DuocUC

Cultura de Internet V

"Códigos de Estado HTTP"

Por:
Dennis Sandoval

Docente:
Luis Felipe Ramirez

Sección:
LPD3111-002D
Santiago, abril 2025

Índice

Resumen

Los códigos de respuesta HTTP son elementos esenciales en la comunicación web, ya que informan sobre el estado de una solicitud entre cliente y servidor, permitiendo detectar errores, optimizar procesos y garantizar una experiencia de usuario fluida y segura. Clasificados en cinco grupos (1xx informativos, 2xx exitosos, 3xx redirecciones, 4xx errores del cliente y 5xx errores del servidor), cada código ofrece información precisa sobre el resultado de una operación. Su correcta interpretación es clave para el diagnóstico de fallos, la protección de datos y la implementación de medidas de ciberseguridad como las cabeceras HTTP (HSTS, CSP, X-XSS-Protection, entre otras), que refuerzan la integridad y privacidad de los sitios web. Además, prácticas como la autenticación básica, la reescritura de encabezados y la personalización de políticas de seguridad permiten un control más estricto de los accesos y reducen la exposición a amenazas. En conjunto, estos mecanismos forman la base para el desarrollo de aplicaciones web robustas, seguras y confiables.

Introducción:

En el ámbito de la programación web, los códigos de respuesta HTTP son esenciales para la comunicación entre clientes y servidores. Estos códigos, representados por números de tres dígitos, indican el estado de una solicitud HTTP y son cruciales para comprender y manejar adecuadamente las respuestas del servidor. Desde los códigos de éxito que confirman una solicitud exitosa hasta los redireccionamientos y errores, cada código desempeña un papel vital en la interacción entre clientes y servidores en la web.

Este informe explora en profundidad los códigos de respuesta HTTP y su impacto en la comunicación web. Se examinan los diferentes tipos de códigos, su significado y cómo influyen en la experiencia del usuario y el funcionamiento de las aplicaciones web. Al comprender estos códigos, los desarrolladores pueden diagnosticar problemas, mejorar la usabilidad y garantizar un funcionamiento fluido de sus aplicaciones web.

¿Qué son Los Códigos de Respuesta HTTP?

"Los códigos de estado de respuesta HTTP indican si se ha completado satisfactoriamente una solicitud HTTP específica." (Códigos de Estado de Respuesta HTTP - HTTP | MDN, 2022)

Un estado de respuesta HTTP es un mensaje enviado por un servidor web en respuesta a una solicitud realizada por un cliente, como un navegador web o una aplicación. Este mensaje contiene un código de estado de tres dígitos que indica el resultado de la solicitud. Los códigos de estado informan al cliente si la solicitud fue exitosa, si hay redirecciones necesarias, si hubo errores por parte del cliente o del servidor, entre otros casos. Los estados de respuesta son fundamentales para la comunicación entre clientes y servidores en la web, ya que permiten que los clientes comprendan y reaccionen adecuadamente a las respuestas del servidor.

Sumergirse en el mundo de los códigos de respuesta HTTP es adentrarse en el corazón mismo de la interacción entre clientes y servidores en la web. Estos códigos, cada uno representando un estado específico de la solicitud realizada por el cliente, son fundamentales para comprender y navegar el vasto paisaje de la comunicación en línea. Desde los códigos de éxito, que confirman que una solicitud ha sido recibida y procesada satisfactoriamente (los venerados 2xx), hasta los redireccionamientos (la misteriosa categoría 3xx) que orientan al cliente hacia nuevas direcciones, y los errores tanto del cliente (los a veces esquivos 4xx) como del servidor (los temidos 5xx), cada uno de estos códigos pinta una parte crucial del panorama digital.

Estos códigos de respuesta, al proporcionar una comunicación clara y estructurada entre

el cliente y el servidor, permiten una interacción fluida y efectiva en la web, facilitando así una experiencia de usuario óptima. Son una herramienta esencial para los desarrolladores web, que utilizan estos códigos para diagnosticar problemas, mejorar la usabilidad y garantizar el funcionamiento sin problemas de aplicaciones y sitios web dinámicos e interactivos. Como tal, los códigos de respuesta HTTP son la columna vertebral de la comunicación en línea, asegurando que la web funcione de manera eficiente y que los usuarios puedan navegar sin problemas.

Respuestas informativas (1xx)

“La clase de código de estado 1xx (Informativo) indica una respuesta provisional para comunicar el estado de la conexión o el progreso de la solicitud antes de completar la acción solicitada y enviar una respuesta final. ” (Fielding et al., 2022)

Comprendidas desde el código de respuesta 100 a la 199, son las llamadas respuestas informativas (o *Informational* en inglés). Los códigos de estado 1xx indican respuestas provisionales en HTTP, las cuales consisten únicamente en la Línea de Estado y encabezados opcionales, finalizando con una línea vacía. No hay encabezados requeridos para esta clase de código de estado. Es importante destacar que, dado que HTTP/1.0 no definió ningún código de estado 1xx, los servidores no deben enviar una respuesta 1xx a un cliente HTTP/1.0, excepto en condiciones experimentales.

El cliente debe estar preparado para aceptar una o más respuestas de estado 1xx antes de una respuesta regular, incluso si no espera un mensaje de estado 100 (Continuar). Sin embargo, las respuestas de estado 1xx inesperadas pueden ser ignoradas por un agente de usuario.

Los proxies deben reenviar las respuestas 1xx, a menos que la conexión entre el proxy y su cliente se haya cerrado o a menos que el propio proxy haya solicitado la generación de la respuesta 1xx. Por ejemplo, si un proxy agrega un campo "Expect: 100-continue" cuando reenvía una solicitud, entonces no es necesario que reenvíe la(s) respuesta(s) 100 (Continuar) correspondiente(s).

En resumen, los códigos de estado 1xx proporcionan una comunicación provisional entre el cliente y el servidor, indicando que la petición ha sido recibida y que el proceso debe continuar. Esto permite una interacción eficiente entre los componentes de la comunicación HTTP.

100 Continue

"Esta respuesta provisional indica que todo hasta ahora está bien y que el cliente debe continuar con la solicitud o ignorarla si ya está terminada." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor ha recibido la parte inicial de una solicitud y no la ha rechazado todavía. Este código se utiliza cuando el cliente ha enviado una solicitud con la expectativa de recibir una confirmación (100-continue). En ese caso, el cliente debe continuar enviando la solicitud sin esperar una respuesta final de 100. Si la solicitud no incluye esta expectativa, el cliente puede simplemente ignorar esta respuesta provisional y continuar con su solicitud. Una vez completada la solicitud, el servidor enviará una respuesta final al cliente.

