

НТТ

2

2

2



HTTP

Fundamentos

HTTP

Fundamentos

HTTPS - La web segura

HTTP

Fundamentos

HTTPS - La web segura

Dominios web

HTTP

Fundamentos

HTTPS - La web segura

Dominios web

Cliente servidor

HTTP

Fundamentos

HTTPS - La web segura

Dominios web

Cliente servidor

Parámetros de las peticiones

HTTP

Fundamentos

HTTPS - La web segura

Dominios web

Cliente servidor

Parámetros de las peticiones

Servicios REST

HTTP

Fundamentos

HTTPS - La web segura

Dominios web

Cliente servidor

Parámetros de las peticiones

Servicios REST

Versión 2

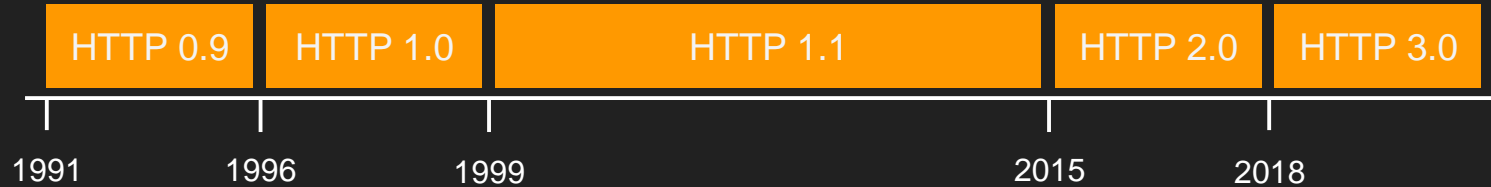
Harland Lohora

HTTP

Fundamentos

HTTP

Es un protocolo de Internet el cual nos permite la transferencia de información.



NAVEGADORES



CLIENTE - SERVIDOR



CLIENTE - SERVIDOR



CLIENTE - SERVIDOR



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

INDEPENDIENTE DE LA PLATAFORMA



HTTP
(PROTOCOLO)



HTTP

La web segura

CLIENTE - SERVIDOR



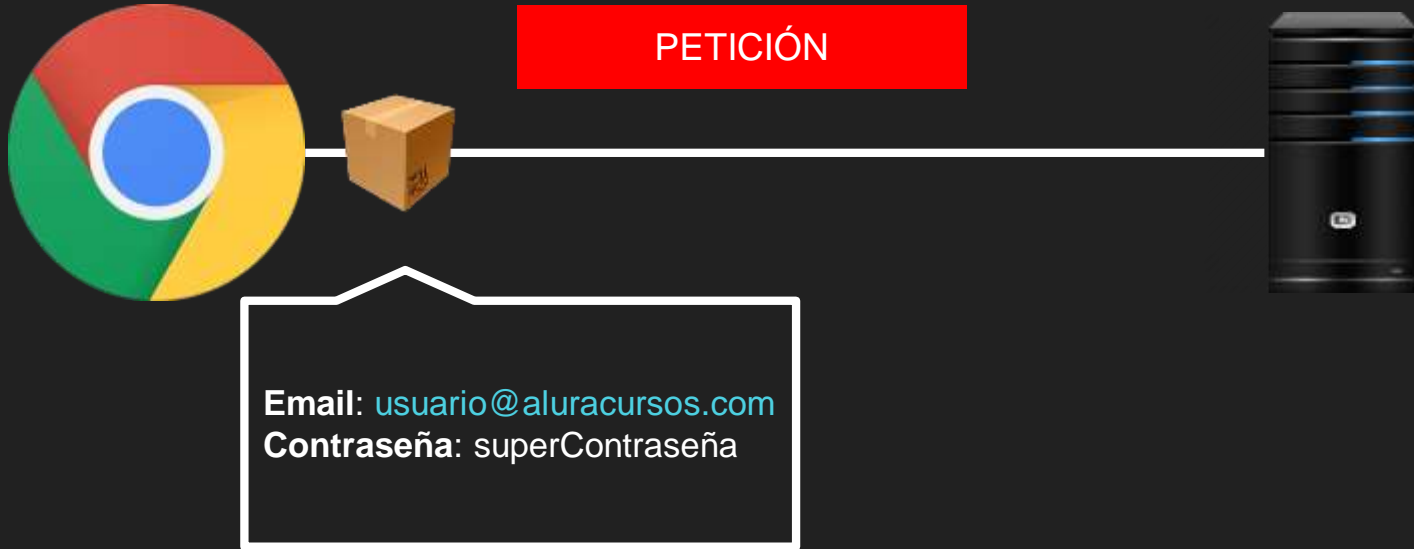
CLIENTE - SERVIDOR



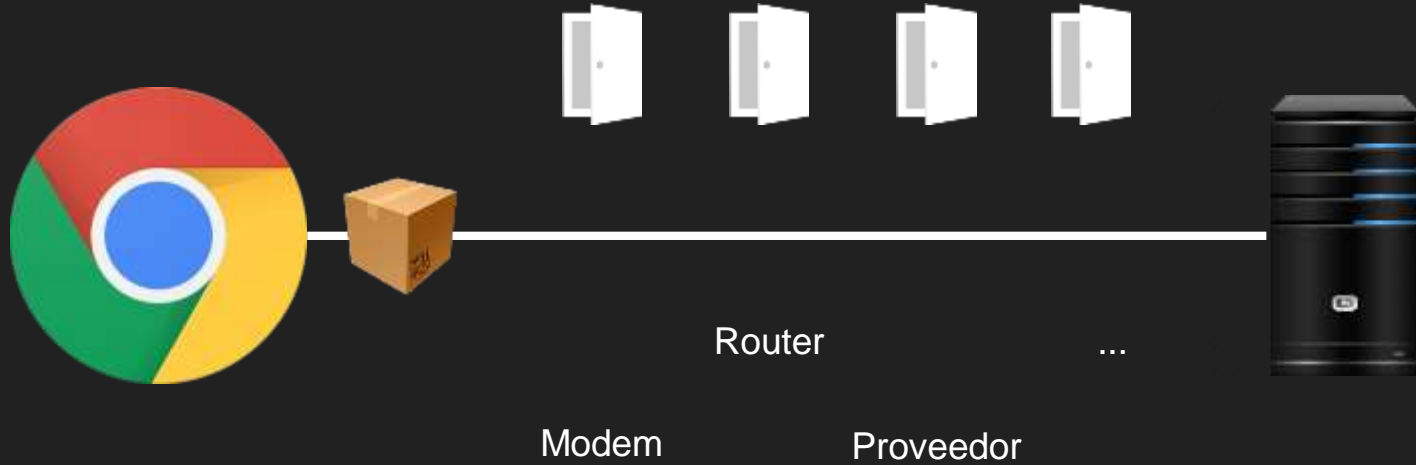
CLIENTE - SERVIDOR



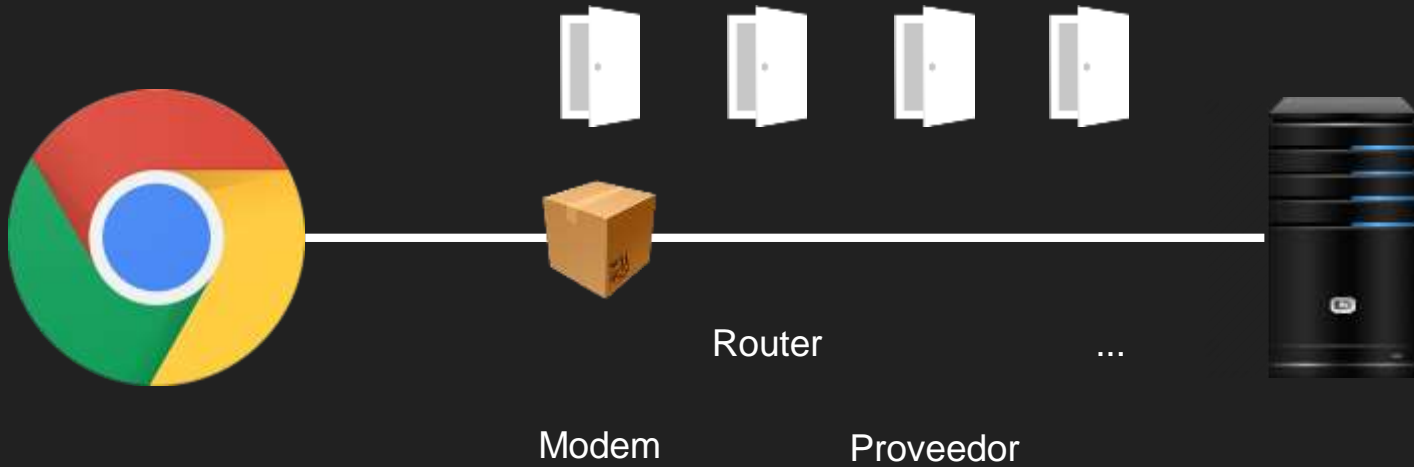
CLIENTE - SERVIDOR



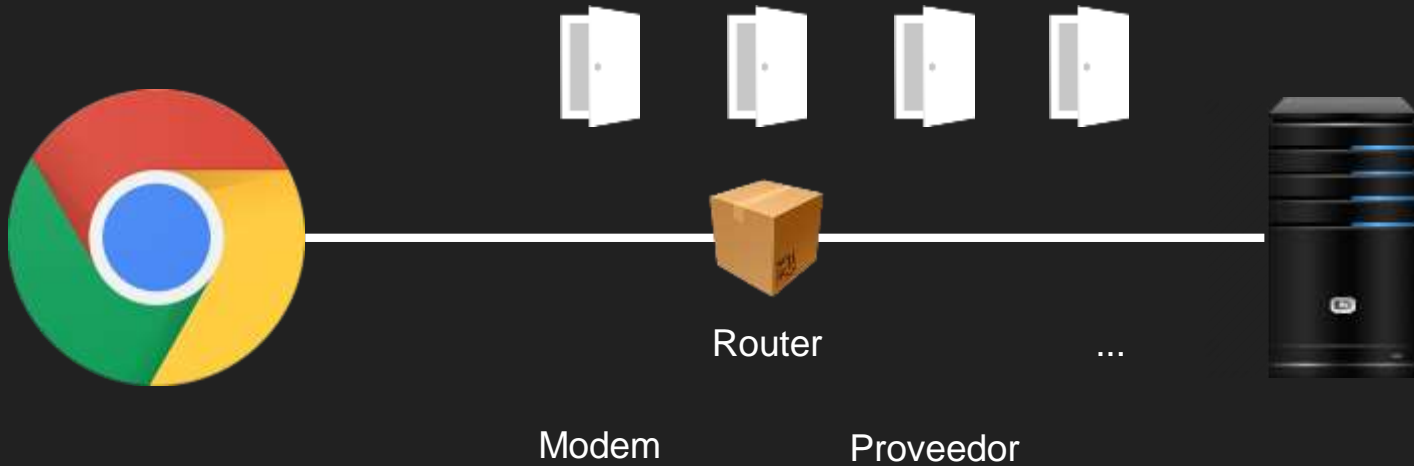
CLIENTE - SERVIDOR



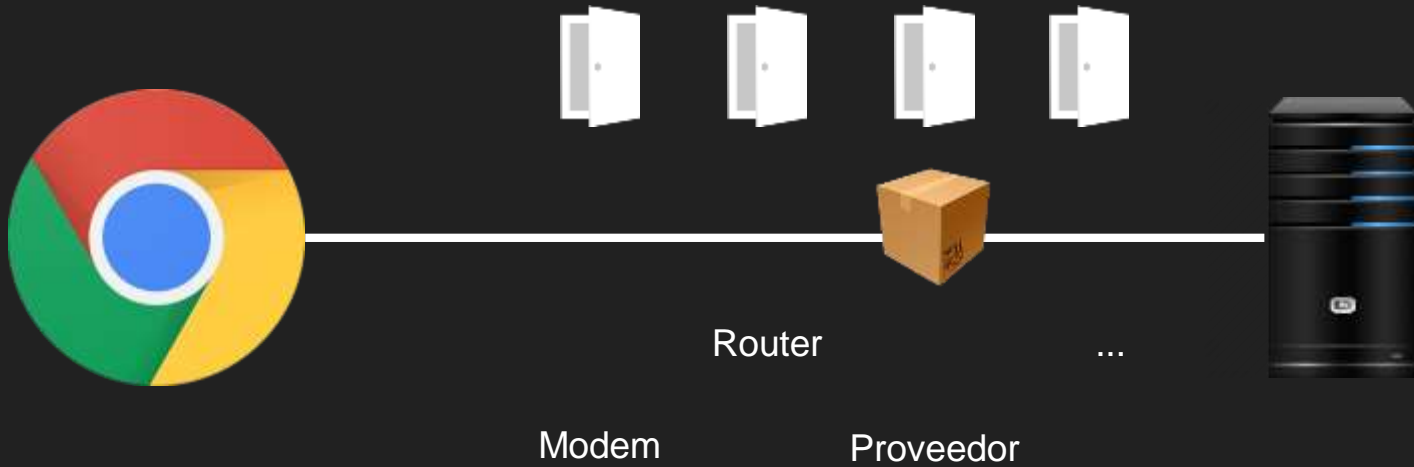
CLIENTE - SERVIDOR



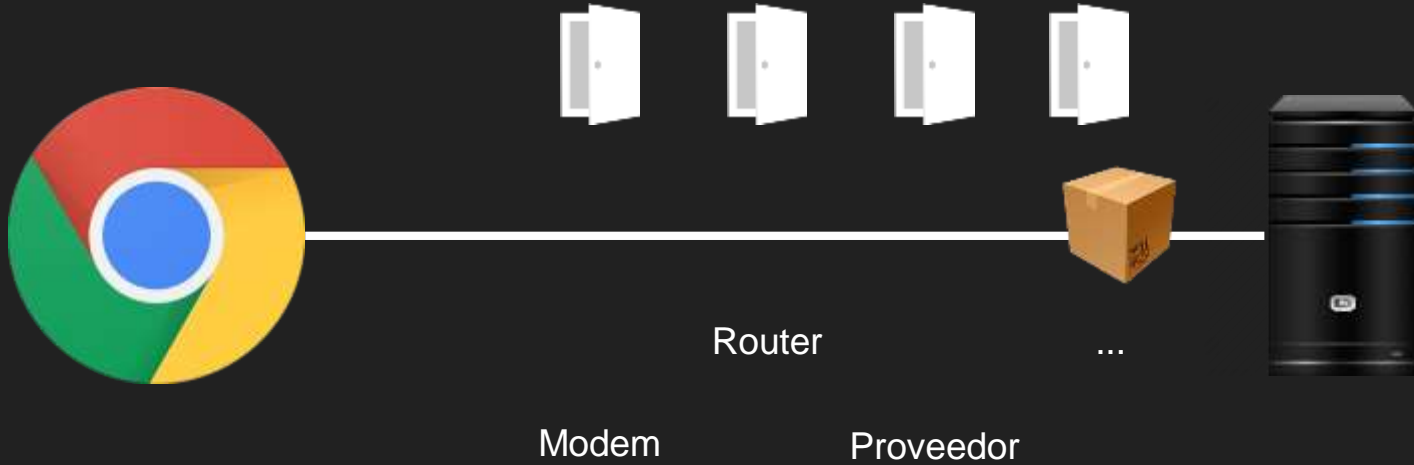
CLIENTE - SERVIDOR



CLIENTE - SERVIDOR



CLIENTE - SERVIDOR

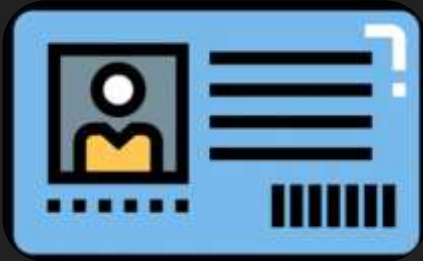


HTTPS

HTTP + TLS

HyperText Transfer Protocol + Transport Layer Security

Identidad



Certificado digital



Certificado digital



Llave pública





Llave pública



Llave privada

CLIENTE - SERVIDOR



```
YTA1VMRYwFAYDVQQKEw1HZW9UcnVzII  
EBDCCAuygAwIBAgIDAjppMA0GCSqGSI  
b3DQEBBQUAMEIxCzAJBgNVBAdCBJbmM  
uMRswGQYDVQQDExJHZW9UcnVzdCBHbG  
9i
```

CLIENTE - SERVIDOR



```
YTA1VMRYwFAYDVQQKEw1HZW9UcnVzII  
EBDCCAuygAwIBAgIDAjppMA0GCSqGSI  
b3DQEBBQUAMEIxCzAJBgNVBAdCBJbmM  
uMRswGQYDVQQDExJHZW9UcnVzdCBHbG  
9i
```


CLIENTE - SERVIDOR



```
YTA1VMRYwFAYDVQQKEw1HZW9UcnVzII  
EBDCCAuygAwIBAgIDAjppMA0GCSqGSI  
b3DQEBBQUAMEIxCzAJBgNVBAdCBJbmM  
uMRswGQYDVQQDExJHZW9UcnVzdCBHbG  
9i
```

CLIENTE - SERVIDOR



```
YTA1VMRYwFAYDVQKEw1HZW9UcnVzII  
EBDCCAuygAwIBAgIDAjppMA0GCSqGSI  
b3DQEBBQUAMEIxCzAJBgNVBAdCBJbmM  
uMRswGQYDVQQDExJHZW9UcnVzdCBHbG  
9i
```

