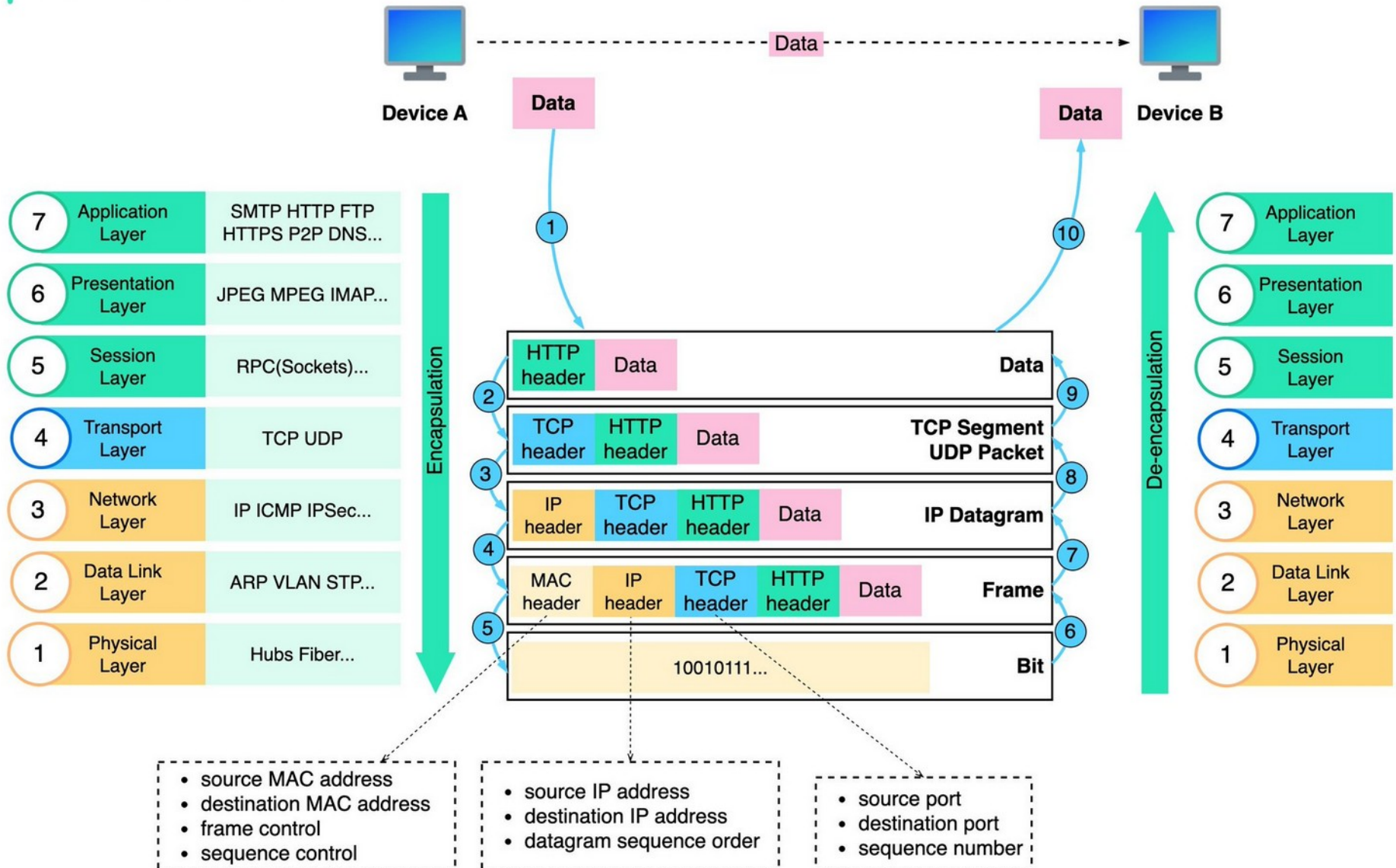
Encapsulamentos

- Na literatura o cenário mais comum é o da utilização de TCP e TLS nas camadas adjacentes.
- Faz o transporte de diversos tipos/formatos de dados, definidos em RFCs, como JSON, HTML, XML, binário, base64, etc.
- Protocolos no payload podem ser os mais diversos: JWT, SAML, DNS, SSH, WebDav, etc.
- Mais recentemente o uso de QUIC (UDP) tem se tornado predominante.
- Outro padrão é o uso de HTTPS em detrimento de HTTP.

