
SISTEMAS WEB

CURSO 2022/2023

HTTP - HyperText Transfer Protocol
Cache



Web Sistemak by [Oskar Casquero](#) & [María Luz Álvarez](#) is licensed under a [Creative Commons Reconocimiento 4.0 Internacional License](#).

FUNCIONAMIENTO DE HTTP:

CACHÉ

- ¿Cómo se evita el intercambio innecesario de tráfico?
 - El protocolo http puede evitar la transferencia de recursos no modificados desde que el cliente los solicitó por última vez:
 - Status: **304**
 - Cabeceras:
 - En la solicitud: **Cache-Control, If-Modified-Since, If-None-Match**
 - En la respuesta: **Cache-Control, Date, Last-Modified, Etag**
 - De esta forma se optimiza el uso del ancho de banda y se evita que el servidor web procese y genere demandas y respuestas redundantes.
 - En relación al caché, tres son los aspectos a tener en cuenta:
 - Frescura
 - Validación en el cliente
 - Validación en servidor

FUNCIONAMIENTO DE HTTP:

CACHÉ

- Supongamos que un navegador solicita una página web. Una vez recibe la respuesta (documento HTML), el navegador observa que dentro de la página web hay un elemento que hace referencia a una imagen jpg.
- Así, el navegador realiza una segunda petición HTTP para obtener la imagen. El servidor web se la facilita y añade ciertos meta-datos a través de las cabeceras de la respuesta

Solicitud HTTP del ejemplo

```
GET /image.jpg HTTP/1.1
Host: sw2020.com:8080
Accept: image/*
User-Agent: Mozilla Windows Escritorio
```

Respuesta HTTP del ejemplo

```
HTTP/1.1 200 OK
Date: Thu, 20 Nov 2015 20:25:52 GMT
Last-Modified: Tue, 17 Sep 2015 13:00:02 GMT
ETag: "1a968-3ec-4e693e61bb8b6"
Cache-Control: max-age=2592000
Content-Length: 12405
Content-Type: image/jpg

IMAGEN (contenido binario)
```

FUNCIONAMIENTO DE HTTP:

CACHÉ - FRESCURA

- Supongamos que el usuario cierra el navegador y lo vuelve a abrir al día siguiente para solicitar la misma página web que contiene la misma imagen que no ha cambiado.

Respuesta HTTP del ejemplo anterior

```
HTTP/1.1 200 OK
Date: Thu, 20 Nov 2015 20:25:52 GMT
Last-Modified: Tue, 17 Sep 2015 13:00:02 GMT
ETag: "1a968-3ec-4e693e61bb8b6"
Cache-Control: max-age=2592000
Content-Length: 12405
Content-Type: image/jpg

CONTENIDO DE LA IMAGEN
```

La cabecera "Cache-Control" " [\(RFC 2616, Sección 14.9\)](#) se utiliza para gestionar diferentes aspectos de los caches de clientes y proxys.

En este caso, el servidor dice al navegador lo siguiente: "Durante un mes, no me preguntes por esta imagen".

Así, el navegador recoge la imagen directamente de su caché

FUNCIONAMIENTO DE HTTP:

CACHÉ - VALIDACIÓN EN CLIENTE

- Supongamos que dos meses después el usuario vuelve a solicitar la misma página web, pero, aunque el contenido de la web ha cambiado, la imagen, que se haya embebida en ella, no se ha modificado.
- El navegador, al ver que ha expirado el tiempo de almacenamiento en cache asociado a la imagen, realiza una nueva petición HTTP de la siguiente forma:

Respuesta HTTP del ejemplo anterior

```
HTTP/1.1 200 OK
Date: Thu, 20 Nov 2015 20:25:52 GMT
Last-Modified: Tue, 17 Sep 2015 13:00:02 GMT
ETag: "1a968-3ec-4e693e61bb8b6"
Cache-Control: max-age=2592000
Content-Length: 12405
Content-Type: image/jpg

CONTENIDO DE LA IMAGEN
```

Petición HTTP del ejemplo

```
GET /image.jpg HTTP/1.1
Host: sw2016.com:8080
Accept: image/*
If-Modified-Since: Tue, 17 Sep 2015 13:00:02 GMT
If-None-Match: "1a968-3ec-4e693e61bb8b6"
User-Agent: Mozilla Windows Escritorio
```

En la cabecera **If-Modified-Since** se introduce el valor de la cabecera "**Last-Modified**" devuelto en la respuesta.

En la cabecera **If-None-Match** se introduce el valor de la cabecera "**ETag**" devuelto en la respuesta.

FUNCIONAMIENTO DE HTTP:

CACHÉ – VALIDACIÓN EN SERVIDOR

- Si desde la fecha indicada en ***If-Modified-Since*** la imagen no ha cambiado y el código en la cabecera ***If-None-Match*** coincide con la etiqueta *Etag* asignada por el servidor a la imagen, entonces el servidor generará una respuesta con código **304 Not Modified** y no se enviara ningún contenido en el mensaje de respuesta.

Respuesta HTTP del ejemplo

```
HTTP/1.1 304 Not Modified
Date: Thu, 20 Jan 2016 20:25:52 GMT
Last-Modified: Tue, 17 Sep 2015 13:00:02 GMT
ETag: "1a968-3ec-4e693e61bb8b6"
Cache-Control: max-age=2592000
```

FUNCIONAMIENTO DE HTTP:

CACHÉ - EJERCICIO

Un servidor dispone de un recurso con la siguiente URI <http://localhost:8080/SW2016/servlet/HolaMundoCache>

NOTA : el recurso no se modifica en ninguno de los casos que se plantean a continuación

CASO 1: En un instante dado, un usuario solicita el recurso a través del navegador; cliente y servidor intercambian el siguiente tráfico HTTP:

```
GET /SW2016/servlet/HolaMundoCache HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:44.0) Gecko/20100101 Firefox/44.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: es-ES;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Date: Tue, 17 May 2016 19:19:05 GMT
Last-Modified: Tue, 17 May 2016 19:19:05 GMT
Cache-Control: max-age=3600
Content-Type: text/html; charset=utf-8
Content-Length: 475

<!DOCTYPE html><html>...</html>
```

CASO 2: 61 minutos después el usuario realiza una segunda la petición; cliente y servidor intercambian el siguiente tráfico HTTP:

```
GET /SW2016/servlet/HolaMundoCache HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:44.0) Gecko/20100101 Firefox/44.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: es-ES;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: keep-alive
If-Modified-Since: Tue, 17 May 2016 19:19:05 GMT
```

```
HTTP/1.1 304 Not Modified
Server: Apache-Coyote/1.1
Date: Tue, 17 May 2016 20:20:05 GMT
Last-Modified: Tue, 17 May 2016 19:19:05 GMT
Cache-Control: max-age=3600
```

¿Por qué la respuesta es diferente si la petición del navegador es la misma?