*apache: httpd.conf*

*Customizable error responses come in three flavors:*

*# 1) plain text 2) local redirects 3) external redirects*

*#*

*# Some examples:*

*#ErrorDocument 500 "The server made a boo boo."*

*#ErrorDocument 404 /missing.html*

*#ErrorDocument 404 "/cgi-bin/missing\_handler.pl"*

*#ErrorDocument 402 http://www.example.com/subscription\_info.html*

**Códigos de estado HTTP**

La siguiente es una lista de códigos de respuesta del [HTTP](https://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol) y, separadas por un guión, la frase estándar asociadas a ese código de respuesta en el protocolo HTTP. Estas frases están destinadas a dar una descripción intuitiva del estatus. Los códigos de estatus están especificados por el RFC 2616 (actualmente obsoleto), y algunos fragmentos en los estándares [RFC 2518](https://tools.ietf.org/html/rfc2518) (también obsoleto), [RFC 2817](https://tools.ietf.org/html/rfc2817), [RFC 2295](https://tools.ietf.org/html/rfc2295) (experimental), [RFC 2774](https://tools.ietf.org/html/rfc2774) (también experimental) y [RFC 4918](https://tools.ietf.org/html/rfc4918); otros no están estandarizados, pero son comúnmente utilizados.

**1xx: Respuestas informativas**: Petición recibida, continuando proceso. Esta respuesta significa que el servidor ha recibido los encabezados de la petición, y que el cliente debería proceder a enviar el cuerpo de la misma. Si el cuerpo de la petición es largo, es ineficiente enviarlo a un servidor, cuando la petición ha sido ya rechazada, debido a encabezados inapropiados. Para hacer que un servidor chequee si la petición podría ser aceptada basada únicamente en los encabezados de la petición, el cliente debe enviar Expect: 100-continue como un encabezado en su petición inicial y verificar si un código de estado 100 Continue es recibido en respuesta, antes de continuar.

* 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.
* 103 Checkpoint: Se va a reanudar una petición POST o PUT que fue abortada previamente.

**2xx: Peticiones correctas**: Esta clase de código de estado indica que la petición fue recibida correctamente, entendida y aceptada.

* 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 pudiere no ser satisfecha, ya que podría ser no permitida o prohibida cuando el procesamiento tenga lugar.
* 203 Non-Authoritative Information:La petición se ha completado con éxito, pero su contenido no se ha obtenido de la fuente originalmente solicitada sino de otro servidor.
* 204 No Content: La petición se ha completado con éxito pero su respuesta no tiene ningún contenido.
* 205 Reset Content: La petición se ha completado con éxito, pero su respuesta no tiene contenidos y además, el navegador tiene que inicializar la página desde la que se realizó la petición.
* 206 Partial Content: La petición servirá parcialmente el contenido solicitado. Esta característica es utilizada por herramientas de descarga como [wget](https://es.wikipedia.org/wiki/Wget) para continuar la transferencia de descargas anteriormente interrumpidas, o para dividir una descarga y procesar las partes simultáneamente.
* 207 Multi-Status: El cuerpo del mensaje que sigue es un mensaje [XML](https://es.wikipedia.org/wiki/XML) y puede contener algún número de códigos de respuesta separados, dependiendo de cuántas sub-peticiones sean hechas.
* 208 Already Reported: El listado de elementos DAV ya se notificó previamente, por lo que no se van a volver a listar.

**3xx: Redirecciones**: El cliente tiene que tomar una acción adicional para completar la petición. Esta clase de código de estado indica que una acción subsecuente necesita efectuarse por el [agente de usuario](https://es.wikipedia.org/wiki/User_agent) para completar la petición. La acción requerida puede ser llevada a cabo por el agente de usuario sin interacción con el usuario si y solo si el método utilizado en la segunda petición es GET o HEAD. El agente de usuario no debe redirigir automáticamente una petición más de 5 veces, dado que tal funcionamiento indica usualmente un [Bucle infinito](https://es.wikipedia.org/wiki/Bucle_infinito).