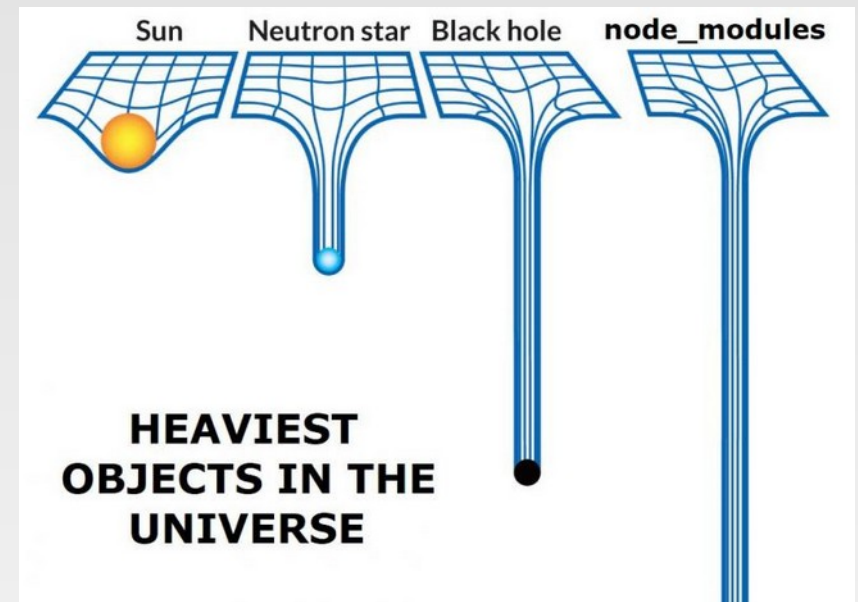
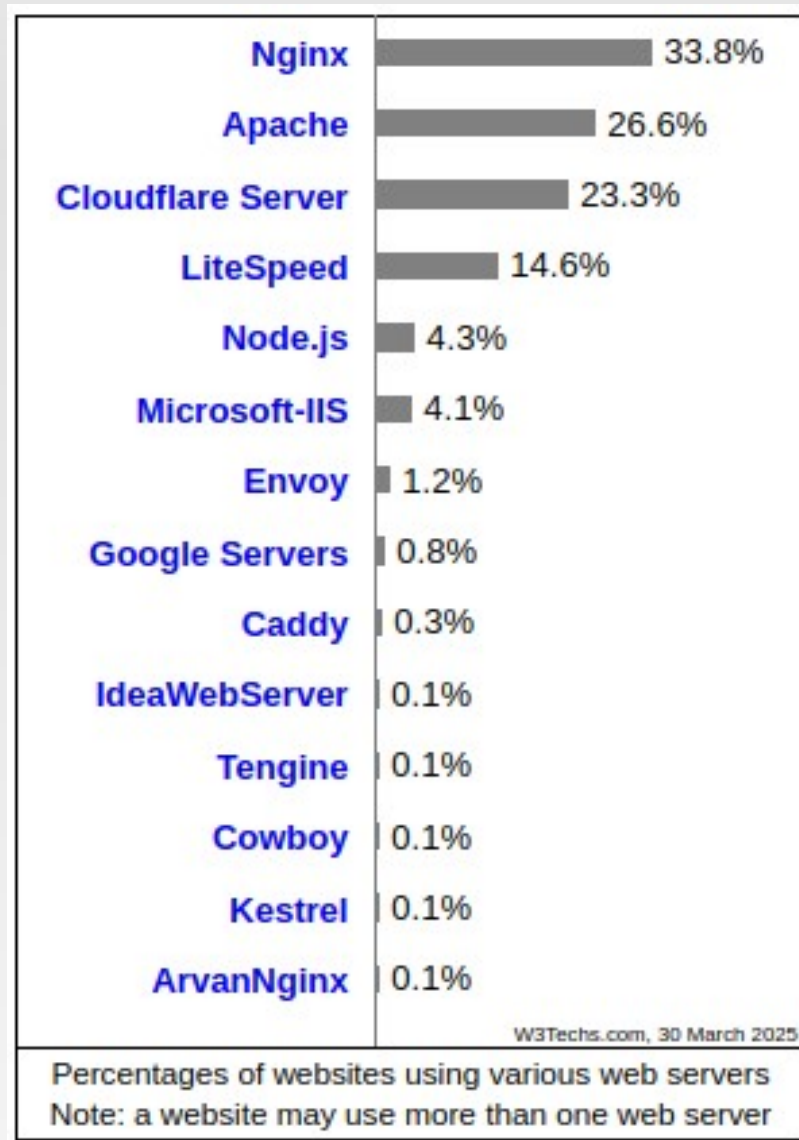
Encapsulamentos

