

# Gestión del almacenamiento en caché

Generalmente en una aplicación, el contenido de algunos recursos cambia con muy poca frecuencia. Para optimizar las transferencias entre el cliente y el servidor, el navegador puede almacenar temporalmente estos recursos. Cuando tiene que realizar de nuevo una petición HTTP para obtener uno de estos recursos, puede usar la versión que ha almacenado localmente en vez de contactar de nuevo con el servidor. Sin embargo, tiene que asegurarse previamente que la versión del recurso que mantiene localmente sigue siendo válida. Para ello, puede usar varias técnicas como una petición condicional o realizar previamente una petición HEAD para obtener únicamente la parte de la cabecera de la respuesta HTTP por parte del servidor.

## 1. Gestión realizada por el cliente

Hay varias cabeceras de petición HTTP para darle un carácter condicional a una petición HTTP. Este carácter condicional lleva, generalmente, una fecha o un identificador asociado al recurso.

La primera solución para el navegador consiste en añadir la cabecera `if-modified-since` seguida de la fecha y hora de creación del recurso que ya tiene en caché. De este modo, solicita al servidor que le transfiera el recurso si éste ha sido modificado posteriormente a esta fecha. Si éste es el caso, el servidor devuelve al cliente el recurso en una respuesta HTTP 200. Si el recurso no se ha modificado en el servidor desde esa fecha, éste genera una respuesta HTTP 304 y el navegador utiliza la versión del recurso que ya tenía en la memoria caché.

La segunda solución es algo similar, pero además usa una etiqueta asociada por el servidor al recurso. Éste modifica el valor de la etiqueta en cada modificación del recurso.

Para generar este tipo de petición, el navegador añade la cabecera `if-none-match` seguida de la etiqueta correspondiente a la versión del recurso que tiene en caché. Como sucede con la solución anterior, el servidor devuelve al cliente el recurso si no tiene la misma versión. Si no, el servidor genera una respuesta HTTP 304.

## 2. Gestión realizada por el servidor

Para que los dos mecanismos mostrados en el apartado anterior funcionen, el servidor tiene que añadir la información necesaria en las respuestas que realiza al cliente. Esta información se añade mediante las cabeceras de respuesta siguientes:

**date:** esta cabecera contiene la fecha y hora de generación de la respuesta HTTP. El navegador la utiliza como referencia para realizar peticiones HTTP condicionales.

**Etag:** esta cabecera representa el valor de la etiqueta asociada por el servidor al recurso cuando éste se transmite al cliente.

**expires:** con esta cabecera el servidor indica al navegador hasta cuando puede conservar en caché el recurso enviado. Una vez que la fecha haya pasado, el navegador tiene que realizar una nueva petición HTTP si necesita usar este recurso de nuevo.

El servidor también puede impedir el almacenamiento en caché de un recurso por el navegador añadiendo la cabecera de respuesta `cache-control` con el valor `no-cache`.

Ese capítulo no pretende reemplazar la norma RFC2616 que describe detalladamente en varias centenas de páginas el funcionamiento del protocolo HTTP. Simplemente permite comprender mejor los elementos que vamos a manipular en los capítulos siguientes.