101 Switching Protocol

"Este código se envía en respuesta a un encabezado de solicitud por el cliente e indica que el servidor acepta el cambio de protocolo propuesto por el agente de usuario." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor está dispuesto a cambiar el protocolo de aplicación utilizado en la conexión en respuesta a la solicitud del cliente, especificada mediante el campo de cabecera Upgrade. Este cambio de protocolo se realiza inmediatamente después de la línea vacía que termina la respuesta 101. El servidor debe generar un campo de cabecera Upgrade en la respuesta para indicar qué protocolo(s) estarán como tal después de este cambio. Este código se utiliza cuando el servidor considera

ventajoso cambiar el protocolo de conexión. Por ejemplo, puede ser beneficioso cambiar a una versión más reciente de HTTP o a un protocolo síncrono en tiempo real, especialmente al entregar recursos que requieran estas características.

Respuestas satisfactorias (2xx)

"El código de estado de clase 2xx (correcto) indica que la solicitud del cliente se ha recibido, comprendido y aceptado correctamente." (Fielding et al., 2022)

Comprendidas desde el código de respuesta 200 a la 299, son las llamadas respuestas satisfactorias (o *Successful* en inglés). Los códigos de estado 2xx en HTTP indican respuestas exitosas. Esto significa que la solicitud del cliente ha sido recibida, comprendida y aceptada correctamente por el servidor. Estas respuestas confirman que la operación solicitada se ha completado satisfactoriamente.

Por ejemplo, el código de estado 200 OK señala que la solicitud ha tenido éxito y el servidor está devolviendo la información solicitada. Dependiendo del método utilizado en la solicitud, como GET, HEAD, POST o TRACE, la información devuelta puede variar.

En el caso del código de estado 201 Created, se informa que la solicitud ha dado lugar a la creación exitosa de un nuevo recurso. La respuesta incluye la URI del nuevo recurso creado, así como información adicional sobre sus características.

La respuesta 202 Accepted indica que la solicitud ha sido aceptada para su procesamiento, pero aún no se ha completado. Esto puede ocurrir en situaciones donde el procesamiento puede llevar tiempo o ser parte de un proceso en segundo plano. Como tal, los códigos de estado 2xx en HTTP indican que la solicitud del cliente se ha procesado con éxito y que el servidor ha respondido de manera adecuada. Estas respuestas son fundamentales para garantizar una comunicación efectiva entre el cliente y el servidor en la web.

200 OK

"La solicitud ha tenido éxito. El significado de un éxito varía dependiendo del método HTTP." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de estado indica que la solicitud se ha completado con éxito. La información devuelta en la respuesta varía según el método utilizado en la solicitud. Por ejemplo, para GET se envía una entidad correspondiente al recurso solicitado, mientras que para HEAD se envían los campos de cabecera de entidad sin cuerpo del mensaje. Para POST, se proporciona una entidad que describe o contiene el resultado de la acción, y para TRACE, se incluye una entidad que contiene el mensaje de solicitud tal como fue recibido por el servidor final. Como tal, el código 200 indica que la solicitud ha tenido éxito y que la información devuelta depende del método de solicitud utilizado. Esto permite una comunicación eficaz entre cliente y servidor, con respuestas específicas adaptadas a cada tipo de solicitud.

201 Created

"La solicitud ha tenido éxito y se ha creado un nuevo recurso como resultado de ello. Ésta es típicamente la respuesta enviada después de una petición PUT." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de estado indica que la solicitud se ha cumplido y ha dado como resultado la creación de uno o más recursos nuevos. La respuesta incluye información sobre la ubicación del recurso recién creado, normalmente proporcionada a través del campo de cabecera *Location*. Además, la respuesta puede contener una entidad que describa y enlace a los recursos creados, junto con cualquier campo de validación que transmita los valores

actuales para una nueva representación creada por la solicitud. Este código es fundamental para indicar el éxito en la creación de recursos en el servidor y proporcionar una referencia directa a ellos para su posterior manipulación o consulta por parte del cliente.

202 Accepted

"La solicitud se ha recibido, pero aún no se ha actuado. Es una petición "sin compromiso", lo que significa que no hay manera en HTTP que permita enviar una respuesta asíncrona que indique el resultado del procesamiento de la solicitud." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de estado indica que la solicitud ha sido aceptada para su procesamiento, pero dicho procesamiento aún no se ha completado. La solicitud puede o no ser finalmente atendida, ya que podría ser rechazada cuando se lleve a cabo el procesamiento real. Este código no proporciona una garantía de que la solicitud se cumplirá, ya que no existe una facilidad en HTTP para volver a enviar un código de estado desde una operación asíncrona.

El propósito del código 202 es permitir que un servidor acepte una solicitud para algún otro proceso, como un proceso orientado por lotes que se ejecuta una vez al día, sin requerir que la conexión del agente de usuario con el servidor persista hasta que se complete el proceso. La entidad devuelta con esta respuesta debería incluir una indicación del estado actual de la solicitud y, preferiblemente, un enlace a un monitor de estado o una estimación de cuándo el usuario puede esperar que se cumpla la solicitud.

203 Non-Authoritative Information

"La petición se ha completado con éxito, pero su contenido no se ha obtenido de la fuente originalmente solicitada, sino que se recoge de una copia local o de un tercero." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de estado indica que la solicitud se ha completado con éxito, pero el contenido devuelto no se ha obtenido de la fuente originalmente solicitada, sino que proviene de una copia local o de un tercero. Esta información metainformativa en el encabezado de entidad no es el conjunto definitivo disponible desde el servidor de origen, sino que se ha recopilado de una copia local o de terceros. El conjunto presentado puede ser un subconjunto o un superconjunto de la versión original. El uso de este código de respuesta no es obligatorio y solo es apropiado cuando la respuesta de otro modo sería 200 (OK).

El propósito del código 203 es permitir que un proxy transformador notifique a los destinatarios cuando se ha aplicado una transformación al contenido, ya que ese conocimiento podría afectar decisiones futuras con respecto al contenido. Por ejemplo, las solicitudes futuras de validación de caché para el contenido solo pueden ser aplicables a lo largo de la misma ruta de solicitud (a través de los mismos proxies). Un 203 se considera cacheable de manera heurística, lo que significa que puede almacenarse en caché a menos que se indique lo contrario por las definiciones del método o los controles de caché explícitos.

204 No Content

"La petición se ha completado con éxito pero su respuesta no tiene ningún contenido, aunque los encabezados pueden ser útiles. El agente de usuario puede actualizar sus encabezados en caché para este recurso con los nuevos valores." (*Códigos de Estado de*

Respuesta HTTP - HTTP | MDN, 2022b)

Es decir, este código de estado indica que el servidor ha cumplido con éxito la solicitud, pero no necesita devolver un cuerpo de entidad. En su lugar, puede incluir nueva o actualizada metainformación en forma de encabezados de entidad, asociados con la variante solicitada. Este código se utiliza para acciones que no requieren cambios en la vista de documento activo del agente de usuario, como guardar un documento, y se asume que el agente de usuario proporcionará alguna indicación al usuario sobre el éxito de la acción. Además, el 204 permite al servidor indicar que la acción se ha aplicado correctamente al recurso objetivo, mientras implica que el agente de usuario no necesita cambiar su vista de documento actual.