* 300 Multiple Choices: Indica opciones múltiples para el [URI](https://es.wikipedia.org/wiki/Uniform_Resource_Identifier) 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](https://es.wikipedia.org/w/index.php?title=Word_sense_disambiguation&action=edit&redlink=1).
* [301 Moved Permanently](https://es.wikipedia.org/wiki/HTTP_301): Esta y todas las peticiones futuras deberían ser dirigidas a la URL dada. (el dominio no lo debería de usar)
* 302 Found: Este es el código de redirección más popular, pero también un ejemplo de las prácticas de la industria contradiciendo el estándar. La especificación HTTP/1.0 requería que el cliente realizara una redirección temporal, pero los navegadores populares lo implementaron como 303 See Other. Por tanto, HTTP/1.1 añadió códigos de estado 303 y 307 para eliminar la ambigüedad entre ambos comportamientos. Sin embargo, la mayoría de aplicaciones web y bibliotecas de desarrollo aún utilizan el código de respuesta 302 como si fuera el 303.
* 303 See Other: La respuesta a la petición puede ser encontrada bajo otra URI utilizando el método GET.
* 304 Not Modified: 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 reprocesamiento tanto del servidor como del cliente.
* 305 Use Proxy: Muchos clientes HTTP no se apegan al estándar al procesar respuestas con este código, principalmente por motivos de seguridad.
* 306 Switch Proxy: Este código se utilizaba en las versiones antiguas de HTTP pero ya no se usa.
* 307 Temporary Redirect: 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.
* 308 Permanent Redirect: 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. (el dominio no lo tengo que usar)

**4xx: Errores del cliente**: La solicitud contiene sintaxis incorrecta o no puede procesarse. La intención de la clase de códigos de respuesta 4xx es para casos en los cuales el cliente parece haber errado la petición. Excepto cuando se responde a una petición HEAD, el servidor debe incluir una entidad que contenga una explicación a la situación de error, y si es una condición temporal o permanente. Estos códigos de estado son aplicables a cualquier método de solicitud. Los agentes de usuario deben desplegar cualquier entidad al usuario. Estos son típicamente los códigos de respuesta de error más comúnmente encontrados.

* 400 Bad Request: El servidor no procesará la solicitud, porque no puede, o no debe, debido a algo que es percibido como un error del cliente. La solicitud contiene sintaxis errónea y no debería repetirse.
* 401 Unauthorized: Similar al 403 Forbidden, pero específicamente para su uso cuando la autentificación es posible pero ha fallado o aún no ha sido provista. Vea [autenticación HTTP básica](https://es.wikipedia.org/wiki/Autenticaci%C3%B3n_de_acceso_b%C3%A1sica) y [Digest access authentication](https://es.wikipedia.org/wiki/Digest_access_authentication).
* 402 Payment Required: La intención original era que este código pudiese ser usado como parte de alguna forma o esquema de [Dinero electrónico](https://es.wikipedia.org/wiki/Dinero_electr%C3%B3nico) o [micropagos](https://es.wikipedia.org/wiki/Micropago), pero eso no sucedió, y este código nunca se utilizó.
* [403 Forbidden](https://es.wikipedia.org/wiki/HTTP_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.
* [404 Not Found](https://es.wikipedia.org/wiki/HTTP_404): Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.
* 405 Method Not Allowed: Una petición fue hecha a una URI utilizando un método de solicitud no soportado por dicha URI; por ejemplo, cuando se utiliza GET en un formulario que requiere que los datos sean presentados vía POST, o utilizando PUT en un recurso de solo lectura.
* 406 Not Acceptable: El servidor no es capaz de devolver los datos en ninguno de los formatos aceptados por el cliente, indicados por éste en la cabecera "Accept" de la petición.
* 407 Proxy Authentication Required
* 408 Request Timeout: El cliente falló al continuar la petición - excepto durante la ejecución de videos Adobe Flash cuando solo significa que el usuario cerró la ventana de video o se movió a otro. [ref](http://livedocs.adobe.com/fms/2/docs/00000338.html)
* 409 Conflict: Indica que la solicitud no pudo ser procesada debido a un conflicto con el estado actual del recurso que esta identifica.
* [410 Gone](https://es.wikipedia.org/wiki/HTTP_410): Indica que el recurso solicitado ya no está disponible y no lo estará de nuevo. Debería ser utilizado cuando un recurso ha sido quitado de forma permanente. Si un cliente recibe este código no debería volver a solicitar el recurso en el futuro. Por ejemplo un buscador lo eliminará de sus índices y lo hará más rápidamente que utilizando un código 404.
* 411 Length Required: El servidor rechaza la petición del navegador porque no incluye la cabecera Content-Length adecuada.
* 412 Precondition Failed: El servidor no es capaz de cumplir con algunas de las condiciones impuestas por el navegador en su petición.
* 413 Request Entity Too Large: La petición del navegador es demasiado grande y por ese motivo el servidor no la procesa.
* 414 Request-URI Too Long: La URI de la petición del navegador es demasiado grande y por ese motivo el servidor no la procesa.
* 415 Unsupported Media Type: La petición del navegador tiene un formato que no entiende el servidor y por eso no se procesa.
* 416 Requested Range Not Satisfiable: El cliente ha preguntado por una parte de un archivo, pero el servidor no puede proporcionar esa parte, por ejemplo, si el cliente preguntó por una parte de un archivo que está más allá de los límites del fin del archivo.
* 417 Expectation Failed: La petición del navegador no se procesa porque el servidor no es capaz de cumplir con los requerimientos de la cabecera Expect de la petición.
* 418 I'm a teapot: "Soy una tetera". Este código fue definido en 1998 como una inocentada, en el Protocolo de Transmisión de Hipertexto de Cafeteras. No se espera que los servidores web implementen realmente este código de error, pero es posible encontrar sitios que devuelvan este código HTTP.
* 422 Unprocessable Entity: La solicitud está bien formada pero fue imposible seguirla debido a errores semánticos.
* 423 Locked: El recurso al que se está teniendo acceso está bloqueado.
* 424 Failed Dependency: La solicitud falló debido a una falla en la solicitud previa.
* 425 Unassigned: Definido en los drafts de [WebDav Advanced Collections](http://tools.ietf.org/html/draft-ietf-webdav-collection-protocol-04#section-5.3.2), pero no está presente en "Web Distributed Authoring and Versioning (WebDAV) Ordered Collections Protocol".
* 426 Upgrade Required: El cliente debería cambiarse a [TLS/1.0](https://es.wikipedia.org/wiki/Transport_Layer_Security).
* 428 Precondition Required: El servidor requiere que la petición del navegador sea condicional.
* 429 Too Many Requests: Hay muchas conexiones desde esta dirección de internet.
* 431 Request Header Fields Too Large: El servidor no puede procesar la petición porque una de las cabeceras de la petición es demasiado grande. Este error también se produce cuando la suma del tamaño de todas las peticiones es demasiado grande.[2](https://es.wikipedia.org/wiki/Anexo:C%C3%B3digos_de_estado_HTTP#cite_note-LibrosWeb-2)​
* 449: Una extensión de Microsoft: La petición debería ser reintentada después de hacer la acción apropiada.
* [451 Unavailable for Legal Reasons](https://es.wikipedia.org/wiki/HTTP_451): El contenido ha sido eliminado como consecuencia de una orden judicial o sentencia emitida por un tribunal.