What is OSI model

blog.bytebytego.com



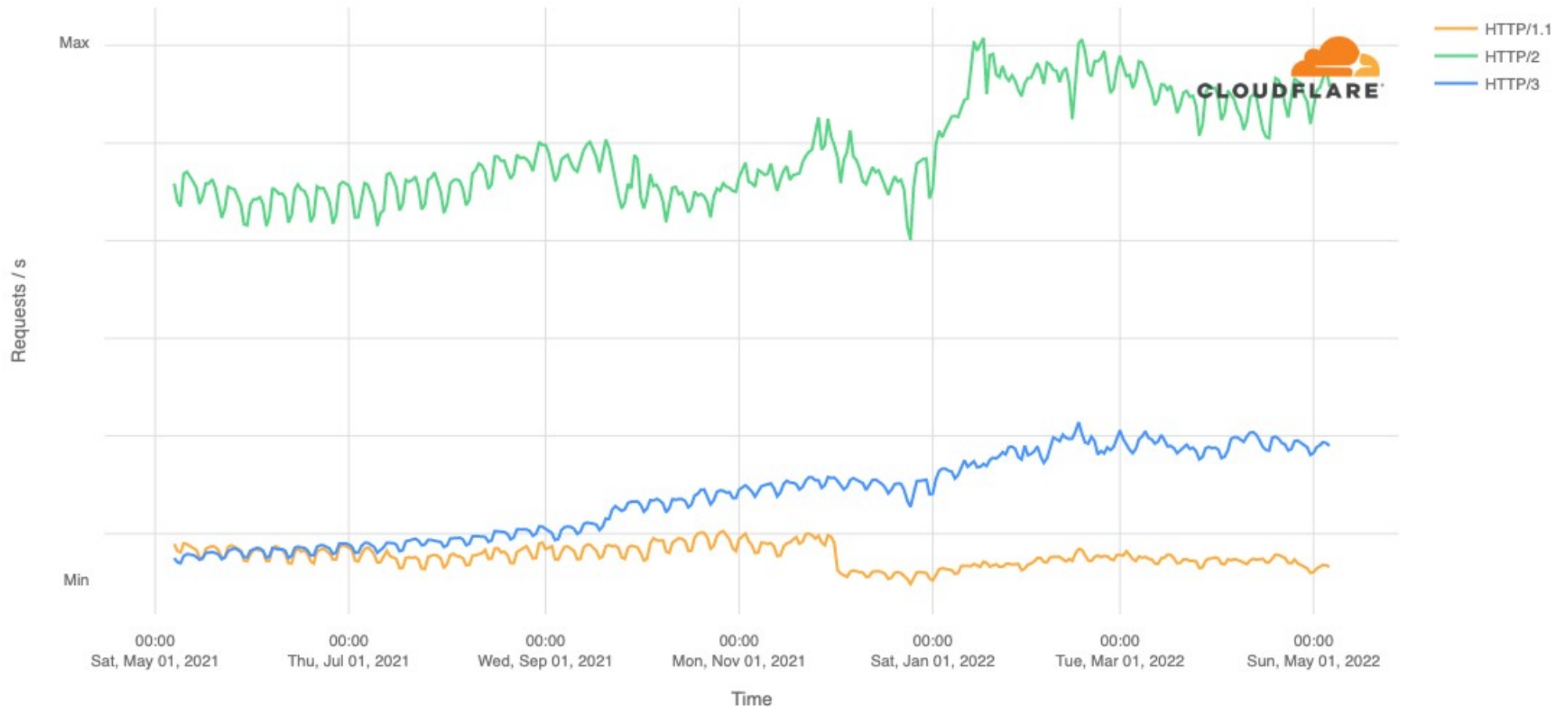
Servidores



https://w3techs.com/technologies/overview/web_server

Versões HTTP

Number of requests by HTTP version (Worldwide)



<https://blog.cloudflare.com/cloudflare-view-http3-usage/>

Exemplos com Apache

Access control

```
<Directory "/var/www/admin">
```

```
    Order Allow,Deny
```

```
    Allow from .ualg.pt
```

```
    Deny from all
```

```
</Directory>
```

```
...
```

```
<Directory "/var/www/secure">
```

```
    Order Allow,Deny
```

```
    Allow from 192.168.1.100
```

```
    Allow from 10.0.0.5
```

```
    Deny from all
```

```
</Directory>
```

```
<Directory "/var/www/public">
```

```
    Order Deny,Allow
```

```
    Deny from 203.0.113.45
```

```
    Deny from 198.51.100.0/24
```

```
    Allow from all
```

```
</Directory>
```

```
...
```

```
<Directory "/var/www/secure">
```

```
    Require ip 192.168.1.100 10.0.0.5
```

```
</Directory>
```

Exemplos com Apache

Virtual host

```
<VirtualHost *:80>

    ServerName ualg.pt

    DocumentRoot /var/www/ualg.pt/public_html

    <FilesMatch \.php$>
        SetHandler "proxy:unix:/run/php/php8.1-fpm.sock|fcgi://localhost"
    </FilesMatch>

    <Directory /var/www/ualg.pt/public_html>
        Options -Indexes +FollowSymLinks

        AllowOverride All

        Require all granted
    </Directory>
</VirtualHost>
```

Exemplos com Apache

Htaccess

```
AuthType Basic  
AuthName "Restricted Area"  
AuthUserFile /var/www/.htpasswd  
Require valid-user
```

Laboratório

- Executar alguns métodos HTTP de forma interativa .
- Configurar e rodar um servidor Apache com capacidade de controle de acesso, virtual hosts e área pública de usuários do sistema.

Atividade extra

- Faça o exercício extra em
<http://rzuolo.com/2025/grs/atividade-extra-09-HTTP.pdf>

The End