205 Reset Content

La petición se ha completado con éxito, pero su respuesta no tiene contenidos y además, el agente de usuario tiene que inicializar la página desde la que se realizó la petición, este código es útil por ejemplo para páginas con formularios cuyo contenido debe borrarse después de que el usuario lo envíe. (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor ha cumplido con la solicitud y desea que el agente de usuario restablezca la "vista del documento" que causó el envío de la solicitud a su estado original tal como fue recibido del servidor de origen. Este código está diseñado para admitir un caso de uso común de entrada de datos, donde el usuario recibe contenido que respalda la entrada de datos (un formulario, bloc de notas, lienzo, etc.), ingresa

o manipula datos en ese espacio, envía los datos ingresados en una solicitud, y luego el mecanismo de entrada de datos se restablece para la siguiente entrada, de modo que el usuario pueda iniciar fácilmente otra acción de entrada. Dado que el código de estado 205 implica que no se proporcionará contenido adicional, un servidor no debe generar contenido en una respuesta 205.

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." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código indica que el servidor está cumpliendo con éxito una solicitud de rango para el recurso objetivo transfiriendo una o más partes de la representación seleccionada. Este código es utilizado cuando el servidor recibe una solicitud GET parcial y desea transferir solo una porción específica de la representación solicitada. El servidor debe incluir en la respuesta el encabezado *Content-Range* que indica el rango incluido en la respuesta, o un tipo de contenido multipart/byteranges si se están transfiriendo múltiples partes. Además, debe incluir los encabezados requeridos como *Date, ETag, Cache-Control, Expires, Content-Location y Vary*.

Cuando se envía una respuesta 206, el servidor debe garantizar que el contenido devuelto coincida con el rango solicitado y que se incluyan los encabezados necesarios para que el cliente comprenda y procese adecuadamente la respuesta parcial. Si el cliente recibe varias respuestas parciales, puede combinarlas en un rango continuo si comparten el mismo

validador fuerte. En tales casos, el cliente puede procesar la respuesta combinada como una respuesta completa (200 OK) si cubre todo el rango solicitado, o como una respuesta parcial (206 Partial Content) si solo cubre una porción del rango solicitado.

Redirecciones (3xx)

"La clase de código de estado 3xx (Redirección) indica que el agente de usuario debe realizar más acciones para completar la solicitud." (Fielding et al., 2022)

Comprendidas desde el código de respuesta 300 a la 399, son las llamadas redirecciones (o *Redirection* en inglés). La clase de códigos de estado 3xx en HTTP, conocida como Redirection (Redirección), indica que se necesita tomar acciones adicionales por parte del agente de usuario para cumplir con la solicitud. Estos códigos abarcan varios tipos de redireccionamientos: redirecciones que indican que este recurso podría estar disponible en una URI diferente, como se proporciona en el campo de encabezado *Location*, como en los códigos de estado 301 (Movido Permanentemente), 302 (Encontrado), 307 (Redirección Temporal) y 308 (Redirección Permanente); redirección que ofrece una elección entre recursos coincidentes capaces de representar este recurso, como en el código de estado 300 (Elecciones Múltiples); redirección a un recurso diferente, identificado por el campo de encabezado *Location*, que puede representar una respuesta indirecta a la solicitud, como en el código de estado 303 (Ver Otro); y redirección a un resultado previamente almacenado, como en el código de estado 304 (No Modificado). Si se proporciona un campo de encabezado *Location*, el agente de usuario puede redirigir automáticamente su solicitud a la URI referenciada por el valor del campo *Location*, incluso si el código de estado específico no es entendido. Sin embargo, la redirección automática debe realizarse con precaución para los métodos no conocidos como seguros, ya que el usuario puede no desear redirigir una solicitud insegura. Cuando sigue automáticamente una solicitud redirigida, el agente de usuario deberá reenviar el mensaje de solicitud original con las siguientes modificaciones: reemplazar la URI de destino con la URI referenciada por el valor del campo *Location* de la respuesta de

redirección después de resolverla en relación con la URI de destino original de la solicitud original; eliminar los campos de encabezado que fueron generados automáticamente por la implementación, reemplazándolos con valores actualizados según corresponda a la nueva solicitud; cambiar el método de solicitud de acuerdo con la semántica del código de estado de redirección, si corresponde; y si el método de solicitud ha sido cambiado a GET o HEAD, eliminar los campos de encabezado específicos de contenido. Un cliente debería detectar e intervenir en redirecciones cíclicas (es decir, bucles de redirección "infinitos"). Además, es importante tener en cuenta que existen limitaciones para el número máximo de redirecciones que un cliente debería seguir para evitar problemas de rendimiento y seguridad.

300 Multiple Choice

"Esta solicitud tiene más de una posible respuesta. User-Agent o el usuario debe escoger uno de ellos. No hay forma estandarizada de seleccionar una de las respuestas." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el recurso objetivo tiene más de una representación, cada una con su propio identificador más específico, y se proporciona información sobre las alternativas para que el usuario o agente de usuario pueda seleccionar una representación preferida redirigiendo su solicitud a uno o más de esos identificadores. En otras palabras, el servidor desea que el agente de usuario participe en una negociación reactiva para seleccionar la(s) representación(es) más adecuada(s) para sus necesidades.

Si el servidor tiene una elección preferida, debe generar un campo de encabezado *Location* que contenga una referencia URI de la elección preferida. El agente de usuario puede utilizar el valor del campo *Location* para la redirección automática.

Para los métodos de solicitud que no sean HEAD, el servidor debe generar contenido en la respuesta 300 que contenga una lista de metadatos de representación y referencias URI, de las cuales el usuario o el agente de usuario pueden elegir la más preferida. El agente de usuario puede hacer una selección de esa lista automáticamente si entiende el tipo de medio proporcionado. Sin embargo, esta especificación no define un formato específico para la selección automática, ya que HTTP intenta permanecer ortogonal a la definición de su contenido. En la práctica, la representación se proporciona en un formato fácilmente analizable que se considera aceptable para el agente de usuario, según lo determinado por el diseño compartido o la negociación de contenido, o en un formato de hipertexto comúnmente aceptado. Una respuesta 300 es heurísticamente cacheable, a menos que se indique lo contrario en la definición del método o en los controles de caché explícitos.