CLIENTE - SERVIDOR

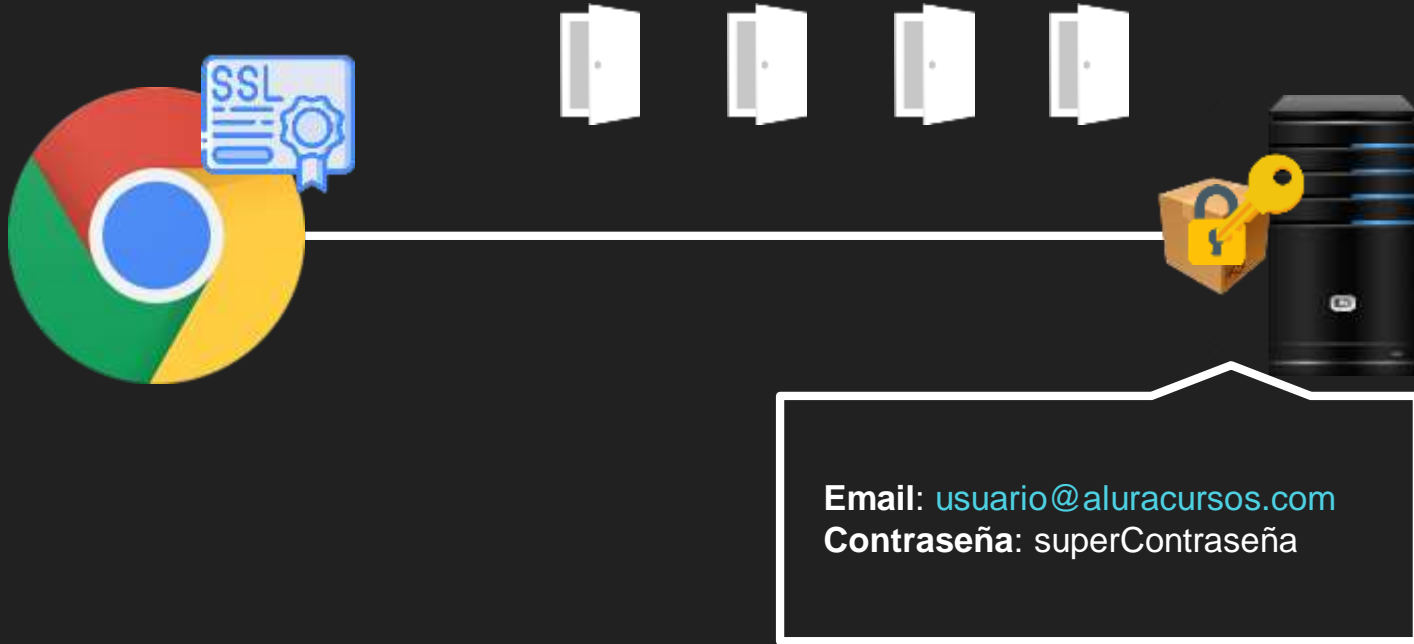


```
YTA1VMRYwFAYDVQKEw1HZW9UcnVzII  
EBDCCAuygAwIBAgIDAjppMA0GCSqGSI  
b3DQEBBQUAMEIxCzAJBgNVBAdCBJbmM  
uMRswGQYDVQQDExJHZW9UcnVzdCBHbG  
9i
```

CLIENTE - SERVIDOR



CLIENTE - SERVIDOR



HTTPS

- Web segura
- Certificado digital (identidad)
- Llaves (pública y privada)