**5xx: Errores de servidor**: El servidor falló al completar una solicitud aparentemente válida. Los códigos de respuesta que comienzan con el dígito "5" indican casos en los cuales el servidor tiene registrado aún antes de servir la solicitud, que está errado o es incapaz de ejecutar la petición. Excepto cuando está respondiendo a un método HEAD, el servidor debe incluir una entidad que contenga una explicación de la situación de error, y si es una condición temporal o permanente. Los agentes de usuario deben desplegar cualquier entidad incluida al usuario. Estos códigos de respuesta son aplicables a cualquier método de petición.

* 500 Internal Server Error: Es un código comúnmente emitido por aplicaciones empotradas en servidores web, mismas que generan contenido dinámicamente, por ejemplo aplicaciones montadas en [IIS](https://es.wikipedia.org/wiki/IIS) o [Tomcat](https://es.wikipedia.org/wiki/Tomcat), cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor web.
* 501 Not Implemented: El servidor no soporta alguna funcionalidad necesaria para responder a la solicitud del navegador.
* 502 Bad Gateway: 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.
* 503 Service Unavailable: El servidor no puede responder a la petición del navegador porque está congestionado o está realizando tareas de mantenimiento.
* 504 Gateway Timeout: 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.
* 505 HTTP Version Not Supported: El servidor no soporta o no quiere soportar la versión del protocolo HTTP utilizada en la petición del navegador.
* 506 Variant Also Negotiates: El servidor ha detectado una referencia circular al procesar la parte de la negociación del contenido de la petición.
* 507 Insufficient Storage: El servidor no puede crear o modificar el recurso solicitado porque no hay suficiente espacio de almacenamiento libre.
* 508 Loop Detected: La petición no se puede procesar porque el servidor ha encontrado un bucle infinito al intentar procesarla.
* 509 Bandwidth Limit Exceeded: Límite de ancho de banda excedido. Este código de estatus, a pesar de ser utilizado por muchos servidores, no es oficial.
* 510 Not Extended: La petición del navegador debe añadir más extensiones para que el servidor pueda procesarla.
* 511 Network Authentication Required: El navegador debe autenticarse para poder realizar peticiones.
* 512 Not updated: Este error prácticamente es inexistente en la red, pero indica que el servidor está en una operación de actualizado y no puede tener conexión.
* 521 Version Mismatch: Este error sale cuando la versión no es compatible con tu hardware