301 Moved Permanently

"Este código de respuesta significa que la URI del recurso solicitado ha sido cambiado. Probablemente una nueva URI sea devuelta en la respuesta." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el recurso objetivo ha sido asignado a una nueva URI permanente y cualquier referencia futura a este recurso debería utilizar una de las URI devueltas. El servidor sugiere que un agente de usuario con capacidad de edición de enlaces puede reemplazar permanentemente las referencias a la URI objetivo con una de las nuevas referencias enviadas por el servidor. Sin embargo, esta sugerencia generalmente se ignora a menos que el agente de usuario esté editando activamente referencias (por ejemplo, participando en la creación de contenido), la conexión esté segura y el servidor de origen sea

una autoridad de confianza para el contenido que se está editando.

El servidor debería generar un campo de encabezado *Location* en la respuesta que contenga una referencia URI preferida para la nueva URI permanente. El agente de usuario puede utilizar el valor del campo *Location* para la redirección automática. El contenido de la respuesta del servidor generalmente contiene una breve nota de hipertexto con un hiperenlace a las nuevas URI(s). También, por razones históricas, un agente de usuario puede cambiar el método de solicitud de POST a GET para la solicitud posterior. Si este comportamiento no es deseado, se puede utilizar en su lugar el código de estado 308 (Redirección Permanente). Una respuesta 301 es heurísticamente cacheable, a menos que se indique lo contrario en la definición del método o en los controles de caché explícitos.

302 Found

"Este código de respuesta significa que el recurso de la URI solicitada ha sido cambiado temporalmente. Nuevos cambios en la URI serán agregados en el futuro. Por lo tanto, la misma URI debe ser usada por el cliente en futuras solicitudes." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el recurso objetivo reside temporalmente bajo una URI diferente. Dado que la redirección podría cambiar ocasionalmente, el cliente debería continuar utilizando la URI objetivo para futuras solicitudes.

El servidor debería generar un campo de encabezado *Location* en la respuesta que contenga una referencia URI para la URI diferente. El agente de usuario puede utilizar el valor del campo *Location* para la redirección automática. El contenido de la respuesta del

servidor generalmente contiene una breve nota de hipertexto con un hiperenlace a la URI diferente(s).También, por razones históricas, un agente de usuario puede cambiar el método de solicitud de POST a GET para la solicitud posterior. Si este comportamiento no es deseado, se puede utilizar en su lugar el código de estado 307 (Redirección Temporal).

303 See Other

"El servidor envía esta respuesta para dirigir al cliente a un nuevo recurso solicitado a otra dirección usando una petición GET." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor está redirigiendo al agente de usuario a un recurso diferente, como se indica por una URI en el campo de encabezado *Location*, con la intención de proporcionar una respuesta indirecta a la solicitud original. Un agente de usuario puede realizar una solicitud de recuperación dirigida a esa URI (una solicitud GET o HEAD si se utiliza HTTP), que también puede ser redirigida, y presentar el resultado final como una respuesta a la solicitud original. Es importante tener en cuenta que la nueva URI en el campo de encabezado *Location* no se considera equivalente a la URI de destino.

Este código de estado es aplicable a cualquier método HTTP. Se utiliza principalmente para permitir que la salida de una acción POST redirija al agente de usuario a un recurso diferente, ya que esto proporciona la información correspondiente a la respuesta POST como un recurso que puede ser identificado, marcado y almacenado en caché por separado.

Una respuesta 303 a una solicitud GET indica que el servidor de origen no tiene una representación del recurso de destino que pueda ser transferida por el servidor a través de

HTTP. Sin embargo, el valor del campo Location se refiere a un recurso que describe el recurso de destino, de modo que realizar una solicitud de recuperación en ese otro recurso podría dar como resultado una representación útil para los destinatarios sin implicar que representa el recurso de destino original. Es importante destacar que las respuestas a las preguntas sobre qué se puede representar, qué representaciones son adecuadas y qué descripción podría ser útil están fuera del alcance de HTTP. Excepto para las respuestas a una solicitud HEAD, la representación de una respuesta 303 debería contener una breve nota de hipertexto con un hipervínculo a la misma referencia URI proporcionada en el campo de encabezado *Location*.

304 Not Modified

"Esta es usada para propósitos de "caché". Le indica al cliente que la respuesta no ha sido modificada. Entonces, el cliente puede continuar usando la misma versión almacenada en su caché." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que se ha recibido una solicitud GET o HEAD condicional y habría resultado en una respuesta 200 (OK) si no fuera por el hecho de que la condición evaluada fue falsa. En otras palabras, no hay necesidad de que el servidor transfiera una representación del recurso de destino porque la solicitud indica que el cliente, que hizo la solicitud condicional, ya tiene una representación válida; por lo tanto, el servidor está redirigiendo al cliente para que haga uso de esa representación almacenada como si fuera el contenido de una respuesta 200 (OK).

El servidor que genera una respuesta 304 DEBE generar cualquiera de los siguientes campos de encabezado que se habrían enviado en una respuesta 200 (OK) a la misma

solicitud: *Content-Location, Date, ETag y Vary Cache-Control y Expires*. Dado que el objetivo de una respuesta 304 es minimizar la transferencia de información cuando el destinatario ya tiene una o más representaciones en caché, el remitente no debe generar metadatos de representación que no sean los campos mencionados anteriormente, a menos que dichos metadatos existan con el propósito de guiar las actualizaciones de caché (por ejemplo, *Last-Modified* podría ser útil si la respuesta no tiene un campo *ETag*). Si la solicitud condicional se originó con un cliente saliente, como un agente de usuario con su propia caché que envía una solicitud GET condicional a un proxy compartido, entonces el proxy debería reenviar la respuesta 304 a ese cliente. Una respuesta 304 se termina al final de la sección de encabezado; no puede contener contenido ni remolques.

306 Unused

"Este código de respuesta ya no es usado más. Actualmente se encuentra reservado. Fue usado en previas versiones de la especificación HTTP1.1." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta fue definido en una versión anterior de esta especificación, ya no se usa y el código está reservado.

307 Temporary Redirect

"El servidor envía esta respuesta para dirigir al cliente a obtener el recurso solicitado a otra URI con el mismo método que se usó la petición anterior. Tiene la misma semántica que el código de respuesta HTTP 302 Found, con la excepción de que el agente usuario no debe

cambiar el método HTTP usado: si un POST fue usado en la primera petición, otro POST debe ser usado en la segunda petición." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el recurso solicitado reside temporalmente bajo un URI diferente y que el agente de usuario no debe cambiar el método de solicitud si realiza una redirección automática a ese URI. Dado que la redirección puede cambiar con el tiempo, el cliente debería seguir utilizando el URI de destino original para futuras solicitudes.

