

## HTTP STATUS CODES

### Información – 100

#### 100 – Continue

El navegador puede continuar realizando su petición (se utiliza para indicar que la primera parte de la petición del navegador se ha recibido correctamente).

#### 101 – Switching protocols

El servidor acepta el cambio de protocolo propuesto por el navegador (puede ser por ejemplo un cambio de HTTP 1.0 a HTTP 1.1).

#### 102 – Processing

El servidor está procesando la petición del navegador pero todavía no ha terminado (esto evita que el navegador piense que la petición se ha perdido cuando no recibe ninguna respuesta).

#### 103 – Early hints

Se va a reanudar una petición `POST` o `PUT` que fue abortada previamente.

### Success – 200

#### 200 – OK

Respuesta estándar para peticiones correctas.

#### 201 – Created

La petición ha sido completada y ha resultado en la creación de un nuevo recurso.

#### 202 – Accepted

La petición ha sido aceptada para procesamiento, pero este no ha sido completado. La petición eventualmente puede no haber sido satisfecha, ya que podría ser no permitida o prohibida cuando el procesamiento tenga lugar.

#### 204 – No content

La petición se ha completado con éxito pero su respuesta no tiene ningún contenido (la respuesta puede incluir información en sus cabeceras HTTP).

#### 206 – Partial Content

La petición servirá parcialmente el contenido solicitado. Esta característica es utilizada por herramientas de descarga como wget para continuar la transferencia de descargas anteriormente interrumpidas, o para dividir una descarga y procesar las partes simultáneamente.

## Redirect – 300

### Multiple choices – 300

Indica opciones múltiples para el URI que el cliente podría seguir. Esto podría ser utilizado, por ejemplo, para presentar distintas opciones de formato para video, listar archivos con distintas extensiones o word sense disambiguation.

### Moved permanently – 301

Esta y todas las peticiones futuras deberían ser dirigidas a la URL dada.

### Not modified – 304

Indica que la petición a la URL no ha sido modificada desde que fue requerida por última vez. Típicamente, el cliente HTTP provee un encabezado como If-Modified-Since para indicar una fecha y hora contra la cual el servidor pueda comparar. El uso de este encabezado ahorra ancho de banda y procesamiento tanto del servidor como del cliente.

### Temporary redirect – 307

Se trata de una redirección que debería haber sido hecha con otra URI, sin embargo aún puede ser procesada con la URI proporcionada. En contraste con el código 303, el método de la petición no debería ser cambiado cuando el cliente repita la solicitud. Por ejemplo, una solicitud POST tiene que ser repetida utilizando otra petición POST.

### Permanent redirect – 308

El recurso solicitado por el navegador se encuentra en otro lugar y este cambio es permanente. A diferencia del código 301, no se permite cambiar el método HTTP para la nueva petición (así por ejemplo, si envías un formulario a un recurso que ha cambiado de lugar, todo seguirá funcionando bien).

## Client Error – 400

### Bad request – 400

El servidor no procesará la solicitud, porque no puede, o no debe, debido a algo que es percibido como un error del cliente (ej: solicitud malformada, sintaxis errónea, etc). La solicitud contiene sintaxis errónea y no debería repetirse.

### Unauthorized – 401

Similar al 403 Forbidden, pero específicamente para su uso cuando la autenticación es posible pero ha fallado o aún no ha sido provista. Vea autenticación HTTP básica y Digest access authentication.

### Forbidden – 403

La solicitud fue legal, pero el servidor rehúsa responderla dado que el cliente no tiene los privilegios para realizarla. En contraste a una respuesta 401 No autorizado, autenticarse previamente no va a cambiar la respuesta.

#### Not found – 404

Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.

#### Conflict – 409

Indica que la solicitud no pudo ser procesada debido a un conflicto con el estado actual del recurso que esta identifica.

#### Server Error – 500

##### Internal server error – 500

Es un código comúnmente emitido por aplicaciones empujadas en servidores web, mismas que generan contenido dinámicamente, por ejemplo aplicaciones montadas en IIS o Tomcat, cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor web.

##### Not implemented – 501

El servidor no soporta alguna funcionalidad necesaria para responder a la solicitud del navegador (como por ejemplo el método utilizado para la petición).

##### Bad gateway – 502

El servidor está actuando de proxy o gateway y ha recibido una respuesta inválida del otro servidor, por lo que no puede responder adecuadamente a la petición del navegador.

##### Service unavailable – 503

El servidor no puede responder a la petición del navegador porque está congestionado o está realizando tareas de mantenimiento.

##### Gateway timeout – 504

El servidor está actuando de proxy o gateway y no ha recibido a tiempo una respuesta del otro servidor, por lo que no puede responder adecuadamente a la petición del navegador.

## **Bibliografía**

- [https://es.wikipedia.org/wiki/Anexo:C%C3%B3digos\\_de\\_estado\\_HTTP](https://es.wikipedia.org/wiki/Anexo:C%C3%B3digos_de_estado_HTTP)