HTTP

Dominios web

DOMINIOS WEB

<https://www.aluracursos.com>

DOMINIOS WEB

<https://www.aluracursos.com>

Protocolo

DOMINIOS WEB

<https://www.aluracursos.com>

Subdominio

DOMINIOS WEB

<https://www.aluracursos.com>

Nombre del dominio

DOMINIOS WEB

<https://www.aluracursos.com>



Extensión

DOMINIOS WEB

172.217.162.206



172.217.162.206

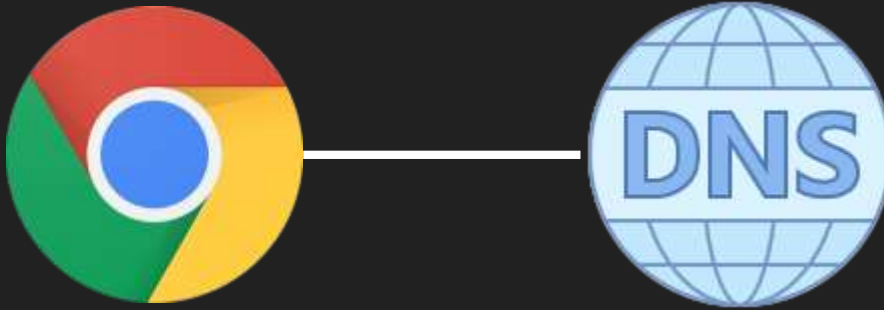
facebook

69.171.250.35

DNS

Domain Name Service

CLIENTE - SERVIDOR



<http://www.facebook.com>

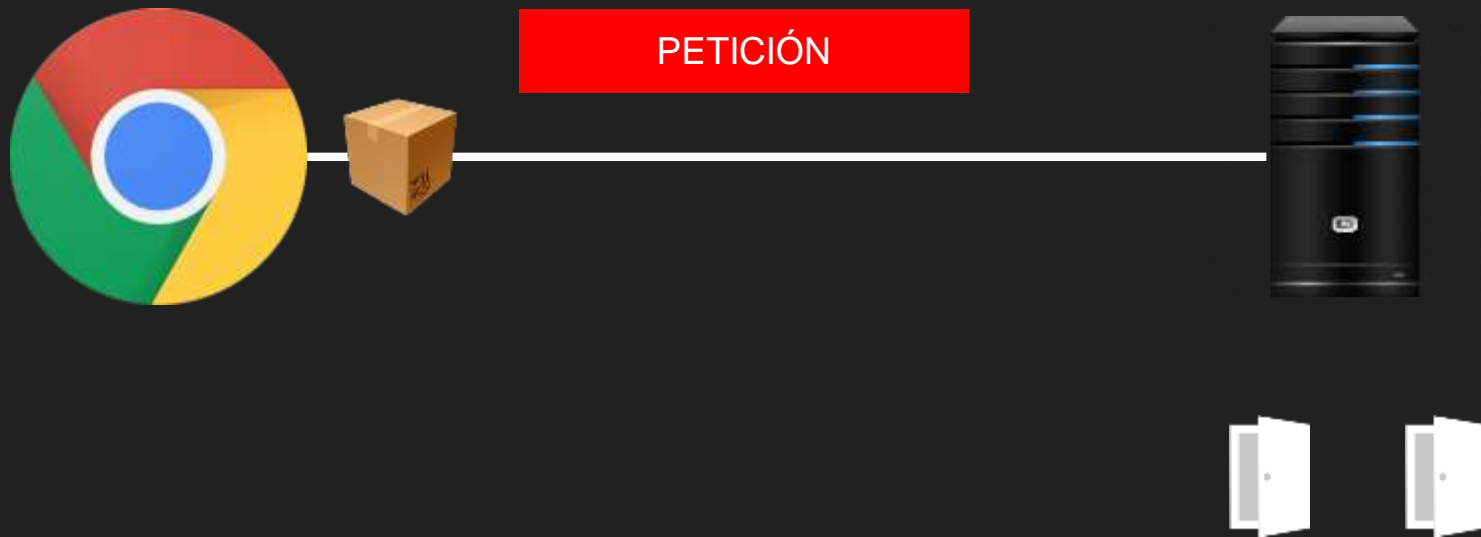


CLIENTE - SERVIDOR

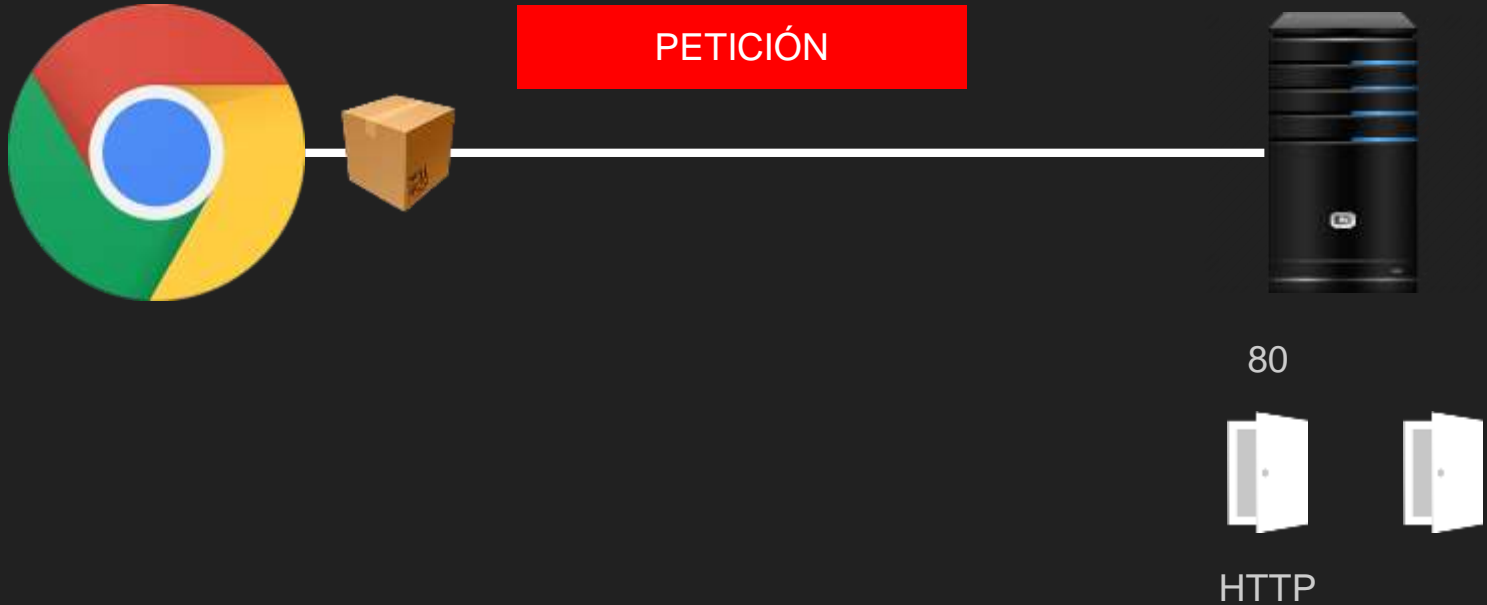


Puertos

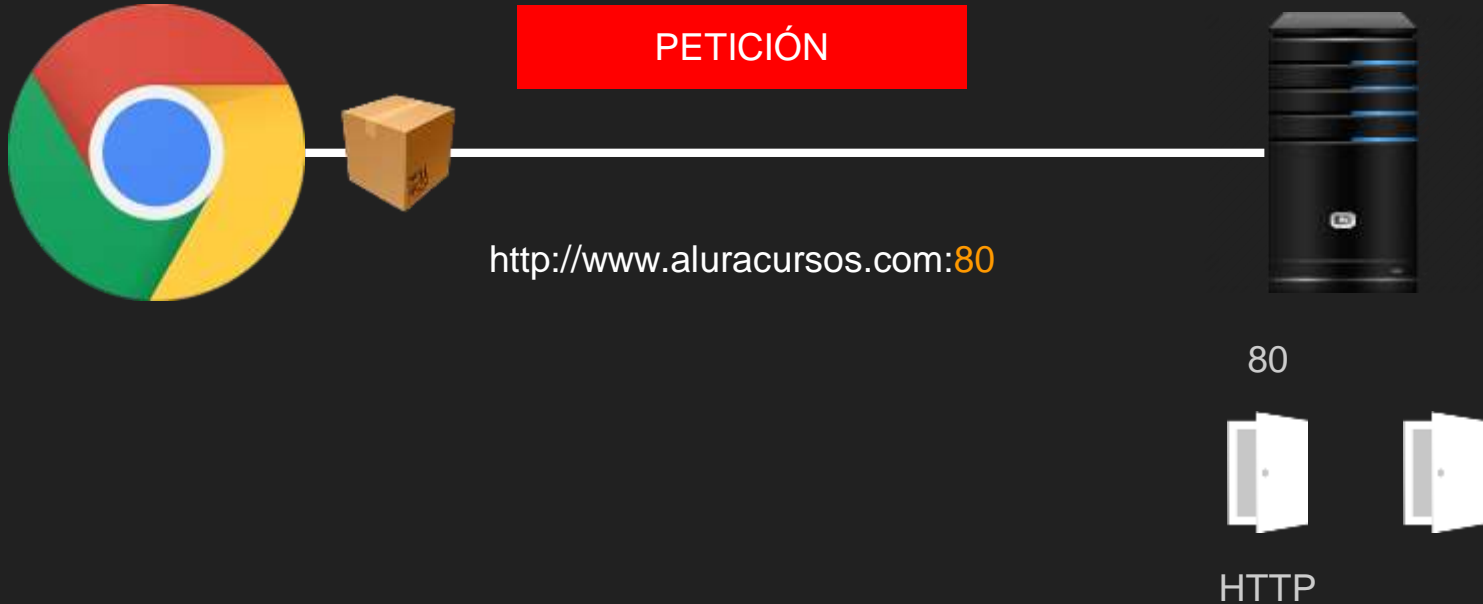
CLIENTE - SERVIDOR



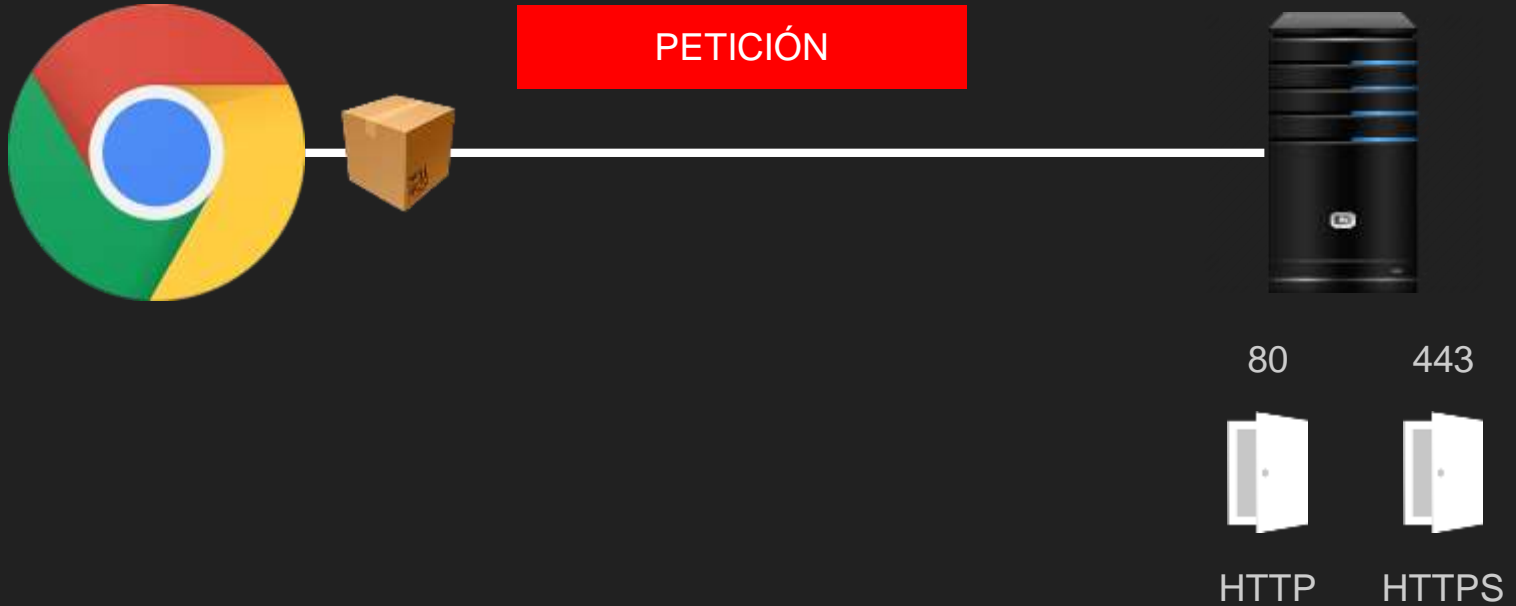
CLIENTE - SERVIDOR



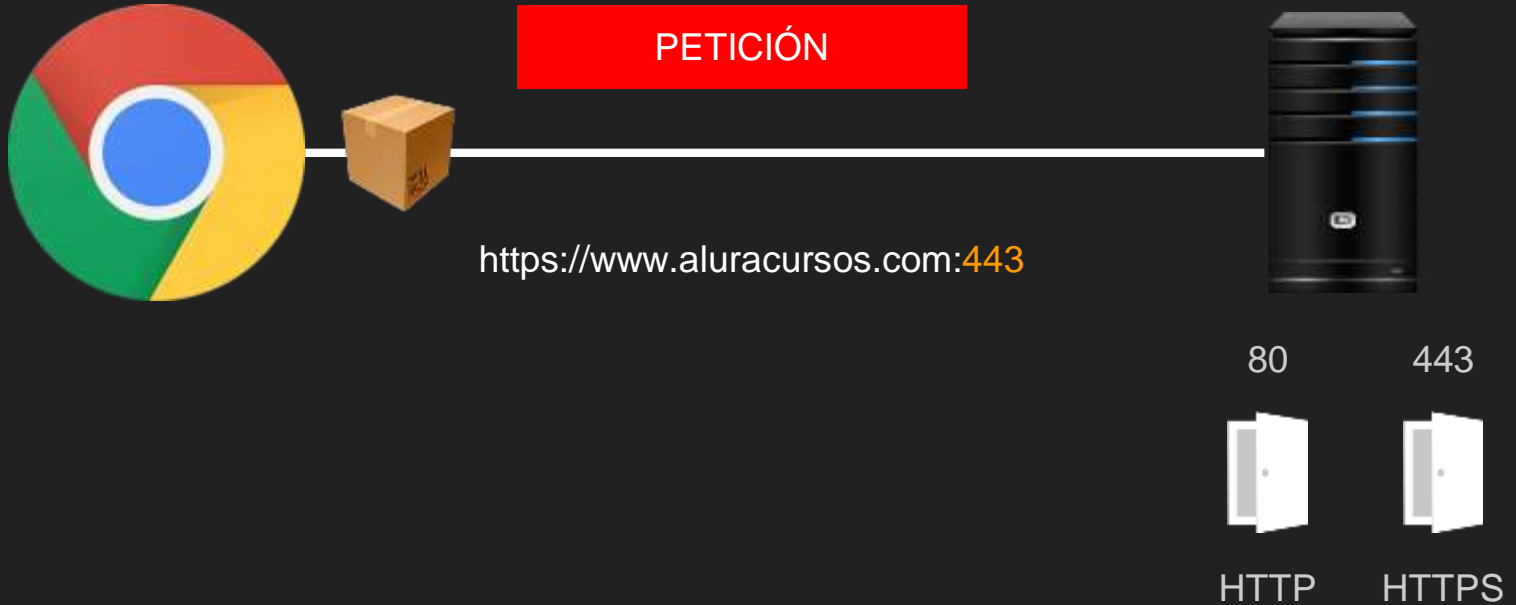
CLIENTE - SERVIDOR



CLIENTE - SERVIDOR



CLIENTE - SERVIDOR



RECURSOS

RECURSOS

<https://app.aluracursos.com/course/html5-css3-primera-pagina-web>

Dirección web

RECURSOS

<https://app.aluracursos.com/course/html5-css3-primera-pagina-web>

Recurso

HTTP

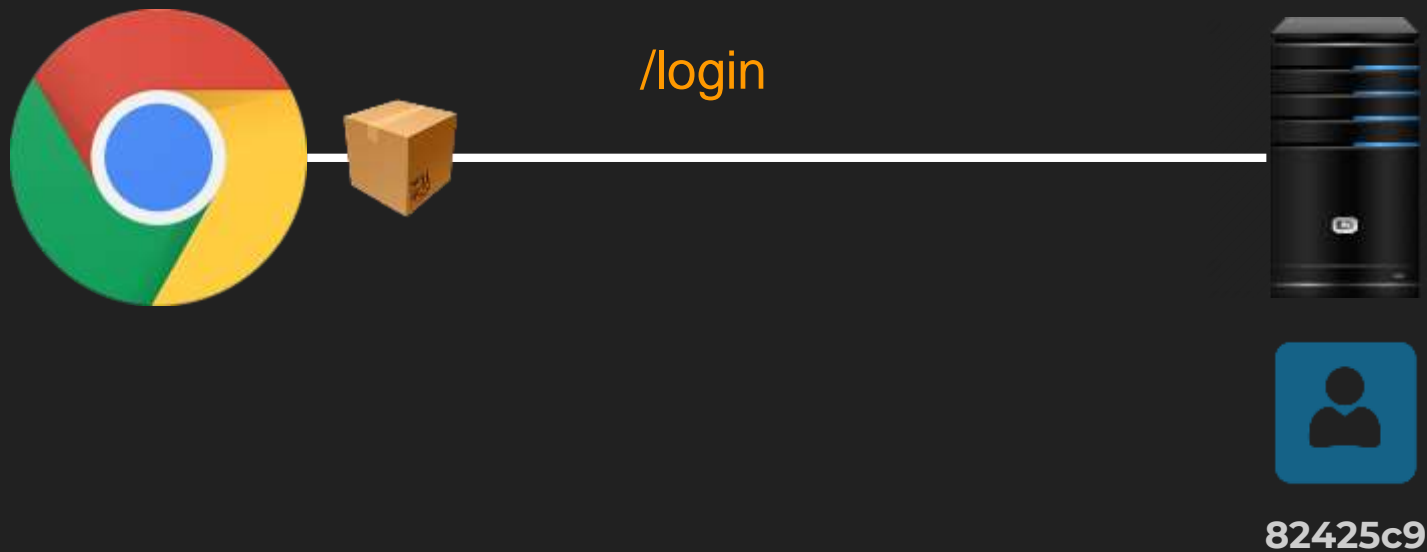
El cliente pide el servidor responde



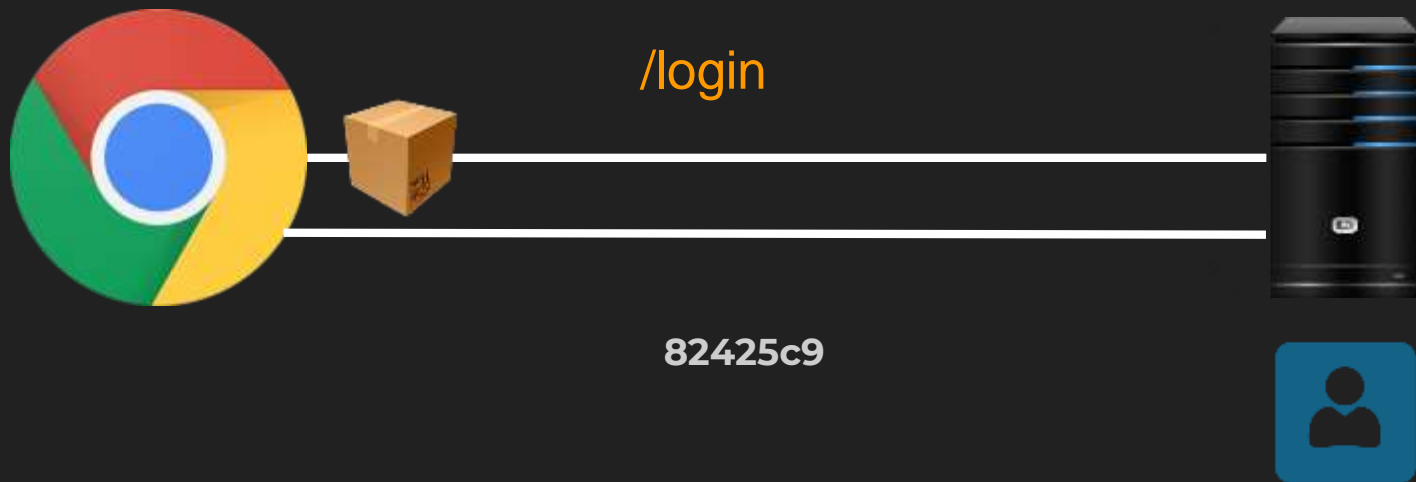


STATELESS

CLIENTE - SERVIDOR



CLIENTE - SERVIDOR



SESIONES

COOKIES



HTTP(Petición respuesta)

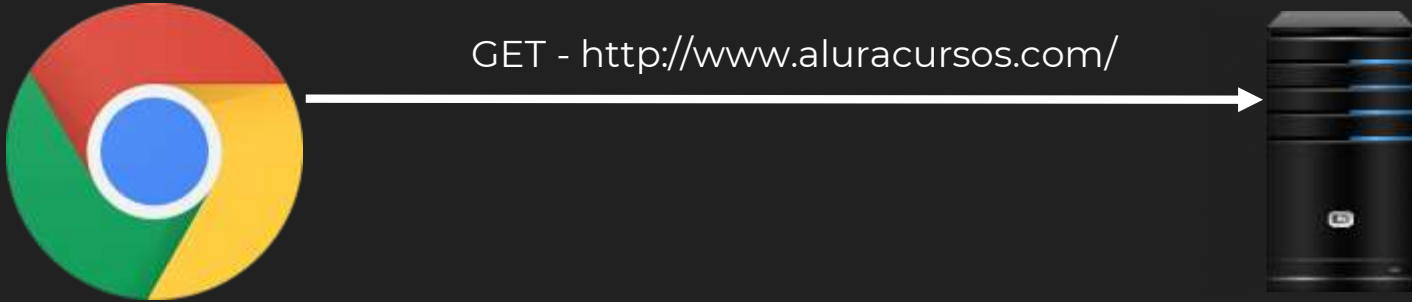
- El protocolo HTTP sigue el modelo petición-respuesta
- Una petición necesita toda la información necesaria a mandar al servidor
- HTTP es STATELESS(No existe un estado en las peticiones)
- Los navegadores utilizan sesiones para almacenar un identificador de usuario
- Las cookies son archivos llave-valor que almacenan datos



HTTP

Depurando las peticiones

CLIENTE - SERVIDOR



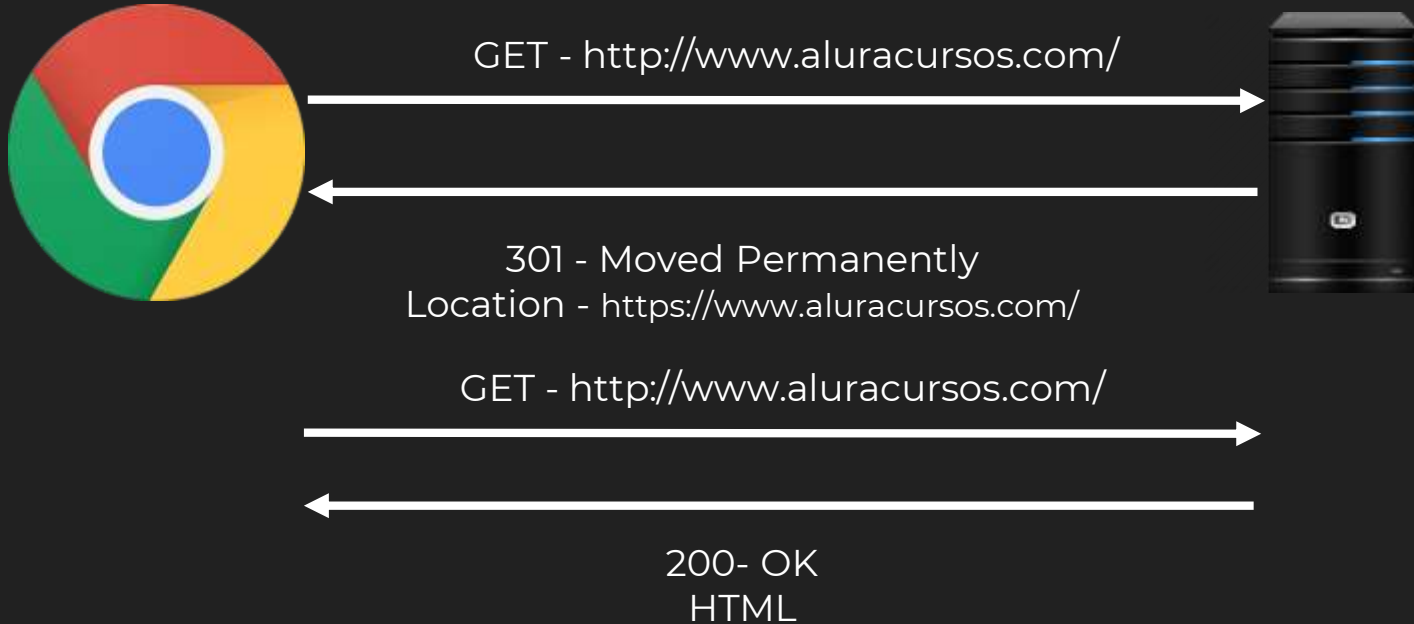
CLIENTE - SERVIDOR



CLIENTE - SERVIDOR

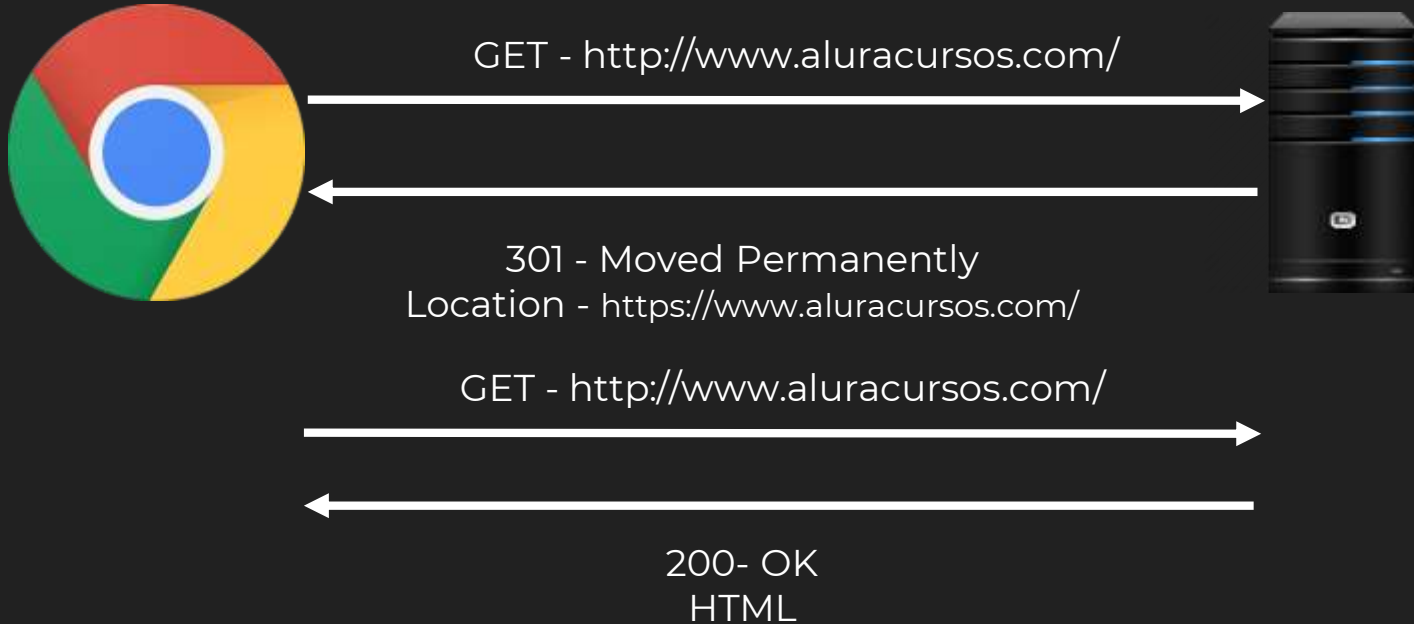


CLIENTE - SERVIDOR



STATUS CODE

CLIENTE - SERVIDOR



STATUS CODE

100 - CONTINUE

200 - OK

301 - MOVED PERMANENTLY

404 - NOT FOUND

500 - INTERNAL SERVER ERROR

STATUS CODE

1XX - INFORMATIVOS

2XX - RESPUESTA SATISFACTORIA

3XX - REDIRECCIONES

4XX - ERROR DEL CLIENTE

5XX - ERROR DEL SERVIDOR

Depurando peticiones

- La consola del navegador
- Método HTTP GET
- Encabezados de la respuesta
- Códigos de respuesta(Status Code)

HTTP

Parámetros de la petición

CLIENTE - SERVIDOR



STATUS CODE

100 - CONTINUE

200 - OK

301 - MOVED PERMANENTLY

404 - NOT FOUND

500 - INTERNAL SERVER ERROR

STATUS CODE

1XX - INFORMATIVOS

2XX - RESPUESTA SATISFACTORIA

3XX - REDIRECCIONES

4XX - ERROR DEL CLIENTE

5XX - ERROR DEL SERVIDOR

ENVIANDO PARÁMETROS

MÉTODOS - PARÁMETROS

GET - Envía los parámetros en la url

POST - Envía los parámetros en el cuerpo de la solicitud

PARÁMETROS

- Son utilizados para definir detalles de la búsqueda
- GET - los parámetros son enviados en la URL(usando ? para iniciar la sección de params y & para concatenarlos)
- POST - los parámetros son enviados en el cuerpo de la petición
- Existen otros métodos HTTP (DELETE; PUT, etc)

HTTP

Servicios REST

CLIENTE - SERVIDOR



INDEPENDIENTE DE LA PLATAFORMA



HTTP
(PROTOCOLO)



MÉTODOS

- **GET** - Obtener información del servidor(parámetros en URL)
- **POST** - Enviar datos(parámetros en el cuerpo de la petición)
- **DELETE** - Eliminar un recurso
- **PUT** - Actualizar un recurso

CLIENTE - SERVIDOR



PETICIÓN



RESPUESTA



CLIENTE - SERVIDOR



GET - <https://alurafood.com/api/foods>



CLIENTE - SERVIDOR



GET - <https://alurafood.com/api/foods>



200 - OK
HTML - ?



```
index.html x
~/Documents/Alura/exampleOoca/index.html
index.html > html > body > div.foodsSpace > div.food
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>AluraFood</title>
8 </head>
9 <body>
10   <div class="foodsSpace">
11     <div class="food">
12       <div>
13         <h4>Pizza</h4>
14         <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Corporis, cum reprehenderit consequatur,