El servidor debería generar un campo de encabezado Location en la respuesta que contenga una referencia URI al URI diferente. El agente de usuario puede utilizar el valor del campo Location para la redirección automática. El contenido de la respuesta del servidor generalmente contiene una breve nota hipertextual con un hipervínculo al URI diferente(s).

308 Permanent Redirect

"Significa que el recurso ahora se encuentra permanentemente en otra URI, especificada por la respuesta de encabezado HTTP *Location*. Tiene la misma semántica que el código de respuesta HTTP 301 Moved Permanently, con la excepción de que el agente usuario no debe cambiar el método HTTP usado: si un POST fue usado en la primera petición, otro POST debe ser usado en la segunda petición." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el recurso de destino ha sido asignado a un nuevo URI permanente y cualquier referencia futura a este recurso debería utilizar uno de

los URIs incluidos. El servidor sugiere que un agente de usuario con capacidad de edición de enlaces puede reemplazar permanentemente las referencias al URI de destino con uno de los nuevos URIs enviados por el servidor. Sin embargo, esta sugerencia generalmente se ignora a menos que el agente de usuario esté editando activamente referencias (por ejemplo, participando en la redacción de contenido), la conexión esté segura y el servidor de origen sea una autoridad de confianza para el contenido que se está editando.

El servidor debería generar un campo de encabezado *Location* en la respuesta que contenga una referencia URI preferida para el nuevo URI permanente. El agente de usuario puede utilizar el valor del campo *Location* para la redirección automática. El contenido de la respuesta del servidor generalmente contiene una breve nota hipertextual con un hipervínculo al nuevo URI(s). Una respuesta 308 es heurísticamente almacenable en caché; es decir, a menos que se indique lo contrario por la definición del método o los controles de caché explícitos. Además, este código de estado es mucho más reciente (junio de 2014) que sus códigos hermanos y, por lo tanto, es posible que no sea reconocido en todas partes.

Errores del cliente (4xx)

La clase de código de estado 4xx (Error de Cliente) indica que el cliente parece haber cometido un error. Excepto cuando se responde a una petición HEAD, el servidor debería enviar una declaración que contenga una explicación de la situación de error, y si se trata de una condición temporal o permanente. Estos códigos de estado son aplicables a cualquier método de petición. Los agentes de usuario deberán mostrar al usuario cualquier declaración incluida. (Fielding et al., 2022)

Comprendidas desde el código de respuesta 400 a la 499, son los llamados errores de clientes (o *Client Error* en inglés). Los códigos de estado 4xx en HTTP representan respuestas del servidor que señalan errores específicos del cliente. Estos indican una variedad de problemas, como solicitudes malformadas, recursos no encontrados, falta de autorización o métodos no permitidos. Por ejemplo, el código 404 (Not Found) indica que el recurso solicitado no se encuentra en el servidor, mientras que el 403 (Forbidden) señala que la solicitud fue entendida pero se rechaza su cumplimiento debido a restricciones de acceso.

Estos códigos proporcionan información crucial al cliente sobre la naturaleza del error y cómo abordarlo. Por ejemplo, el 401 (Unauthorized) sugiere que se requiere autenticación adicional, y el cliente puede intentar nuevamente la solicitud con credenciales válidas. En casos como el 404, donde un recurso no se encuentra, el cliente puede ajustar la URL solicitada o realizar acciones para corregir la referencia rota.

Como tal, los códigos de estado 4xx son una forma fundamental en la que el servidor comunica al cliente sobre errores en sus solicitudes. Estos códigos permiten una depuración

eficaz y orientan al cliente sobre cómo resolver los problemas para que la comunicación entre el cliente y el servidor pueda restablecerse de manera adecuada.

400 Bad Request

"Esta respuesta significa que el servidor no pudo interpretar la solicitud dada una sintaxis inválida." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor no puede o no procesará la solicitud debido a algo que se percibe como un error del cliente (por ejemplo, sintaxis de solicitud malformada, estructura de mensaje de solicitud inválida o enrutamiento de solicitud engañoso).

La solicitud no pudo ser comprendida por el servidor debido a una sintaxis malformada. El cliente no debería repetir la solicitud sin modificaciones.

401 Unauthorized

"Es necesario autenticar para obtener la respuesta solicitada. Esta es similar a 403, pero en este caso, la autenticación es posible." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que la solicitud no se ha aplicado porque carece de credenciales de autenticación válidas para el recurso solicitado. El servidor que genera una respuesta 401 debe enviar un campo de encabezado *WWW-Authenticate* que contenga al menos un desafío aplicable al recurso solicitado.

Si la solicitud incluía credenciales de autenticación, entonces la respuesta 401 indica

que se ha rechazado la autorización para esas credenciales. El agente de usuario puede repetir la solicitud con un campo de encabezado de *Authorization* nuevo o reemplazado. Si la respuesta 401 contiene el mismo desafío que la respuesta anterior, y el agente de usuario ya ha intentado la autenticación al menos una vez, entonces el agente de usuario debería presentar la representación incluida al usuario, ya que generalmente contiene información diagnóstica relevante.

403 Forbidden

"El cliente no posee los permisos necesarios para cierto contenido, por lo que el servidor está rechazando otorgar una respuesta apropiada." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor entendió la solicitud pero se niega a cumplirla. Un servidor que desee hacer público por qué se ha prohibido la solicitud puede describir esa razón en el contenido de la respuesta (si lo hay).

Si se proporcionaron credenciales de autenticación en la solicitud, el servidor considera que son insuficientes para otorgar acceso. El cliente no debería repetir automáticamente la solicitud con las mismas credenciales. El cliente puede repetir la solicitud con credenciales nuevas o diferentes. Sin embargo, una solicitud podría ser prohibida por razones no relacionadas con las credenciales. Un servidor de origen que desee "ocultar" la existencia actual de un recurso de destino prohibido puede, en su lugar, responder con un código de estado 404 (No encontrado).

404 Not Found

"El servidor no pudo encontrar el contenido solicitado. Este código de respuesta es uno de los más famosos dada su alta ocurrencia en la web." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor de origen no encontró una representación actual para el recurso de destino o no está dispuesto a revelar que existe una. Un código de estado 404 no indica si esta falta de representación es temporal o permanente; se prefiere el código de estado 410 (Desaparecido) sobre el 404 si el servidor de origen sabe, presumiblemente a través de algún medio configurable, que la condición es probablemente permanente. Una respuesta 404 es heurísticamente almacenable en caché; es decir, a menos que se indique lo contrario por la definición del método o controles de caché explícitos.

405 Method Not Allowed

"El método solicitado es conocido por el servidor pero ha sido deshabilitado y no puede ser utilizado. Los dos métodos obligatorios, GET y HEAD, nunca deben ser deshabilitados y no deberían retornar este código de error." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el método recibido en la línea de solicitud es conocido por el servidor de origen pero no es compatible con el recurso de destino. El servidor de origen debe generar un campo de encabezado *Allow* en una respuesta 405 que contenga una lista de los métodos actualmente admitidos por el recurso de destino. Una respuesta 405 es heurísticamente almacenable en caché; es decir, a menos que se indique

lo contrario por la definición del método o controles de caché explícitos.

406 Not Acceptable

"Esta respuesta es enviada cuando el servidor, después de aplicar una negociación de contenido servidor-impulsado, no encuentra ningún contenido seguido por la criteria dada por el usuario" (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el recurso de destino no tiene una representación actual que sea aceptable para el agente de usuario, según los campos de encabezado de negociación proactiva recibidos en la solicitud, y el servidor no está dispuesto a proporcionar una representación predeterminada.

El servidor debería generar contenido que contenga una lista de características de representación disponibles e identificadores de recursos correspondientes de los cuales el usuario o el agente de usuario pueden elegir el más apropiado. Un agente de usuario puede seleccionar automáticamente la opción más apropiada de esa lista.

407 Proxy Authentication Required

"Esto es similar al código 401, pero la autenticación debe estar hecha a partir de un proxy." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el cliente debe autenticarse primero con el proxy para realizar la solicitud. El proxy debe enviar un campo de encabezado *Proxy-Authenticate* que contenga un desafío aplicable al proxy para la solicitud. El cliente puede repetir la solicitud con un nuevo o reemplazado campo de encabezado

Proxy-Authorization.

408 Request Timeout

Esta respuesta es enviada en una conexión inactiva en algunos servidores, incluso sin alguna petición previa por el cliente. Significa que el servidor quiere desconectar esta conexión sin usar. Esta respuesta es muy usada desde algunos navegadores, como Chrome, Firefox 27+, o IE9, usa mecanismos de pre-conexión HTTP para acelerar la navegación. También hay que tener en cuenta que algunos servidores simplemente desconecta la conexión sin enviar este mensaje. (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor no recibió un mensaje de solicitud completo dentro del tiempo que estaba dispuesto a esperar. Si el cliente tiene una solicitud pendiente en tránsito, puede repetir esa solicitud. Si la conexión actual no es utilizable (por ejemplo, en HTTP/1.1 porque se pierde la delimitación de la solicitud), se utilizará una nueva conexión.

409 Conflict

"Esta respuesta puede ser enviada cuando una petición tiene conflicto con el estado actual del servidor." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que la solicitud no pudo completarse debido a un conflicto con el estado actual del recurso de destino. Este código se utiliza en situaciones donde el usuario podría resolver el conflicto y volver a enviar la solicitud. El servidor debe generar contenido que incluya suficiente información para que el usuario reconozca la fuente del conflicto.

Los conflictos son más propensos a ocurrir en respuesta a una solicitud PUT. Por ejemplo, si se estuviera utilizando el control de versiones y la representación que se está PUT incluyera cambios en un recurso que entran en conflicto con los realizados por una solicitud anterior de terceros, el servidor de origen podría usar una respuesta 409 para indicar que no puede completar la solicitud. En este caso, es probable que la representación de la respuesta contenga información útil para fusionar las diferencias basadas en el historial de revisiones.

410 Gone

"Esta respuesta puede ser enviada cuando el contenido solicitado ha sido borrado del servidor." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el acceso al recurso de destino ya no está disponible en el servidor de origen y que esta condición es probablemente permanente. Si el servidor de origen no sabe, o no tiene la capacidad para determinar, si la condición es permanente o no, se debería utilizar el código de estado 404 (Not Found) en su lugar.

La respuesta 410 está principalmente destinada a ayudar en la tarea de mantenimiento web al notificar al destinatario que el recurso está intencionalmente no disponible y que los propietarios del servidor desean que se eliminen los enlaces remotos a ese recurso. Este evento es común para servicios promocionales de tiempo limitado y para recursos pertenecientes a personas que ya no están asociadas con el sitio del servidor de origen. No es necesario marcar todos los recursos permanentemente no disponibles como "gone" o mantener la marca durante algún tiempo; eso queda a discreción del propietario del servidor. Una respuesta 410 es heurísticamente almacenable en caché; es decir, a menos que se indique lo contrario en la definición del método o en los controles de caché explícitos.

411 Length Required

"El servidor rechaza la petición porque el campo de encabezado Content-Length no está definido y el servidor lo requiere." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor se niega a aceptar la solicitud sin una longitud de contenido definida. El cliente puede repetir la solicitud si agrega un campo de encabezado *Content-Length* válido que contenga la longitud del contenido de la solicitud.

412 Precondition Failed

"El cliente ha indicado pre-condiciones en sus encabezados la cual el servidor no cumple." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que una o más condiciones especificadas en los campos de encabezado de la solicitud se evaluaron como falsas cuando se probaron en el servidor. Este código de respuesta permite al cliente establecer condiciones previas sobre el estado actual del recurso (sus representaciones y metadatos actuales) y, por lo tanto, evitar que el método de solicitud se aplique si el recurso de destino se encuentra en un estado inesperado.

413 Payload Too Large

"La entidad de petición es más larga que los límites definidos por el servidor; el servidor puede cerrar la conexión o retornar un campo de encabezado Retry-After." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor está rechazando procesar una solicitud porque el contenido de la solicitud es más grande de lo que el servidor está dispuesto o capaz de procesar. El servidor puede terminar la solicitud, si la versión del protocolo utilizada lo permite; de lo contrario, el servidor puede cerrar la conexión.

Si la condición es temporal, el servidor debería generar un campo de encabezado *Retry-After* para indicar que es temporal y después de cuánto tiempo el cliente puede intentarlo nuevamente.

414 URI Too Long

"La URI solicitada por el cliente es más larga de lo que el servidor está dispuesto a interpretar." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor está rechazando servir la solicitud porque el URI de destino es más largo de lo que el servidor está dispuesto a interpretar. Esta condición rara solo es probable que ocurra cuando un cliente ha convertido incorrectamente una solicitud POST en una solicitud GET con información de consulta larga, cuando el cliente ha caído en un bucle infinito de redirección (por ejemplo, un prefijo de URI redirigido que apunta a un sufijo de sí mismo) o cuando el servidor está bajo ataque por parte de un cliente que intenta explotar posibles agujeros de seguridad. Una respuesta 414 es heurísticamente cachéable; es decir, a menos que se indique lo contrario en la definición del método o en los controles de caché explícitos.

415 Unsupported Media Type

"El formato multimedia de los datos solicitados no está soportado por el servidor, por lo cual el servidor rechaza la solicitud." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor de origen se está negando a procesar la solicitud porque el contenido está en un formato no compatible con este método en el recurso de destino. El problema de formato puede ser debido al tipo de contenido indicado en *Content-Type* o *Content-Encoding* de la solicitud, o como resultado de la inspección directa de los datos.