15       </div>
16       <div>
17         
18       </div>
19     </div>
20     <div class="food">
21       <div>
22         <h4>Sushi</h4>
23         <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Corporis, cum reprehenderit consequatur,
24       </div>
25       <div>
26         
27       </div>
28     </div>
29     <div class="food">
30       <div>
31         <h4>Bistecca</h4>
32         <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Corporis, cum reprehenderit consequatur,
33       </div>
34       <div>
35         
36       </div>
37     </div>
38   </div>
39 </body>
```

XML (eXtensible Markup Language)

```
1 <foodlist>
2   <food>
3     <name>Pizza</name>
4     <description>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Optio
5     <image>https://alurafood.com/img/pizza</image>
6   </food>
7   <food>
8     <name>Sushi</name>
9     <description>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Optio
10    <image>https://alurafood.com/img/sushi</image>
11  </food>
12  <food>
13    <name>Bistecca</name>
14    <description>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Optio
15    <image>https://alurafood.com/img/bistecca</image>
16  </food>
17 </foodlist>
```

JSON(JavaScript Object Notation)

{-} res.json x

{-} res.json > ...

```
1  {
2    "foods": [
3      {
4        "name": "Pizza",
5        "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Optio odit laborum ipsa
6        "image": "https://alurafood.com/img/pizza"
7      },
8      {
9        "name": "Sushi",
10       "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Optio odit laborum ipsa
11       "image": "https://alurafood.com/img/sushi"
12     },
13     {
14       "name": "Bistecca",
15       "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit. Optio odit laborum ipsa
16       "image": "https://alurafood.com/img/bistecca"
17     }
18   ]
19 }
```


CLIENTE - SERVIDOR



GET - <https://alurafood.com/api/foods>
Accept: application/json



CLIENTE - SERVIDOR



GET - <https://alurafood.com/api/foods>
Accept: application/json



200 - OK
Content-Type: application/json

```
200 OK
{
  "foods": [
    {
      "name": "Pizza",
      "description": "Lorem ipsum dolor sit amet.",
      "image": "https://alurafood.com/img/pizza"
    },
    {
      "name": "Sushi",
      "description": "Lorem ipsum dolor sit amet.",
      "image": "https://alurafood.com/img/sushi"
    },
    {
      "name": "Bistecca",
      "description": "Lorem ipsum dolor sit amet.",
      "image": "https://alurafood.com/img/bistecca"
    }
  ]
}
```

¿Qué es REST?

URI

<http://alurafood.com/api/foods>

<http://alurafood.com/api/foods/1>

<http://alurafood.com/api/foods/1/description>

<http://alurafood.com/api/user/10/orders>

URI + MÉTODOS

<http://alurafood.com/api/foods> - GET // Obtener todas las comidas

<http://alurafood.com/api/foods> - POST // Crear una nueva comida

<http://alurafood.com/api/foods/1> - PUT // Actualizar la comida 1

<http://alurafood.com/api/foods/1> - DELETE // Eliminar la comida 1

<http://alurafood.com/api/foods/1> - GET // Obtener la comida 1

CRUD

- CREATE - POST
- READ - GET
- UPDATE - PUT
- DELETE - DELETE

REST



*REpresentational State Transfer

REST

RECURSO
(URI)



*REpresentational State Transfer

REST

RECURSO
(URI)



OPERACIONES
(GET/POST/PUT/DELETE
)

*REpresentational State Transfer

REST

RECURSO
(URI)



OPERACIONES
(GET/POST/PUT/DELETE
)

REPRESENTACIÓN
(JSON/XML/HTML)

*REpresentational State Transfer

REST

RECURSO
(URI)



HTTP

OPERACIONES
(GET/POST/PUT/DELETE
)

REPRESENTACIÓN
(JSON/XML/HTML)

*REpresentational State Transfer

REST

- Es una arquitectura para la comunicación entre aplicaciones
- Aprovecha la estructura que HTTP proporciona
- Los recursos son definidos vía URI
- Operaciones con los métodos (GET/POST/PUT/DELETE)
- Encabezados(Accept/Content-Type) para especificar la representación (JSON,XML,...)

MÉTODOS

- **GET** - Obtener información del servidor(parámetros en URL)
- **POST** - Enviar datos(parámetros en el cuerpo de la petición)
- **DELETE** - Eliminar un recurso
- **PUT** - Actualizar un recurso

HTTP

Versión 2

CLIENTE - SERVIDOR



ENCABEZADOS BINARIOS, GZIP/HPACK y TLS


```
> GET / HTTP/1.1  
> Host: www.aluracursos.com  
> User-Agent: curl/7.64.1  
> Accept: */*
```

```
HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Language: es  
Accept-Ranges: bytes  
Content-Length: 87363
```

```
<!DOCTYPE html>  
<html class="client-nojs" lang="es" dir="ltr">  
<head>  
<meta charset="UTF-8"/>
```

Petición

```
> GET / HTTP/1.1  
> Host: www.aluracursos.com  
> User-Agent: curl/7.64.1  
> Accept: */*
```

```
HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Language: es  
Accept-Ranges: bytes  
Content-Length: 87363
```

```
<!DOCTYPE html>  
<html class="client-nojs" lang="es" dir="ltr">  
<head>  
<meta charset="UTF-8"/>
```

Petición

```
> GET / HTTP/1.1  
> Host: www.aluracursos.com  
> User-Agent: curl/7.64.1  
> Accept: */*
```

Respuesta

```
HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Language: es  
Accept-Ranges: bytes  
Content-Length: 87363
```

```
<!DOCTYPE html>  
<html class="client-nojs" lang="es" dir="ltr">  
<head>  
<meta charset="UTF-8"/>
```

Método y URL

```
> GET / HTTP/1.1  
> Host: www.aluracursos.com  
> User-Agent: curl/7.64.1  
> Accept: */*
```

Status Code

```
HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Language: es  
Accept-Ranges: bytes  
Content-Length: 87363
```

Respuesta

```
<!DOCTYPE html>  
<html class="client-nojs" lang="es" dir="ltr">  
<head>  
<meta charset="UTF-8"/>
```

TEXTO

```
> GET / HTTP/1.1  
> Host: www.aluracursos.com  
> User-Agent: curl/7.64.1  
> Accept: */*
```

```
HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Language: es  
Accept-Ranges: bytes  
Content-Length: 87363
```

```
<!DOCTYPE html>  
<html class="client-nojs" lang="es" dir="ltr">  
<head>  
<meta charset="UTF-8"/>
```

```
> GET / HTTP/1.1
> Host: www.aluracursos.com
> User-Agent: curl/7.64.1
> Accept: */*
```

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Content-Language: es
Accept-Ranges: bytes
Content-Length: 87363
```

```
H4sIAAAAAAAAA/12Q3U4CMRBGX2XsdRfj/S
4JQwxoiS+FZaGaOUDS723/jqe4EWCUAS
+Y8oeUEW23a/mwABwU+v7arYNZOIcarl9
13Gasd/2ewJas0cdsSAiGEROUU7813+OWn
```

GZIP

10010110 01011011 10110011 10010111

Binario

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Content-Language: es
Accept-Ranges: bytes
Content-Length: 87363

H4slAAAAAAAAA/12Q3U4CMRBGX2XsdRfj/S
4JQwxoiS+FZaGaOUDS723/jqe4EWCUAS
+Y8oeUEW23a/mwABwU+v7arYNZOIcarl9
13Gasd/2ewJas0cdsSAiGEROUU7813+OWn

GZIP

10010110 01011011 10110011 10010111

Binario

00101101 01101001 01001101
00011101 01001101 11000110
11000110 11010010 10001101

Binario

H4slAAAAAAAAA/12Q3U4CMRBGX2XsdRfj/S
4JQwxoiS+FZaGaOUDS723/jqe4EWCUAS
+Y8oeUEW23a/mwABwU+v7arYNZOIcarl9
13Gasd/2ewJas0cdsSAiGEROUU7813+OWn

GZIP

%&".\$.#&%&/((&/)943

Binario + HPACK

/"\$.254"%#1&="!"/&##&%
&.\$!#/&%(/.\$)=&.*\$6422&
64/%\$#5#&/232#%\$&/)

Binario + HPACK

H4sIAAAAAAAAAA/12Q3U4CMRBGX2XsdRfj/S
4JQwxoiS+FZaGaOUDS723/jqe4EWCUAS
+Y8oeUEW23a/mwABwU+v7arYNZOIcarI9
13Gasd/2ewJas0cdsSAiGEROUU7813+OWn

GZIP

????????????????????

Binario > HPACK >
TLS

????????????????????
????????????????????
????????????????????

Binario > HPACK >
TLS

????????????????????
????????????????????
????????????????????
????????????????????

GZIP > TLS

ENCABEZADOS STATEFUL

GET /

HOST: www.aluracursos.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_3) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/89.0.4389.128 Safari/537.36

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.9

Accept-Language: es-MX

Accept-Encoding: gzip, deflate

GET /

HOST: www.aluracursos.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.128 Safari/537.36

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Accept-Language: es-MX

Accept-Encoding: gzip, deflate

GET /script.js

HOST: www.aluracursos.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.128 Safari/537.36

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Accept-Language: es-MX

Accept-Encoding: gzip, deflate

GET /

HOST: www.aluracursos.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.128 Safari/537.36

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.9

Accept-Language: es-MX

Accept-Encoding: gzip, deflate

GET /script.js

GET /logo.png

HOST: imagenes.aluracursos.com

Accept: image/png

Servidor PUSH

CLIENTE

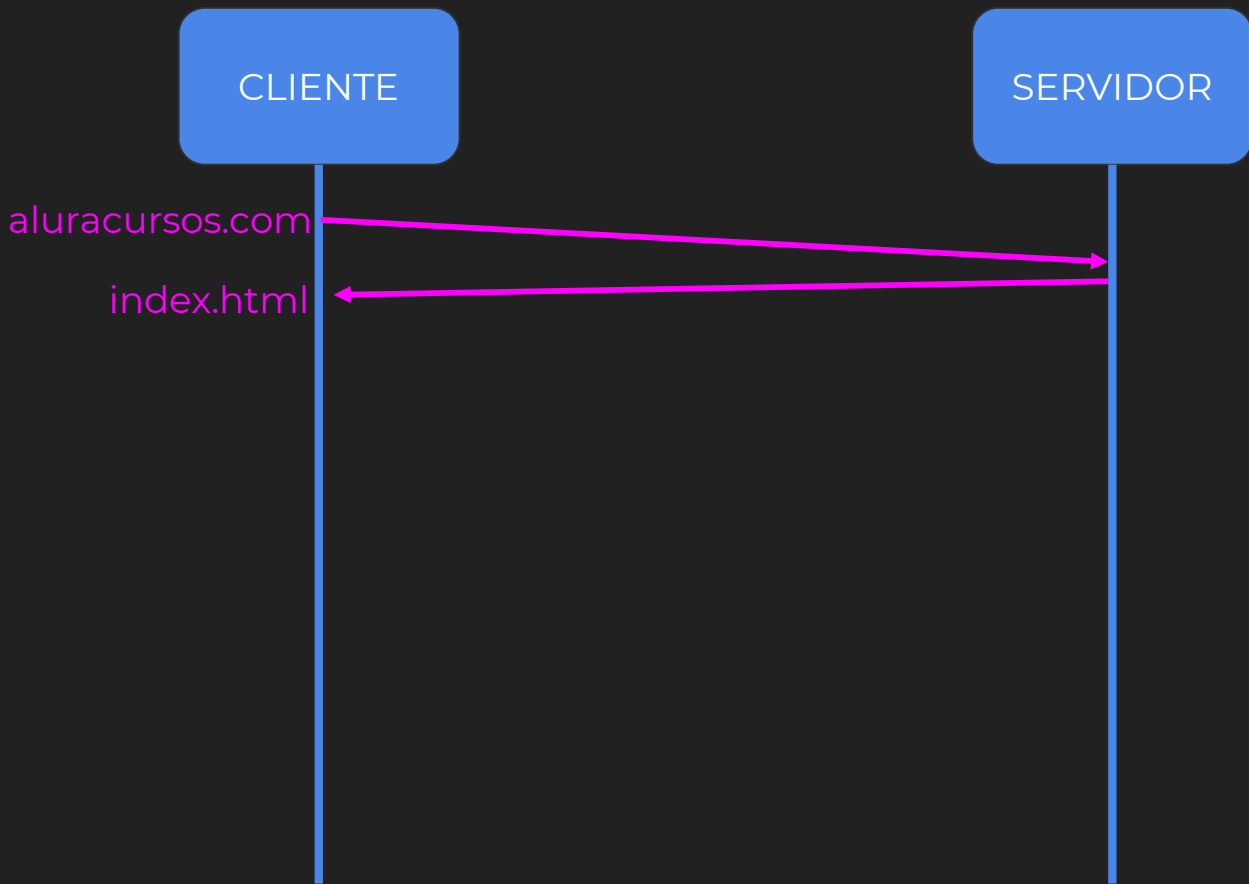
SERVIDOR

CLIENTE

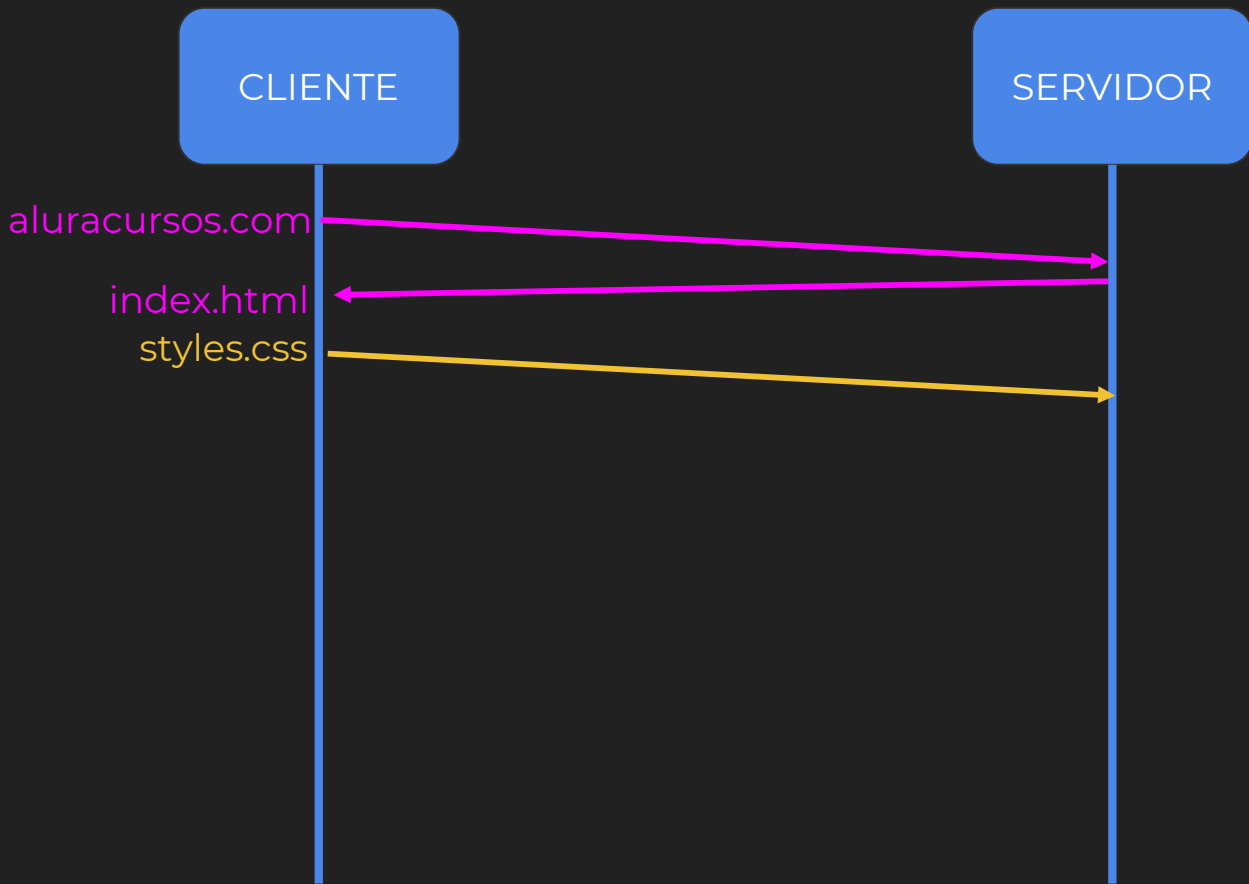
SERVIDOR

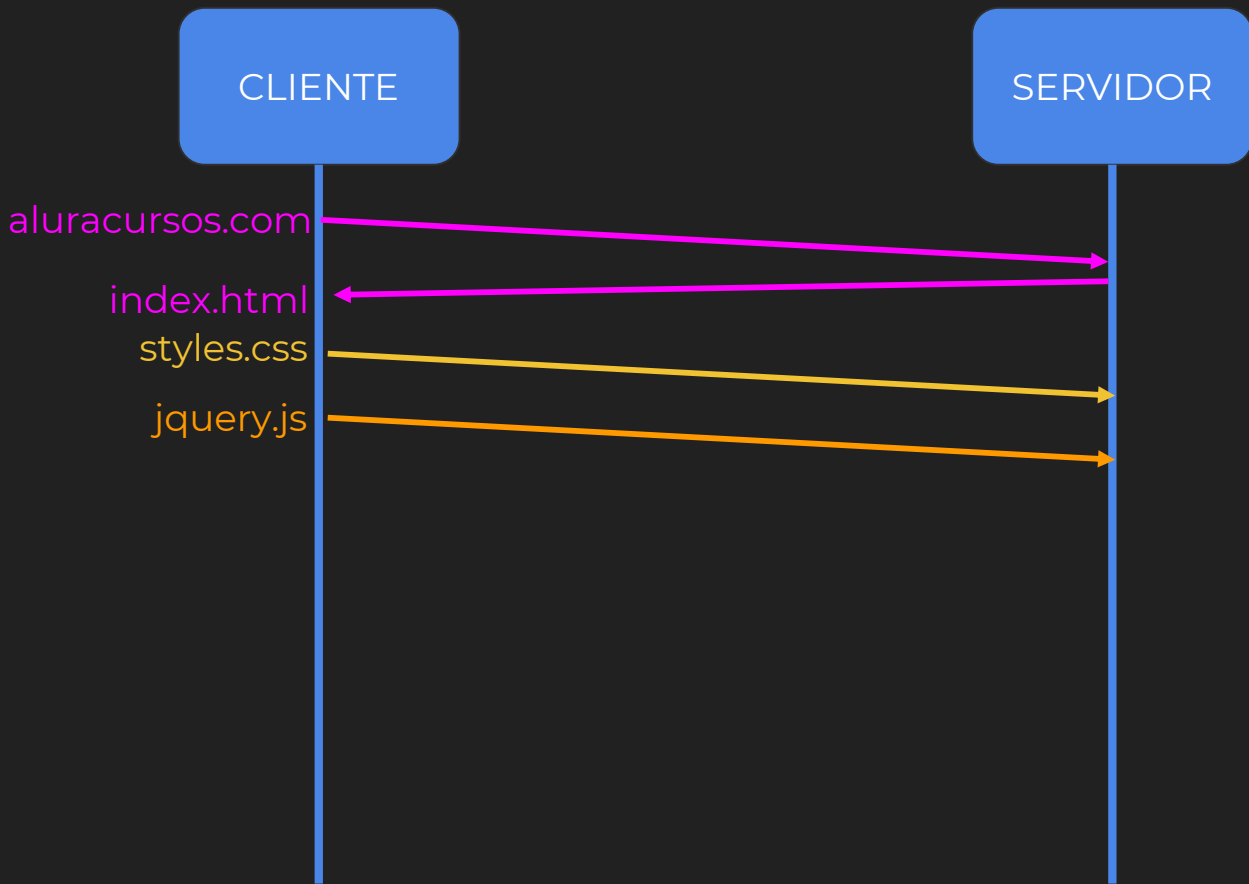
aluracursos.com

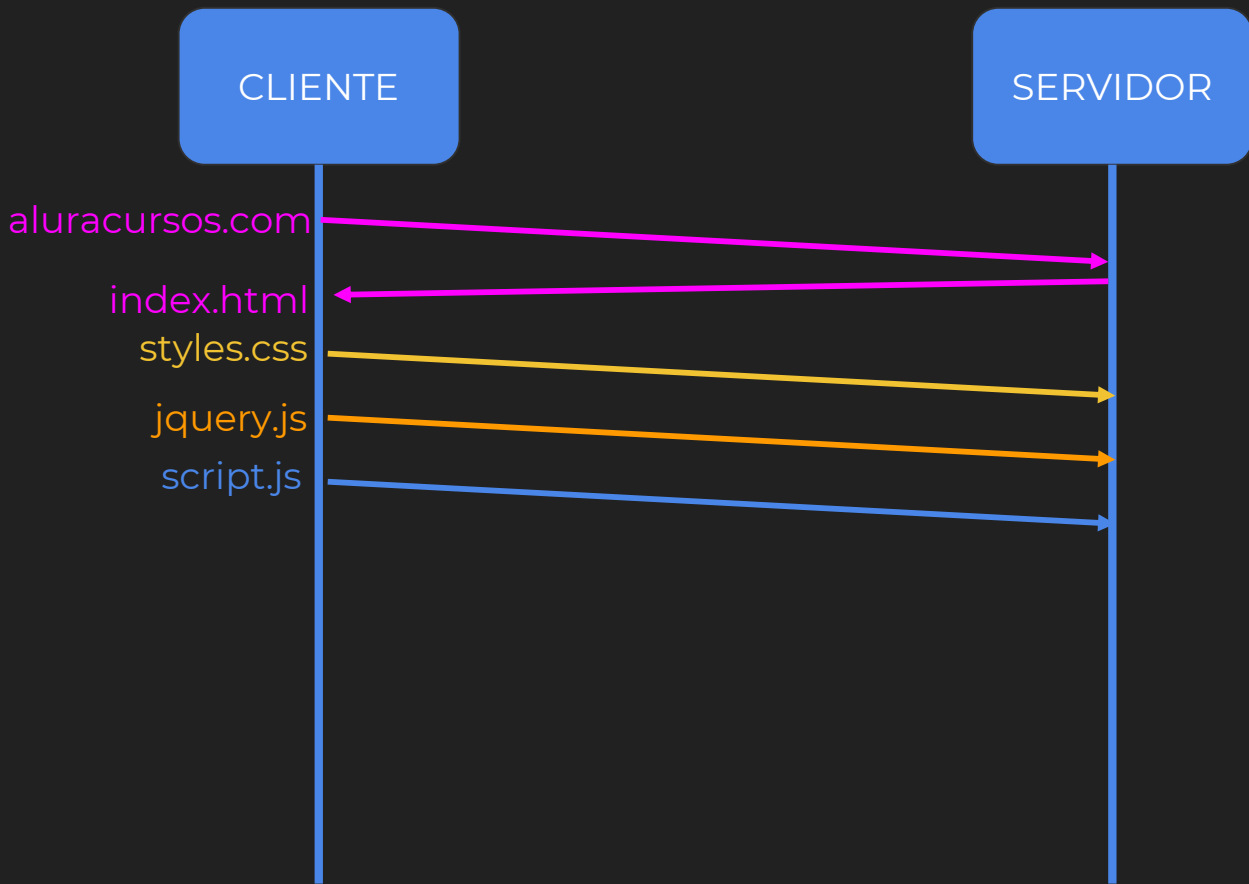


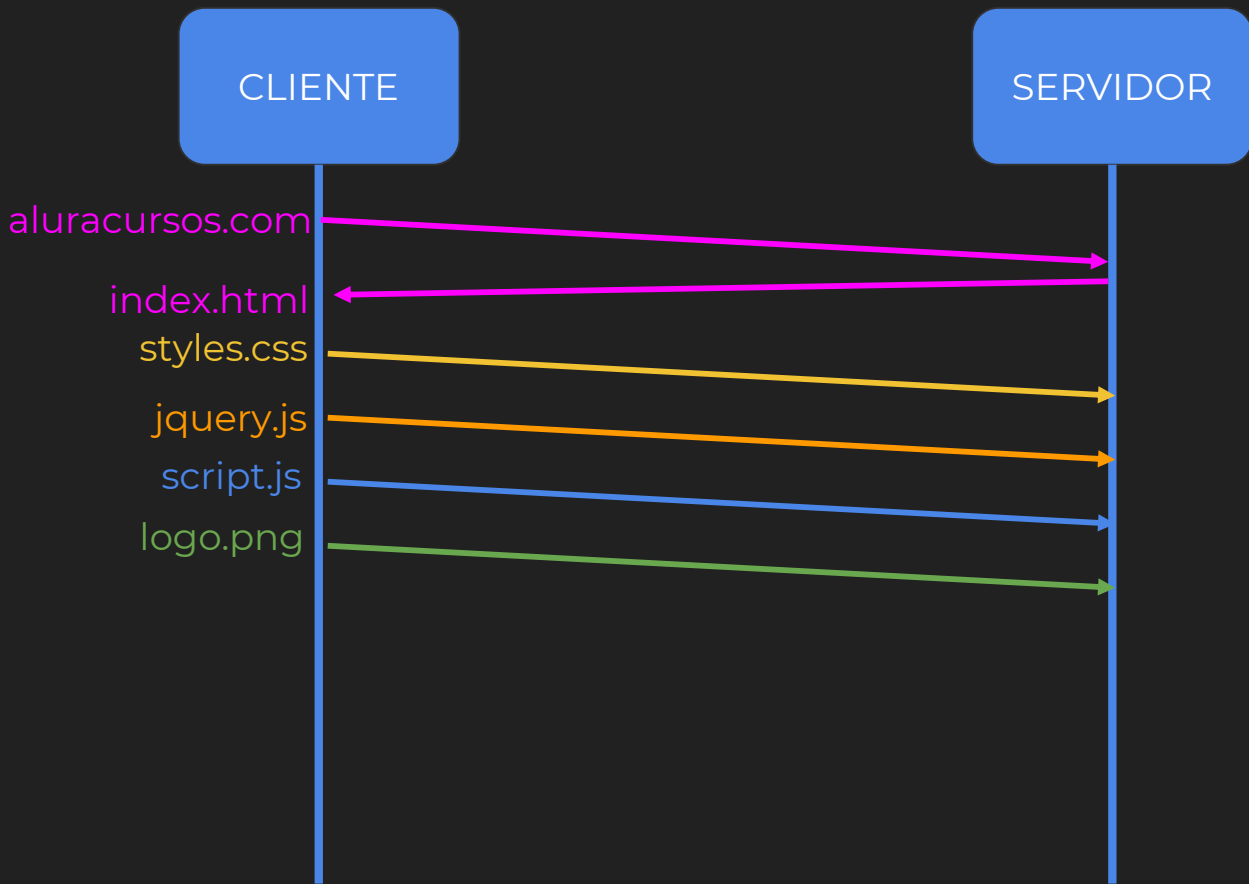


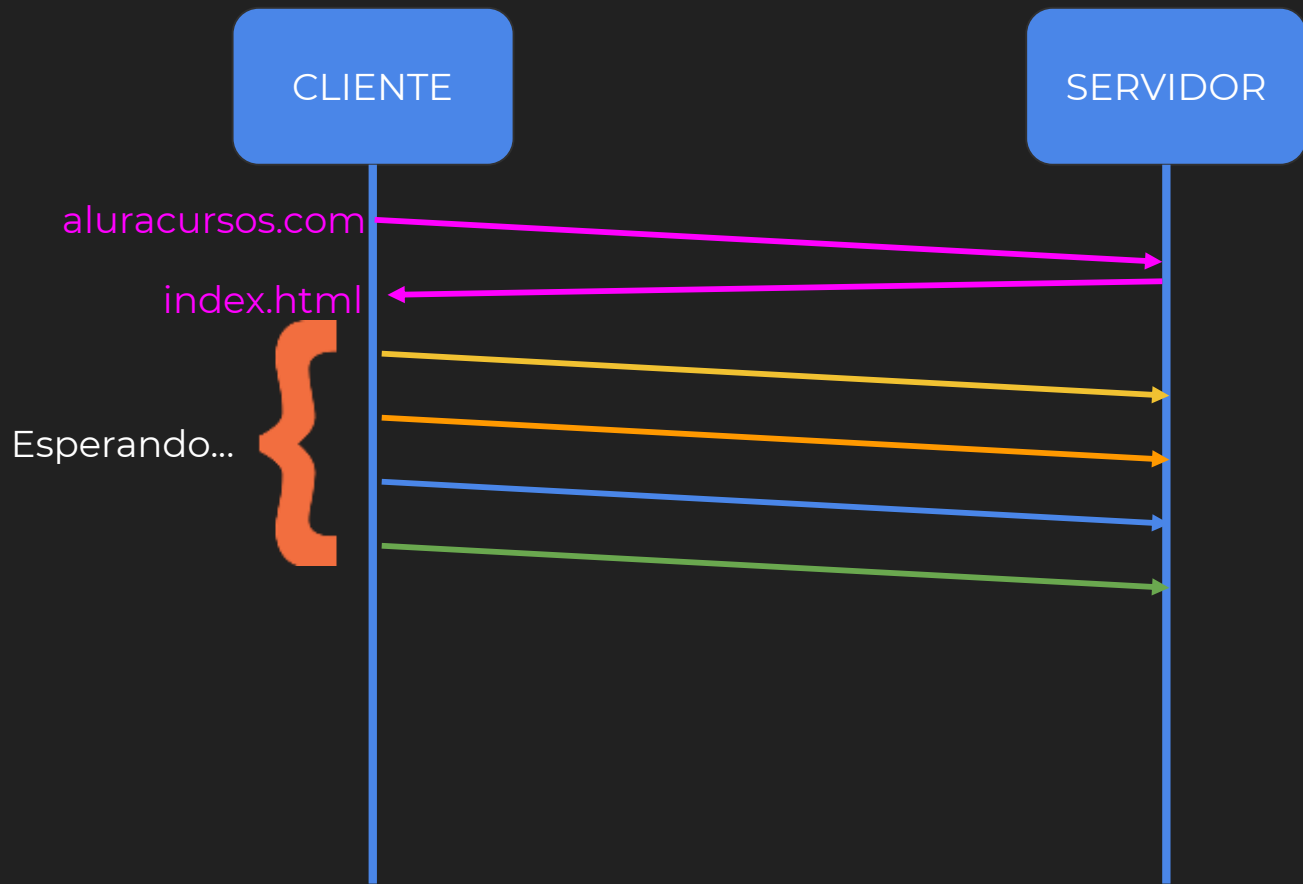
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Alura Cursos</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
  <script type="text/javascript" src="jquery.js"></script>
  <script type="text/javascript" src="script.js"></script>
</head>
<body>
  ...
  
</body>
</html>
```