Si el problema fue causado por una codificación de contenido no admitida, se debería usar el encabezado de respuesta *Accept-Encoding* para indicar qué codificaciones de contenido (si las hubiera) se habrían aceptado en la solicitud. Por otro lado, si la causa fue un tipo de medio no admitido, el encabezado de respuesta *Accep*t se puede usar para indicar qué tipos de medio se habrían aceptado en la solicitud.

416 Requested Range Not Satisfiable

"El rango especificado por el campo de encabezado Range en la solicitud no cumple; es posible que el rango está fuera del tamaño de los datos objetivo del URI." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el conjunto de rangos en el campo de encabezado de solicitud Range ha sido rechazado porque ninguno de los rangos solicitados es satisfactorio o porque el cliente ha solicitado un número excesivo de rangos pequeños o superpuestos (un posible ataque de denegación de servicio).

Cada unidad de rango define lo que se requiere para que sus propios conjuntos de rangos sean satisfactorios. Un servidor que genere una respuesta 416 a una solicitud de rango de bytes debería generar un campo de encabezado Content-Range especificando la longitud actual de la representación seleccionada. Debido a que los servidores pueden ignorar Range, muchas implementaciones responderán con la representación seleccionada completa en una respuesta 200 (OK). Esto se debe en parte a que la mayoría de los clientes están preparados para recibir un 200 (OK) para completar la tarea (aunque menos eficientemente) y en parte porque los clientes pueden no dejar de hacer una solicitud de rango inválida hasta que hayan recibido una representación completa. Por lo tanto, los clientes no pueden depender de recibir una respuesta 416 (Rango No Satisfactorio) incluso cuando sea más apropiado.

417 Expectation Failed

"Significa que la expectativa indicada por el campo de encabezado Expect solicitada no puede ser cumplida por el servidor." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b.) Es decir, este código de respuesta indica que la expectativa dada en el campo de encabezado *Expect* de la solicitud no pudo ser cumplida por al menos uno de los servidores de entrada.

418 I'm a teapot

El código de error HTTP 418 Soy una tetera indica que el servidor se rehúsa a preparar café porque es una tetera. Este error es una referencia al Hyper Text Coffee Pot Control Protocol, creado como parte de una broma del *April Fools'* de 1998. (*418 Soy una Tetera - HTTP | MDN*, 2025)

Es decir, se establece que cualquier intento de hacer café con una tetera debe generar este

código de error, y que el cuerpo de la respuesta podría ser corto y rechoncho ("*short and stout*"). Aunque no se utiliza en aplicaciones reales, se ha convertido en una curiosidad popular dentro del mundo del desarrollo web.

421 Misdirected Request

El código de estado HTTP 421 Misdirected Request indica que la solicitud fue enviada a un servidor que no está configurado para generar una respuesta autoritativa para el URI de destino. Esto puede ocurrir cuando el servidor no reconoce el origen especificado en la solicitud o cuando la conexión utilizada no coincide con el contexto requerido. Es común en situaciones donde se reutiliza una conexión TLS para múltiples dominios (por ejemplo, con certificados wildcard), y el servidor no puede manejar correctamente la solicitud debido a una mala coincidencia entre el host solicitado y la configuración del servidor.

Este tipo de error lo puede generar únicamente el servidor de origen o un gateway actuando en su nombre; los proxies no deben enviar respuestas 421. Los clientes pueden volver a intentar la solicitud sobre una nueva conexión específica para el origen del recurso o utilizando un servicio alternativo. Un ejemplo típico es con servidores Apache que usan SNI (Server Name Indication) y enfrentan problemas al manejar múltiples dominios sobre una sola conexión reutilizada.

422 Unprocessable Content

El código de estado HTTP 422 Unprocessable Content indica que el servidor ha entendido el tipo de contenido y la sintaxis de la solicitud enviada, pero no puede procesar las instrucciones que contiene. A diferencia de un error 415, aquí el problema no está en el formato del contenido, sino en su validez semántica. Es decir, aunque el contenido esté correctamente estructurado, puede incluir errores lógicos o de validación que impidan su

procesamiento, como ocurre con datos que no cumplen ciertos requisitos del servidor.

Un ejemplo típico es cuando una API espera datos codificados en Base64 con un formato muy específico, y al no cumplirse esa codificación estricta, responde con un 422. Repetir la misma solicitud sin corregir el contenido provocará el mismo error. Este tipo de respuesta ayuda a identificar fallos en la lógica de los datos enviados, más que en su estructura o tipo.

423 Locked

El código HTTP 423 Locked indica que un recurso está bloqueado y no puede ser accedido ni modificado. Este estado es utilizado principalmente en servidores que implementan WebDAV, donde los recursos pueden bloquearse para evitar conflictos de edición. La respuesta suele incluir detalles en formato XML, especificando el motivo del bloqueo, como la falta de un token de bloqueo válido. Este código no es común en navegadores, que suelen tratarlo como un error genérico 400 si llega a presentarse.

424 Failed Dependency

El código HTTP 424 Failed Dependency señala que una acción no pudo completarse porque dependía de otra operación que falló previamente. Este tipo de error es común en protocolos como WebDAV, donde, por ejemplo, si una instrucción dentro de una solicitud PROPPATCH falla, las demás acciones relacionadas también se consideran fallidas y retornan este mismo código. No es habitual en servidores web convencionales, ya que su uso está más ligado a operaciones encadenadas o dependientes dentro de sistemas colaborativos.

425 Too Early

El código HTTP 425 Too Early indica que el servidor rechazó procesar la solicitud porque consideró riesgoso hacerlo en ese momento, ya que podría tratarse de un intento de reproducción (replay attack). Este escenario puede darse cuando un cliente envía datos

tempranos (early data) aprovechando la función de 0-RTT en conexiones TLS, es decir, antes de que finalice por completo el proceso de enlace seguro. Para proteger la integridad de la comunicación, el servidor puede negarse a procesar esas solicitudes tempranas si considera que podrían repetirse de forma maliciosa.

426 Upgrade Required

El código HTTP 426 Upgrade Required indica que el servidor se niega a procesar la solicitud utilizando el protocolo actual, pero podría estar dispuesto a hacerlo si el cliente actualiza a un protocolo diferente. En esta respuesta, el servidor debe incluir un encabezado Upgrade para especificar el protocolo o protocolos requeridos. Un ejemplo típico sería un servidor que exige el uso de HTTP/3.0 para completar la solicitud, como se muestra en el encabezado de respuesta.