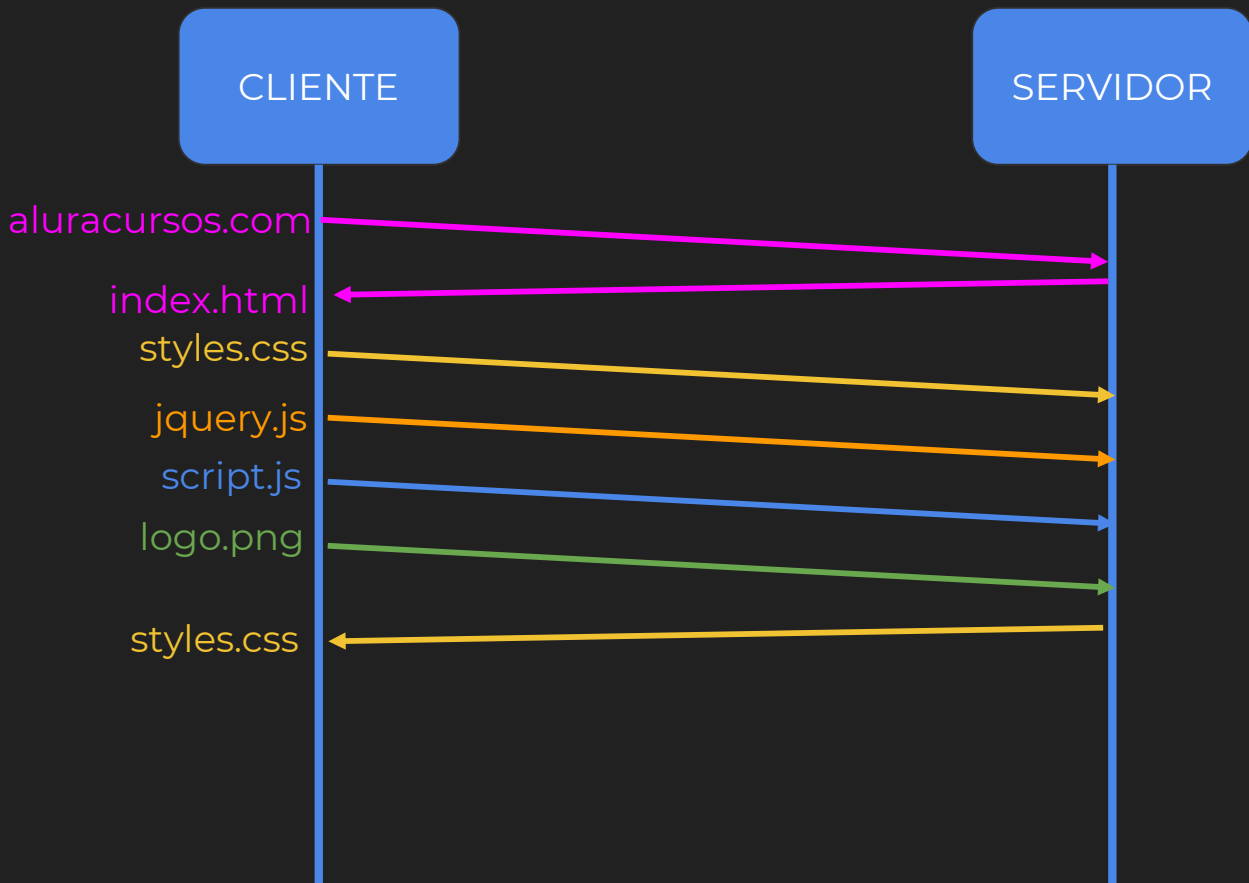


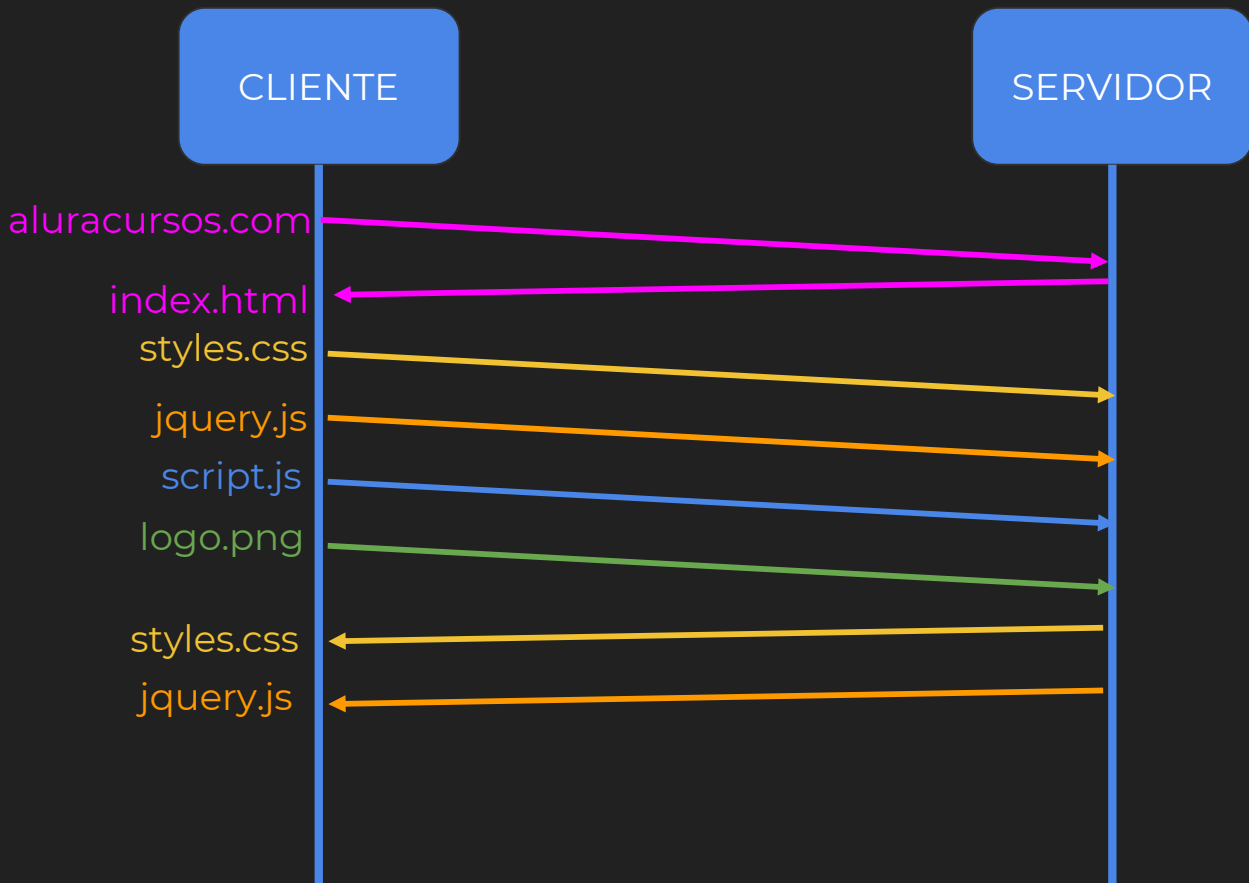


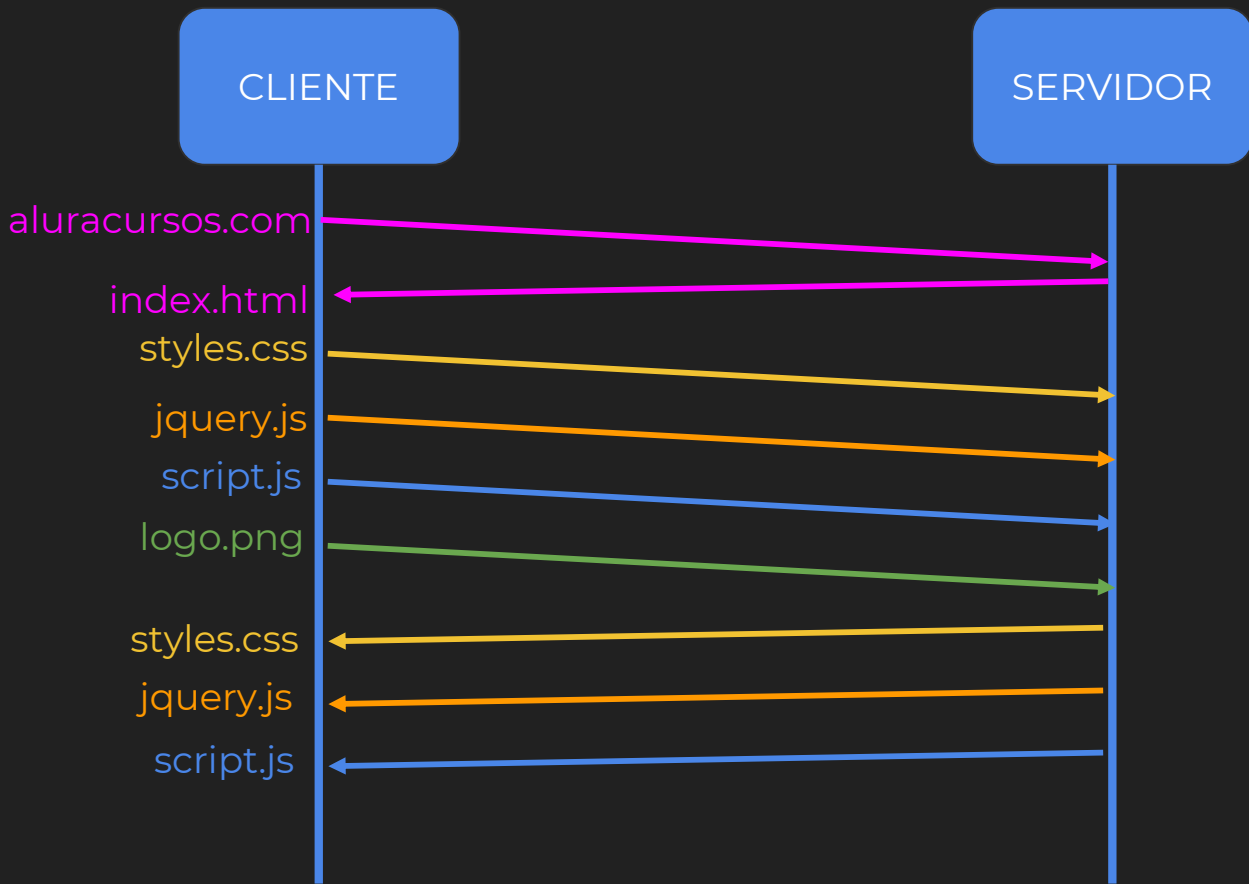


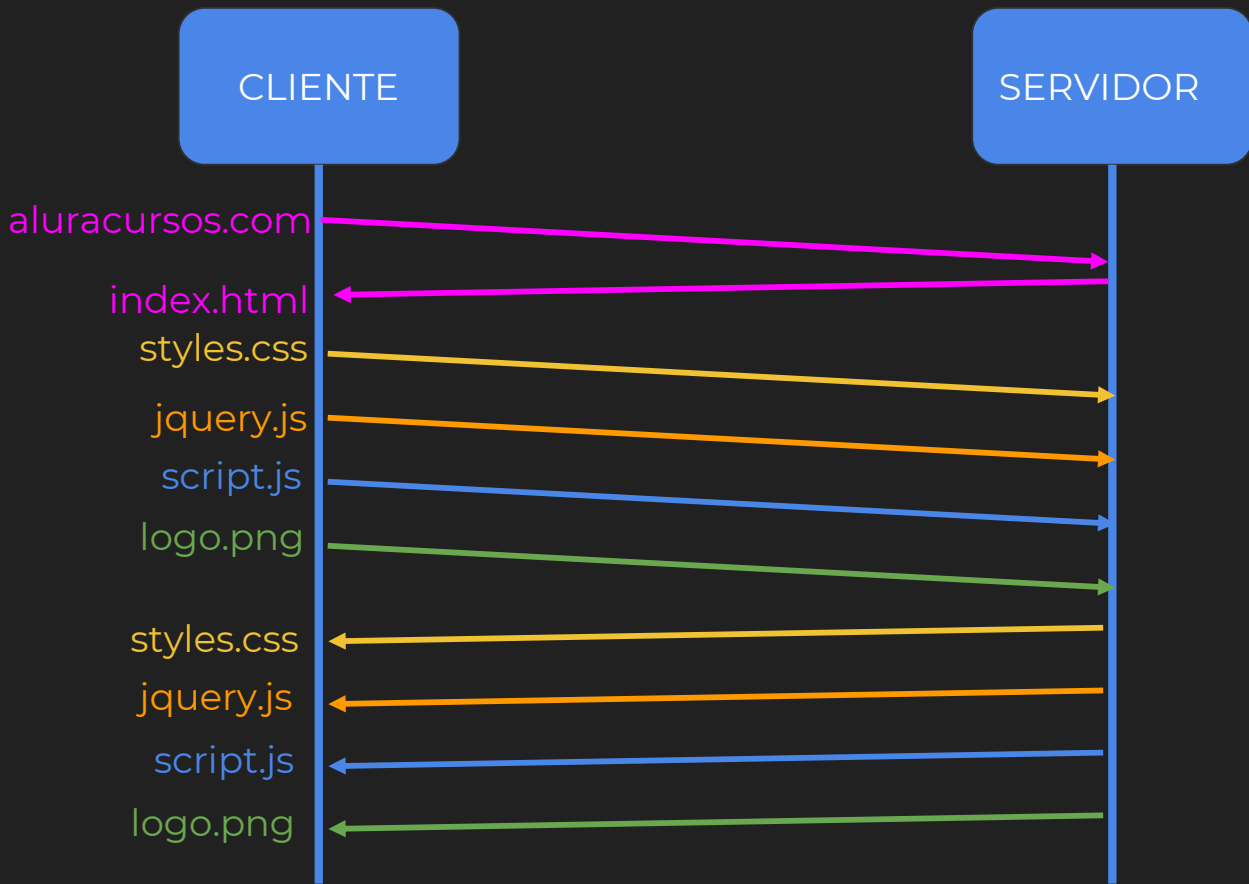








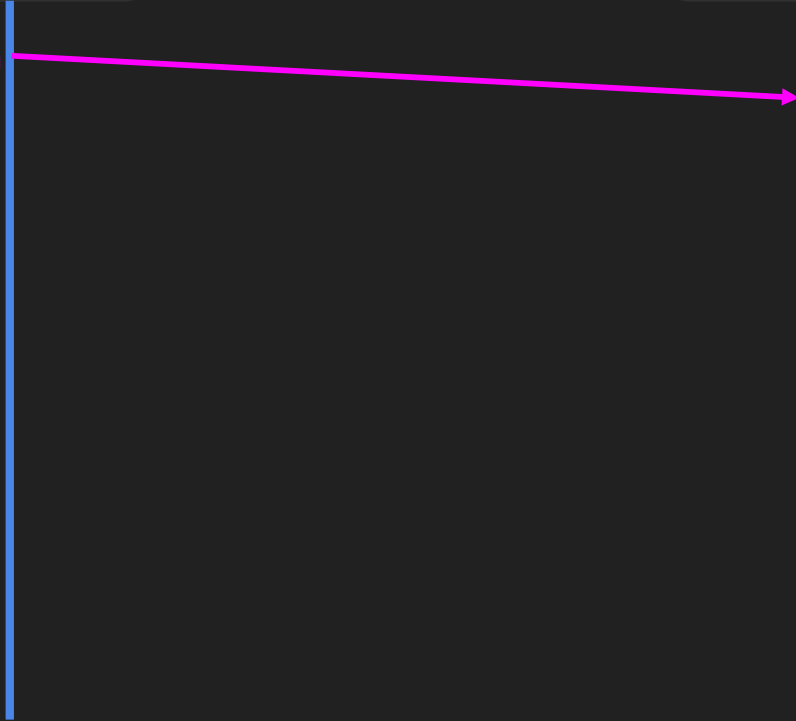


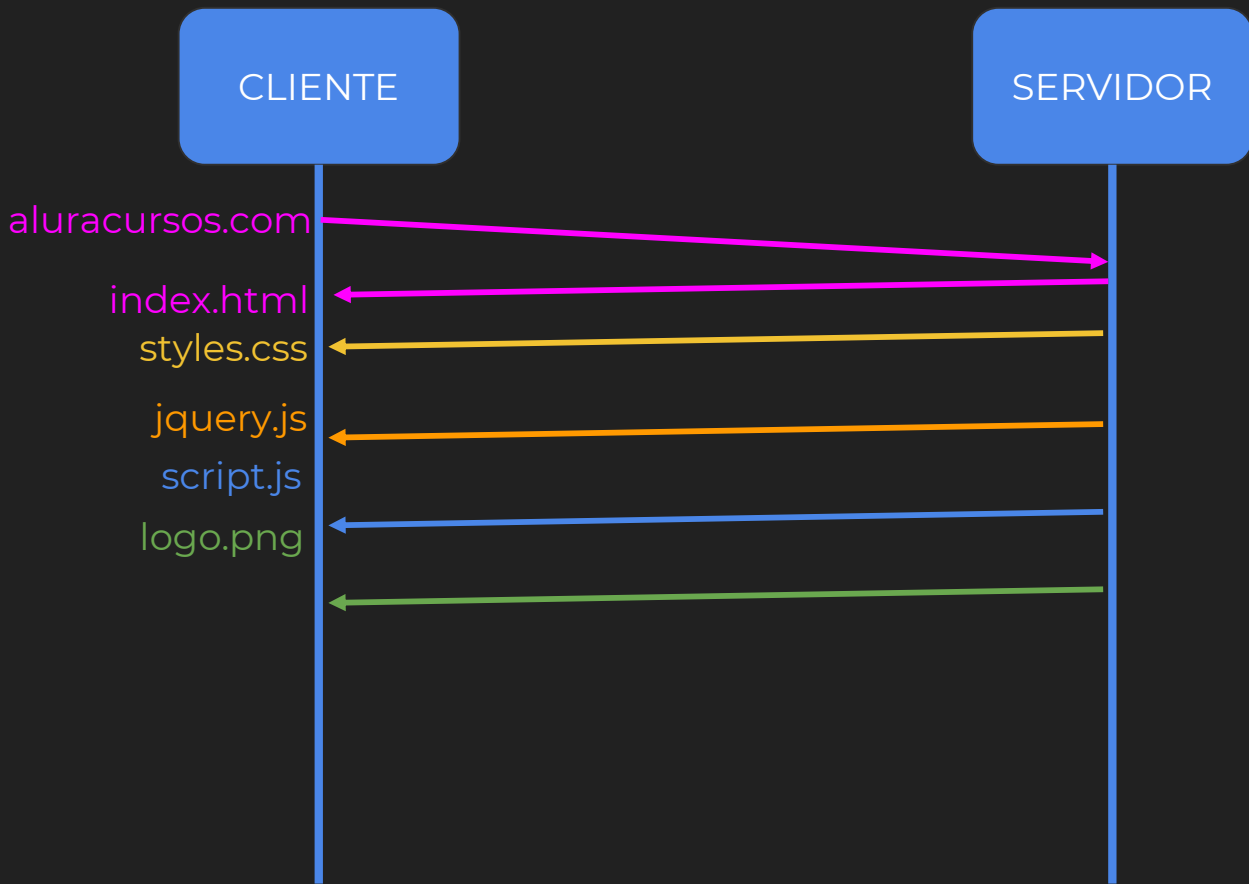


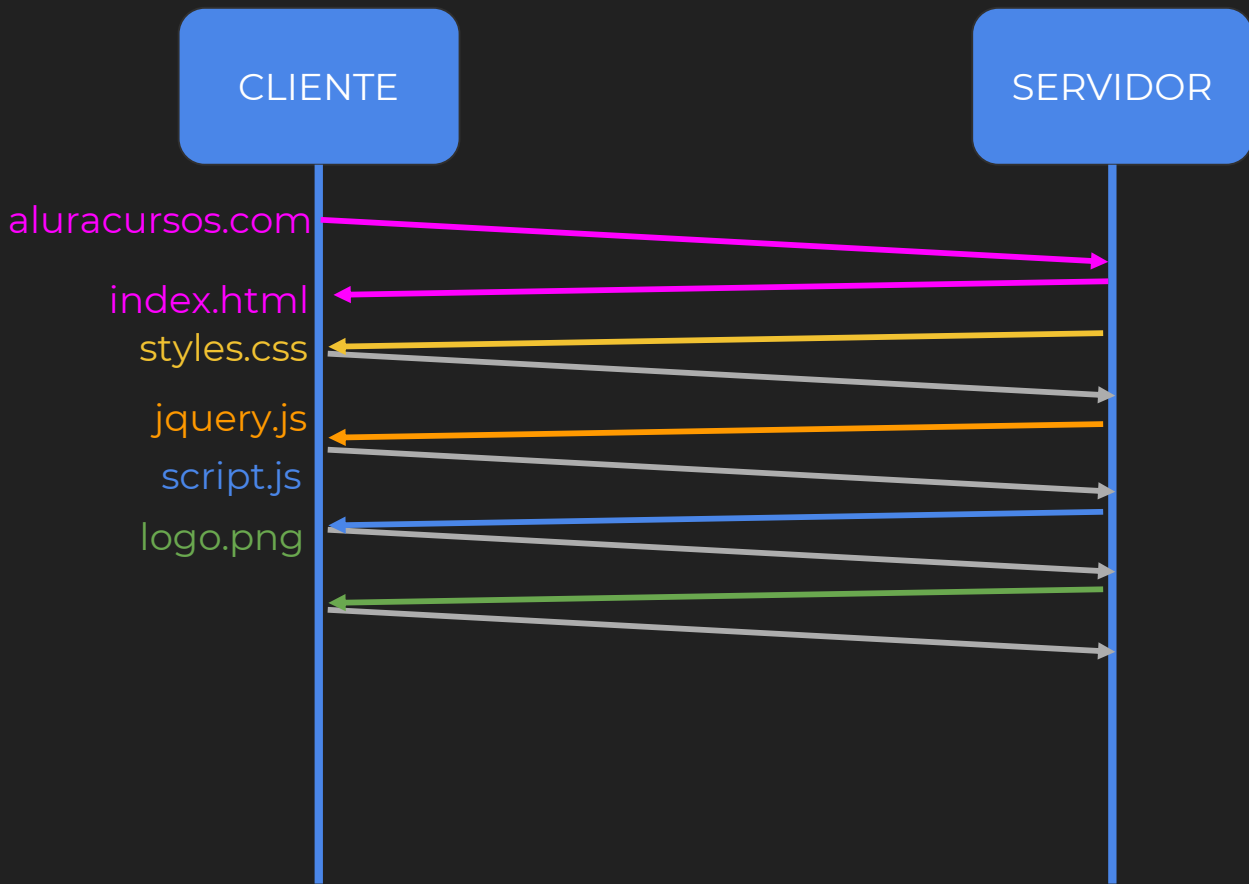
CLIENTE

SERVIDOR

aluracursos.com





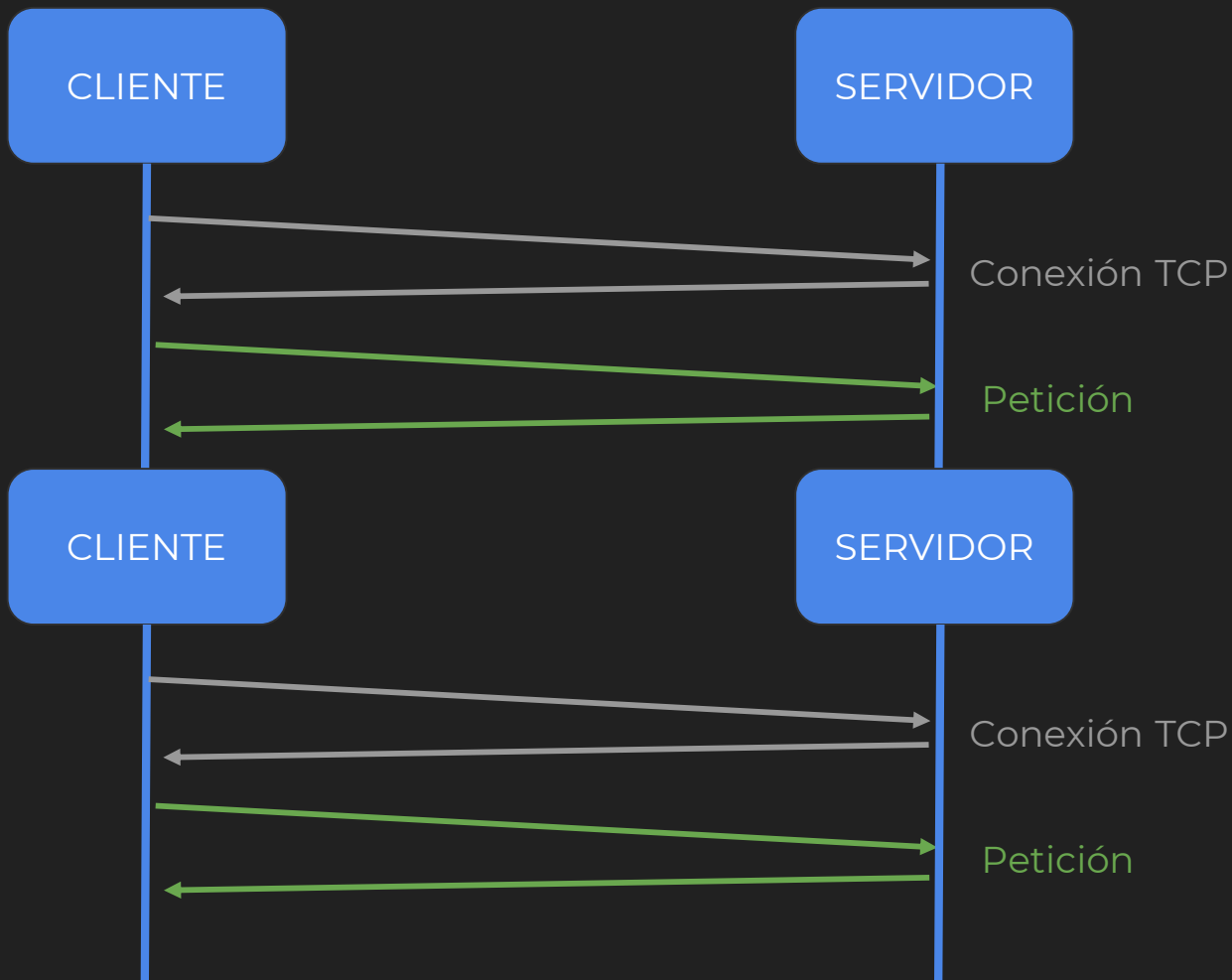


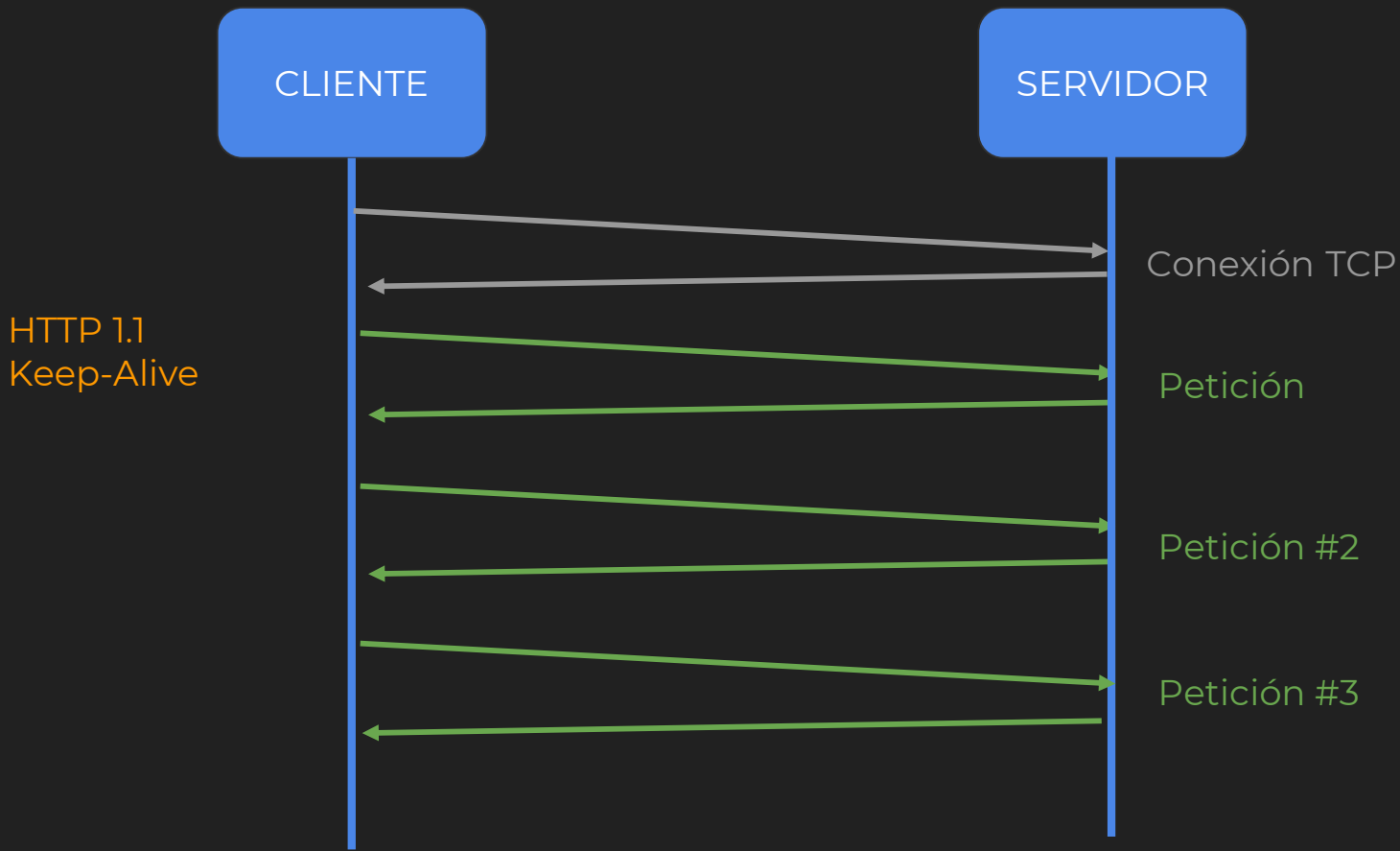
Multiplexación

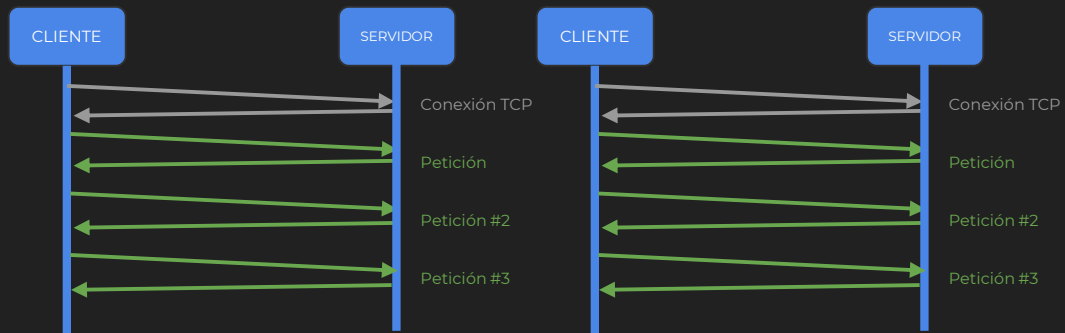
HTTP 1.1



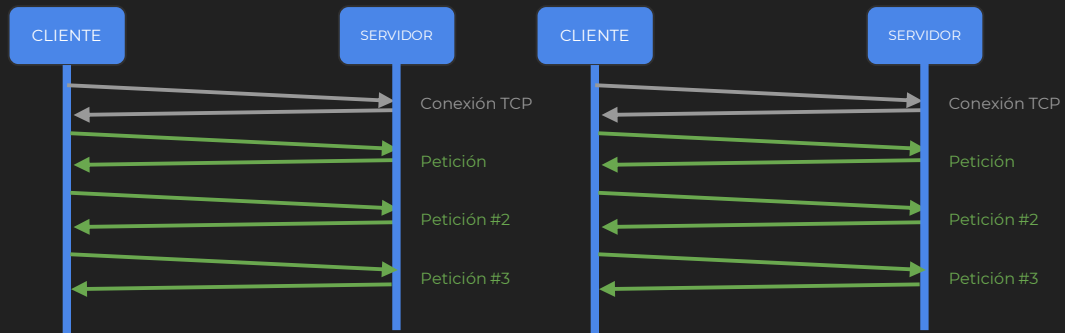
HTTP 1.1

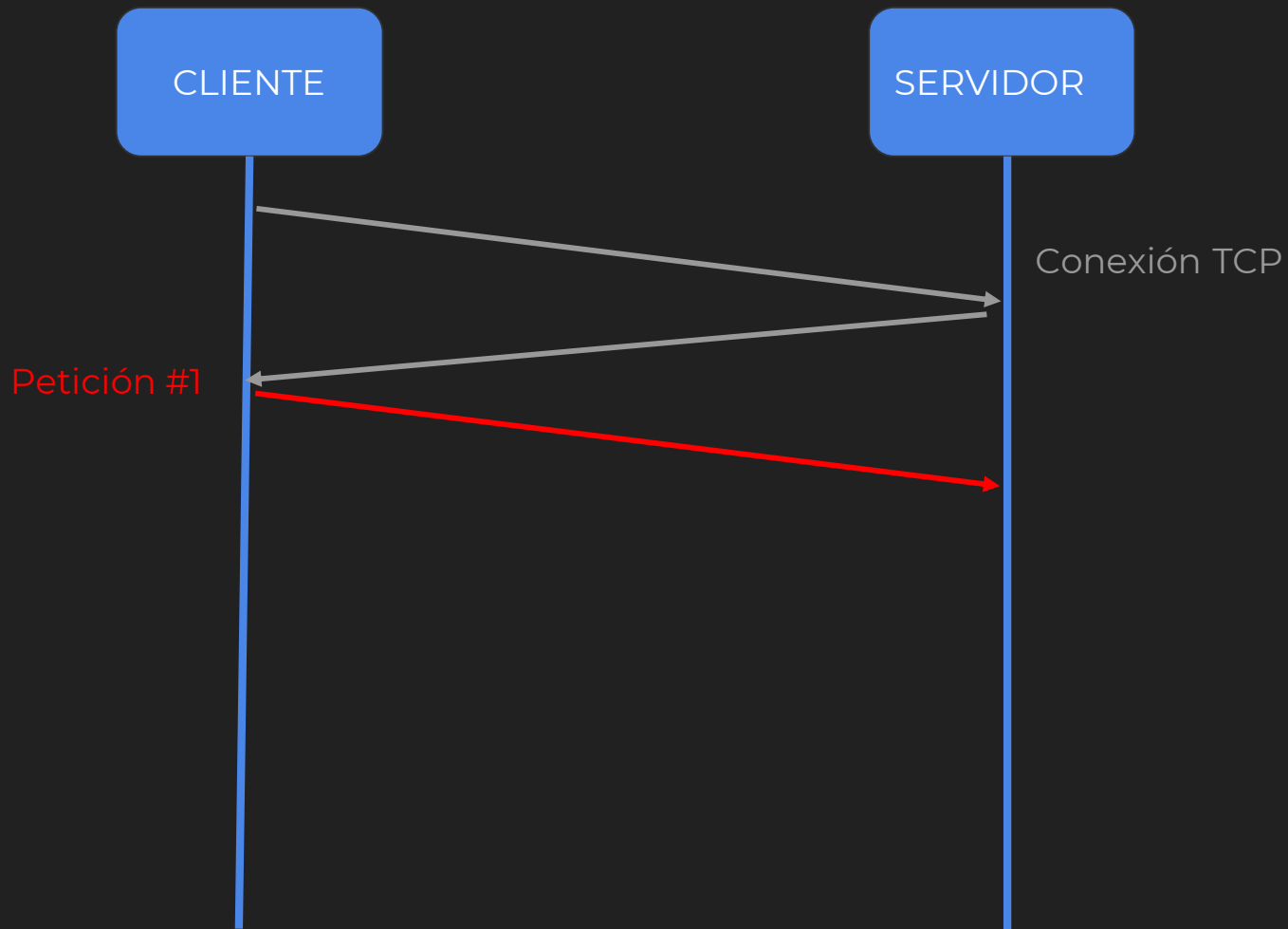


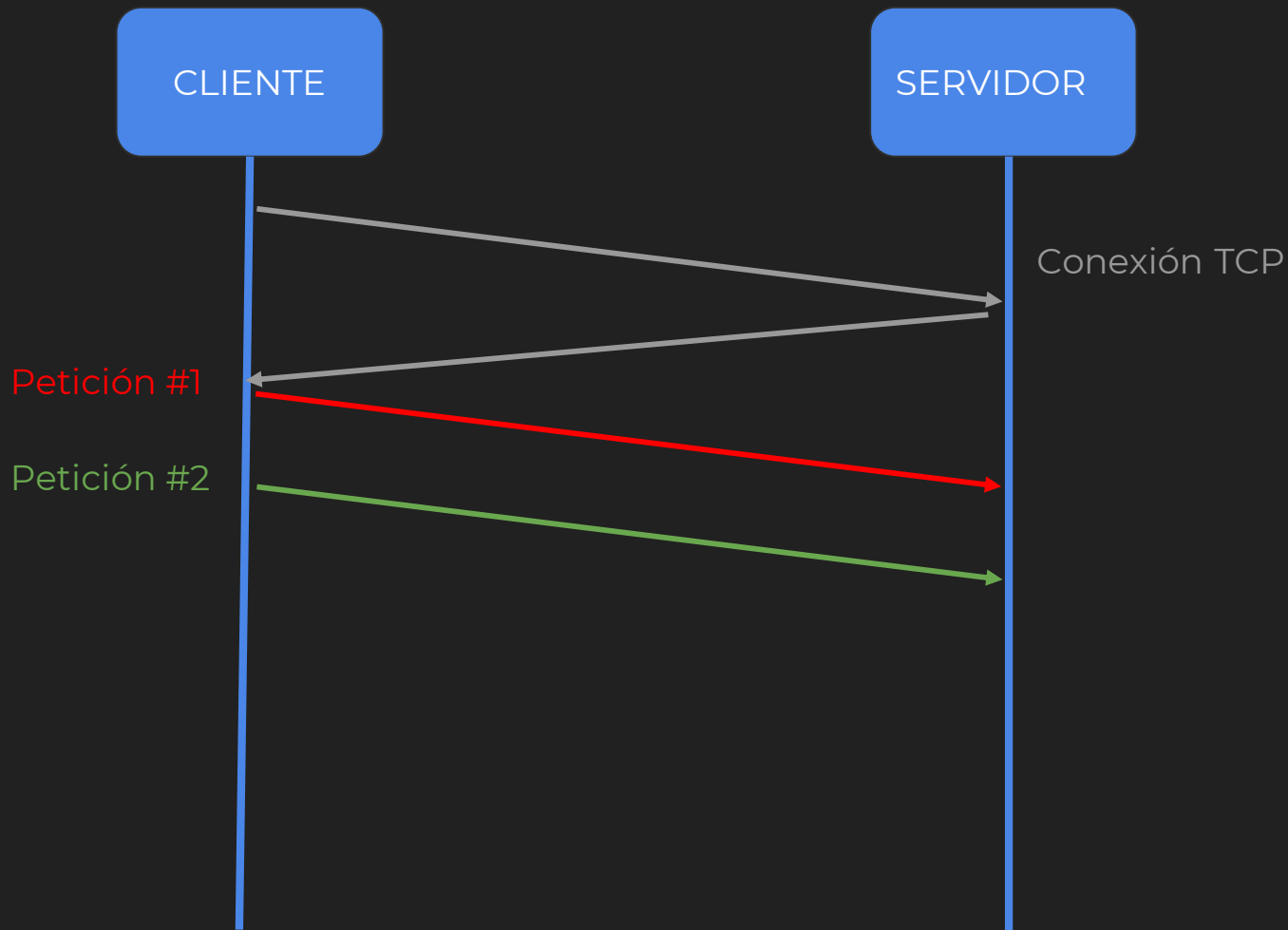


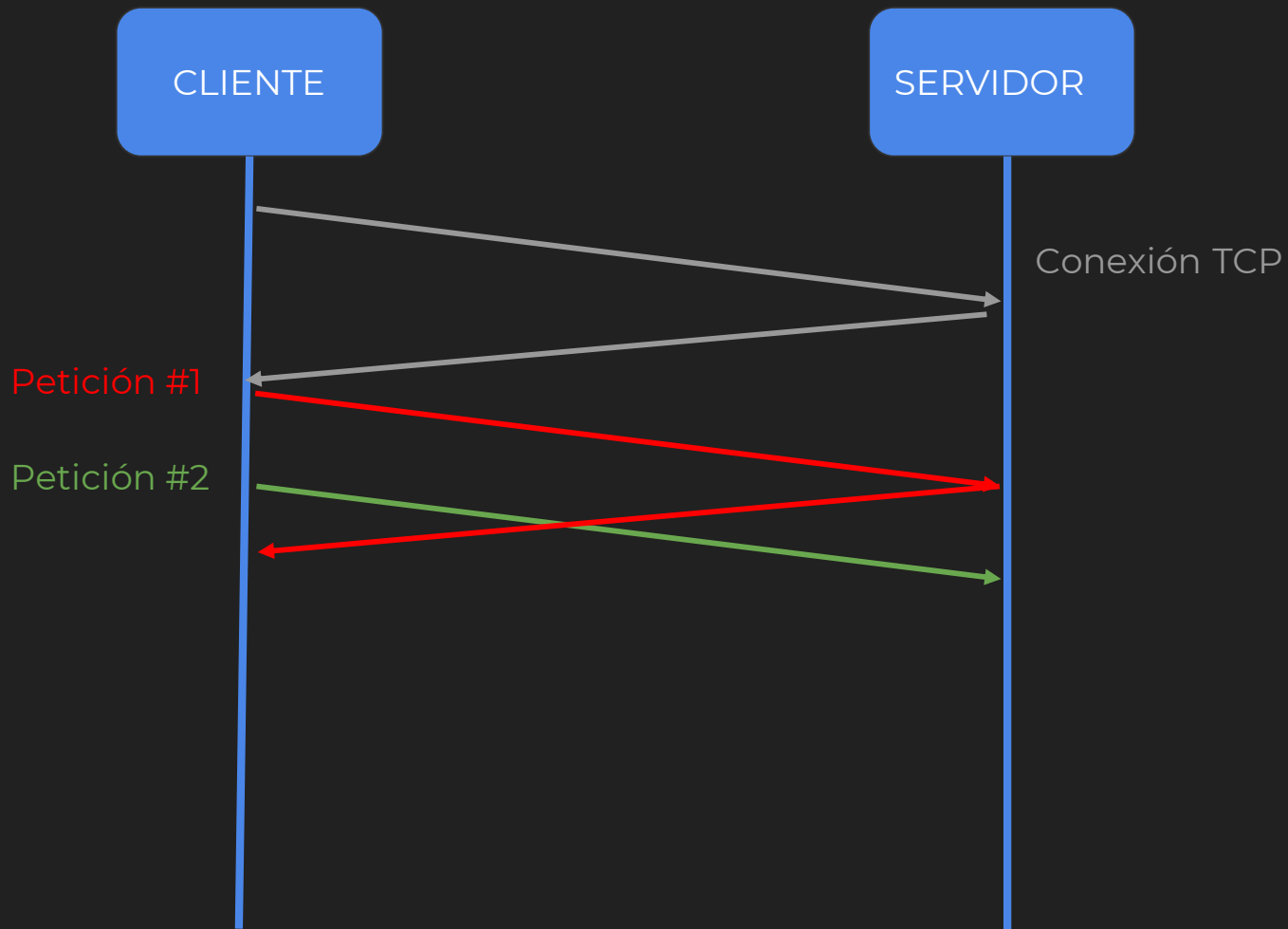


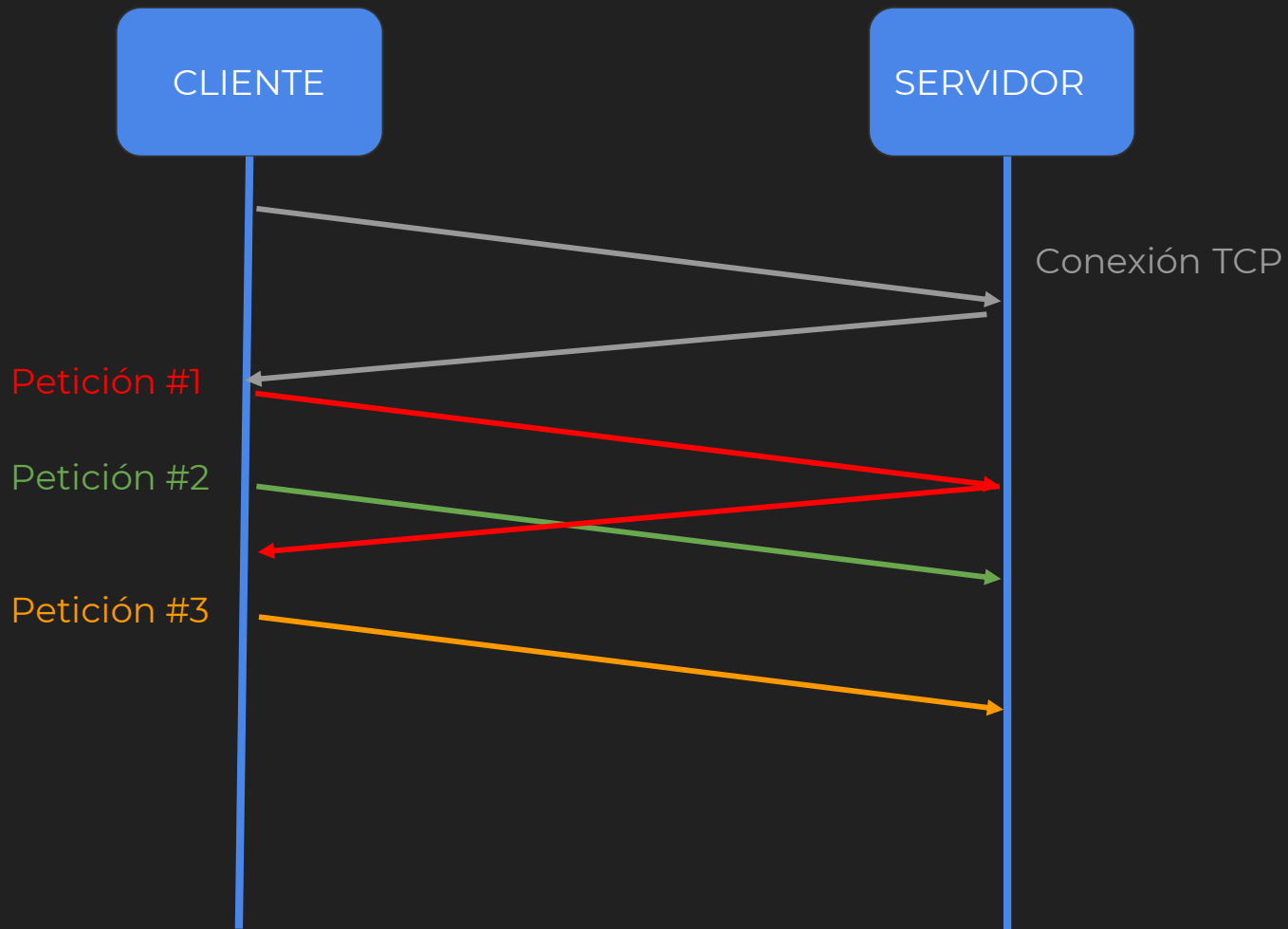
De 4 a 8 conexiones

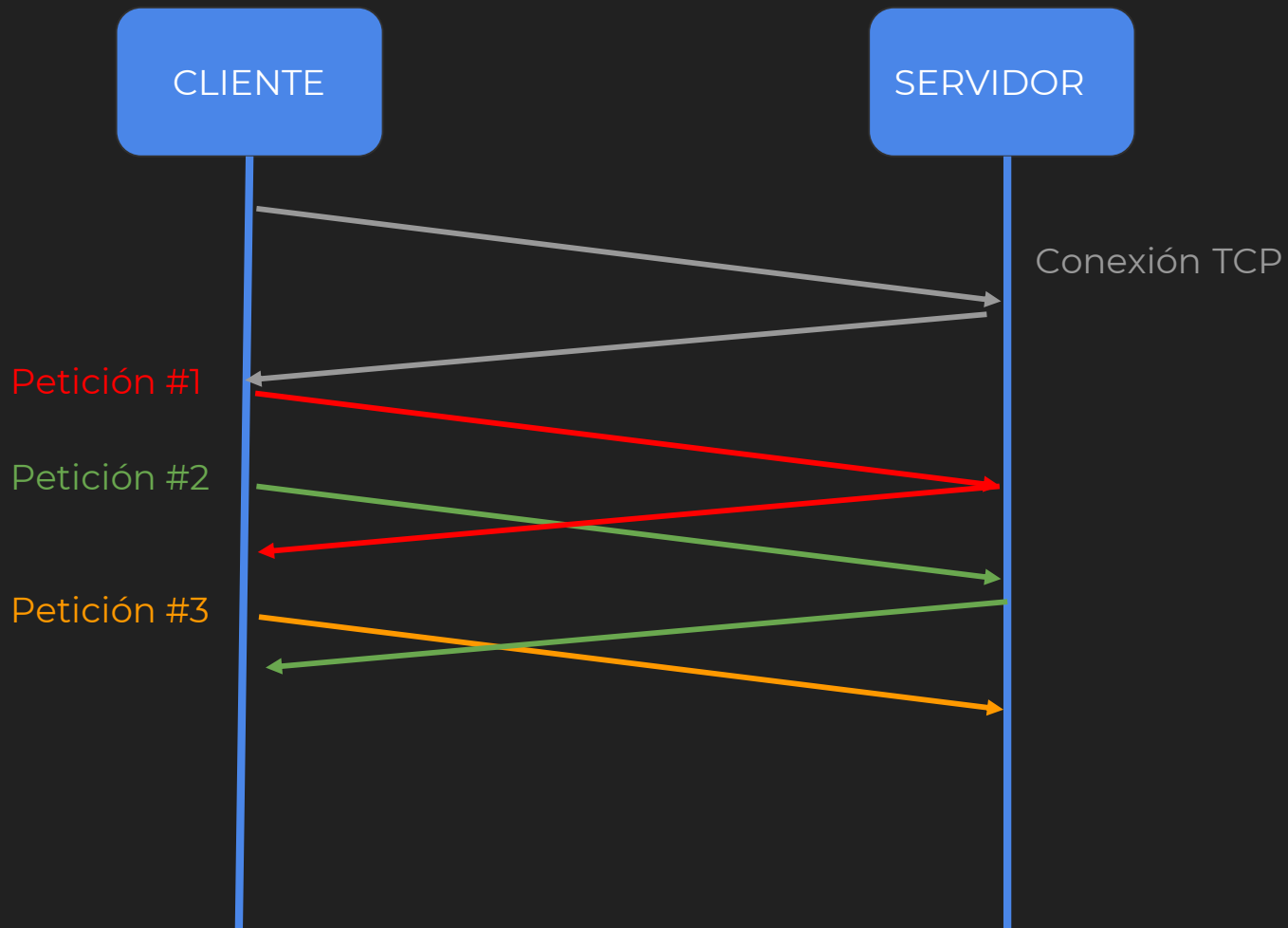


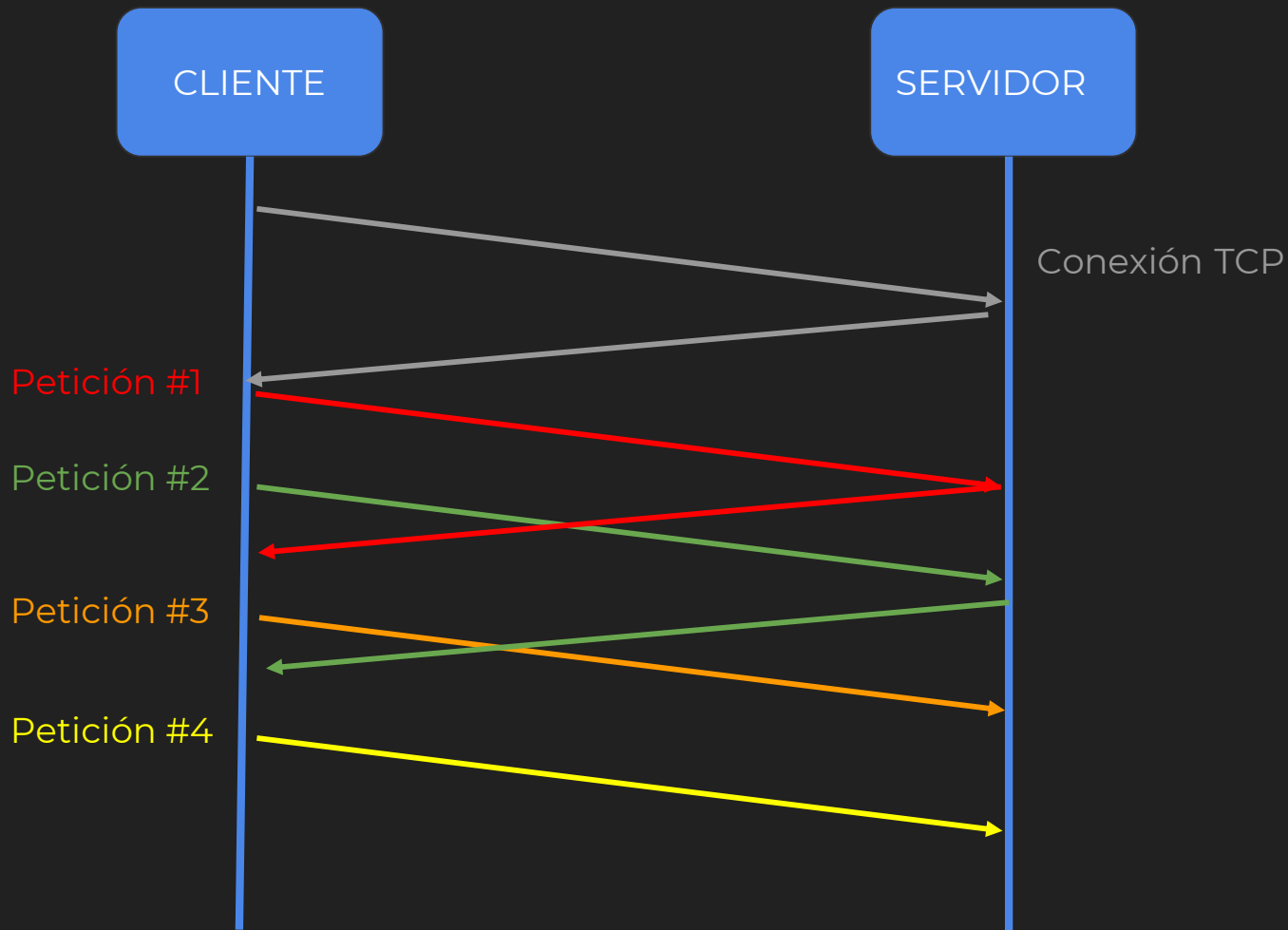


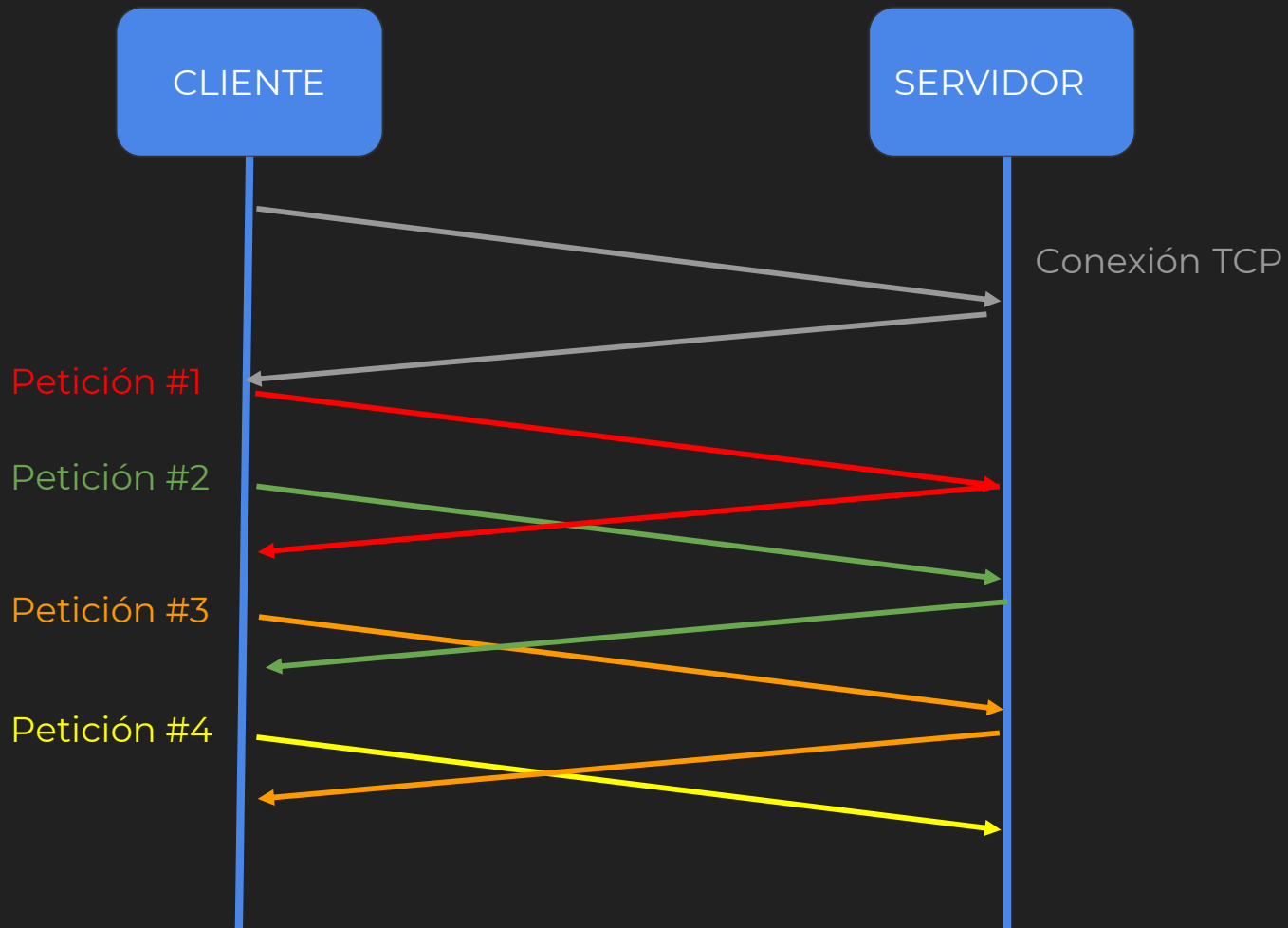


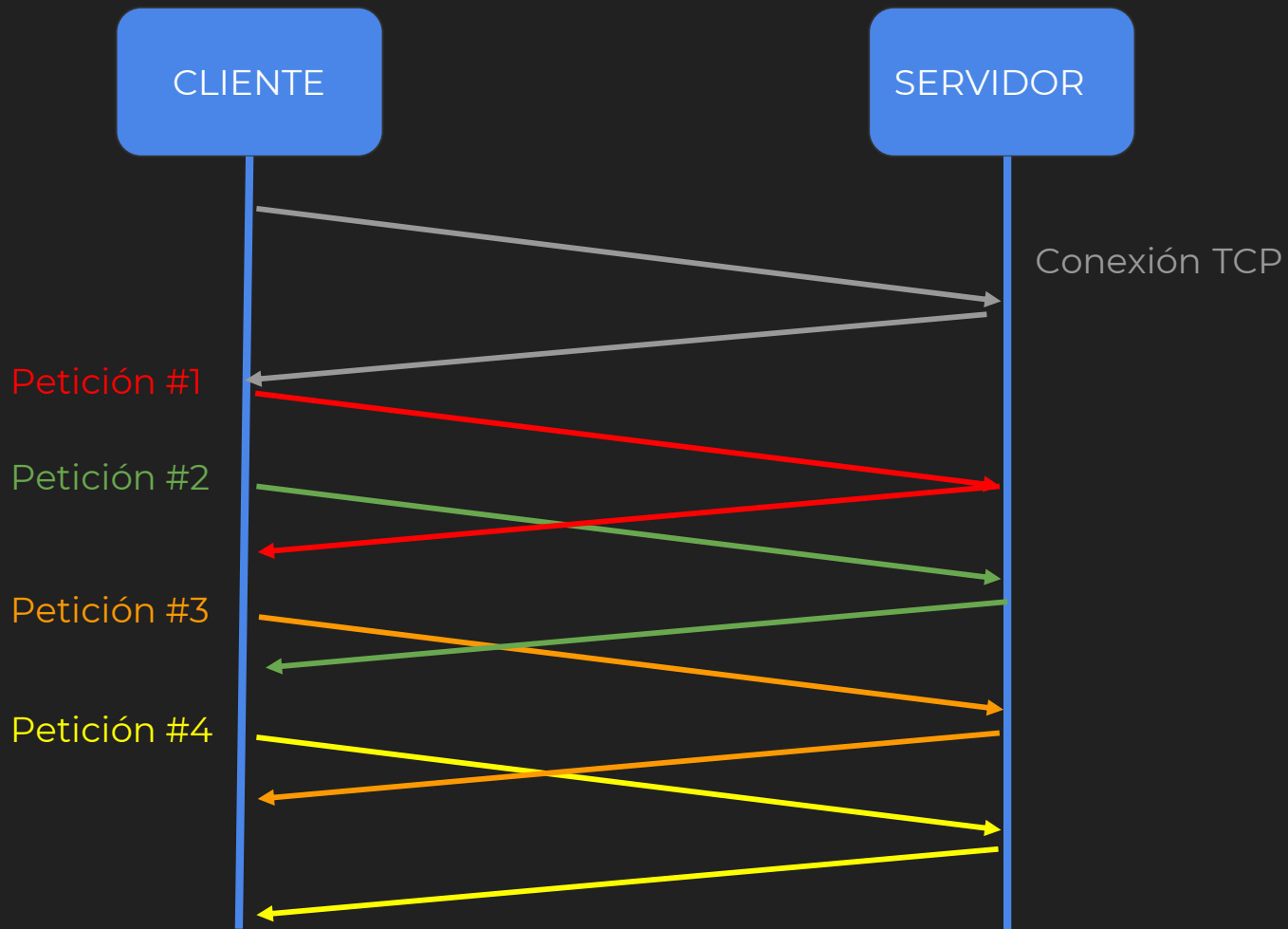












HTTP/2

- Se basa en lo que ya establece HTTP

HTTP/2

- Se basa en lo que ya establece HTTP
- Encabezados binarios y compresión(HPACK)

HTTP/2

- Se basa en lo que ya establece HTTP
- Encabezados binarios y compresión(HPACK)
- GZIP por defecto en las respuestas

HTTP/2

- Se basa en lo que ya establece HTTP
- Encabezados binarios y compresión(HPACK)
- GZIP por defecto en las respuestas
- Multiplexación(Petición y respuesta son paralelas)

HTTP/2

- Se basa en lo que ya establece HTTP
- Encabezados binarios y compresión(HPACK)
- GZIP por defecto en las respuestas
- Multiplexación(Petición y respuesta son paralelas)
- Encabezados con estado(solo se mandan los encabezados que cambian)

HTTP/2

- Se basa en lo que ya establece HTTP
- Encabezados binarios y compresión(HPACK)
- GZIP por defecto en las respuestas
- Multiplexación(Petición y respuesta son paralelas)
- Encabezados con estado(solo se mandan los encabezados que cambian)
- Servidor PUSH

URI

URL

<https://www.aluracursos.com/introduccion-html-css>

URI

<urn:www.aluracursos:introduccion-html-css>