428 Precondition Required

El código HTTP 428 Precondition Required indica que el servidor requiere que la solicitud sea condicional. Esto generalmente ocurre cuando falta un encabezado de precondición necesario, como *If-Match*, que asegura que el cliente esté trabajando con la versión más reciente del recurso. Este código se utiliza para evitar el problema de "actualización perdida", en el cual un cliente modifica un recurso después de obtener su estado, pero antes de enviarlo de vuelta, otro cliente ya ha modificado el mismo recurso en el servidor, causando un conflicto. Al requerir solicitudes condicionales, el servidor garantiza que los clientes trabajen con las versiones correctas. Además, las respuestas con este código deben explicar cómo reenviar correctamente la solicitud, por ejemplo, sugiriendo el uso de *"If-Match"*.

429 Too Many Requests

El código HTTP 429 Too Many Requests indica que el usuario ha enviado demasiadas solicitudes en un periodo de tiempo determinado, lo que se conoce como "limitación de tasa".

La respuesta debe incluir detalles sobre la condición y, opcionalmente, un encabezado Retry-After que indique el tiempo que debe esperar el cliente antes de realizar una nueva solicitud. Por ejemplo, un servidor podría permitir solo un número específico de solicitudes por hora y responder con este código si se excede ese límite. Además, este código no define cómo se cuenta el número de solicitudes ni cómo se identifica al usuario, ya que puede basarse en diversos factores, como credenciales de autenticación o cookies. Las respuestas con este código no deben ser almacenadas en caché.

431 Request Header Fields Too Large

El código HTTP 431 Request Header Fields Too Large indica que el servidor no puede procesar la solicitud porque los campos del encabezado son demasiado grandes. La solicitud puede ser reenviada después de reducir el tamaño de los encabezados. Este código puede aplicarse tanto cuando el total de los encabezados es excesivo como cuando un solo encabezado es demasiado grande. En este último caso, la respuesta debe especificar cuál encabezado causó el problema. Las respuestas con este código no deben ser almacenadas en caché.

451 Unavailable For Legal Reasons

El código HTTP 451 Unavailable For Legal Reasons se utiliza cuando el acceso a un recurso solicitado está restringido por razones legales. Esto puede deberse a una orden judicial, una ley local o una solicitud formal que impida la visualización del contenido, como sucede frecuentemente con páginas web sujetas a censura o bloqueos regionales. Este tipo de restricción no implica necesariamente que el recurso exista o no, solo que no se puede acceder a él legalmente desde el contexto del usuario.

En las respuestas con este código, es recomendable incluir una explicación en el cuerpo del mensaje que detalle la razón legal del bloqueo. Esto podría mencionar la ley

aplicable, la entidad que impuso la restricción y a quiénes afecta. Además, es común que se incluya un encabezado o *Link* con la relación "*blocked-by*" para señalar qué entidad está aplicando la restricción, aunque la responsabilidad legal real se describe mejor dentro del cuerpo de la respuesta.

Es importante destacar que, aunque un servidor indique un bloqueo legal con el código 451, los usuarios aún podrían eludir estas restricciones mediante herramientas como redes privadas virtuales (VPN) o la red Tor. A menos que se indique lo contrario, este tipo de respuestas pueden ser almacenadas en caché según las normas del protocolo HTTP.

Errores del servidor (5xx)

Los códigos de estado de respuesta que comienzan con el dígito "5" indican casos en los que el servidor es consciente de que ha cometido un error o es incapaz de realizar la petición. Excepto cuando responde a una petición HEAD, el servidor debería incluir una entidad que contenga una explicación de la situación de error, y si se trata de una condición temporal o permanente. Los agentes de usuario deberían mostrar al usuario cualquier entidad incluida. Estos códigos de respuesta son aplicables a cualquier método de petición. (*RFC 4918: HTTP Extensions For Web Distributed Authoring And Versioning (WebDAV)*, s. f.)

Comprendidas desde el código de respuesta 500 a la 599, son los llamados errores de servidor (o *Server Error* en inglés). Los códigos de estado 5xx en HTTP indican que el servidor ha encontrado un error interno o es incapaz de completar la solicitud del cliente. Estos códigos, que van desde el 500 hasta el 599, señalan una variedad de situaciones problemáticas, como errores internos del servidor, incapacidad para manejar la solicitud debido a sobrecarga temporal o mantenimiento programado, o incapacidad para recibir una respuesta oportuna de un servidor superior o auxiliar.

Cuando un servidor devuelve un código de estado 5xx, debería incluir una representación que explique la situación del error y si es una condición temporal o permanente. Esto es especialmente importante para ayudar al cliente a comprender el problema y tomar las medidas necesarias. Por ejemplo, el 503 (Service Unavailable) podría incluir un encabezado Retry-After para indicar al cliente cuándo volver a intentar la solicitud.

Como tal, los códigos de estado 5xx indican errores del servidor que pueden ser temporales o permanentes y requieren acción por parte del cliente para manejar

adecuadamente la situación. Estas respuestas proporcionan información valiosa sobre el estado del servidor y orientan al cliente sobre cómo proceder, ya sea esperando un tiempo determinado antes de reintentar la solicitud o tomando otras medidas para abordar el problema.

500 Internal Server Error

"El servidor ha encontrado una situación que no sabe cómo manejarla." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor encontró una condición inesperada que le impidió cumplir con la solicitud.

501 Not Implemented

"El método solicitado no está soportado por el servidor y no puede ser manejado. Los únicos métodos que los servidores requieren soporte (y por lo tanto no deben retornar este código) son GET y HEAD." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor no admite la funcionalidad necesaria para cumplir con la solicitud. Esto es apropiado cuando el servidor no reconoce el método de solicitud y no es capaz de admitirlo para ningún recurso.

502 Bad Gateway

"Esta respuesta de error significa que el servidor, mientras trabaja como una puerta de enlace para obtener una respuesta necesaria para manejar la petición, obtuvo una respuesta

inválida.” (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor, mientras actúa como una puerta de enlace o proxy, recibió una respuesta inválida de un servidor de entrada al que accedió al intentar cumplir con la solicitud.

503 Service Unavailable

El servidor no está listo para manejar la petición. Causas comunes puede ser que el servidor está caído por mantenimiento o está sobrecargado. Hay que tomar en cuenta que junto con esta respuesta, una página usuario-amigable explicando el problema debe ser enviada. Estas respuestas deben ser usadas para condiciones temporales y el encabezado HTTP Retry-After: debería, si es posible, contener el tiempo estimado antes de la recuperación del servicio. El webmaster debe también cuidar los encabezados relacionados al caché que son enviados junto a esta respuesta, ya que estas respuestas de condición temporal deben usualmente no estar en el caché. (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor actualmente no puede manejar la solicitud debido a una sobrecarga temporal o mantenimiento programado, lo que probablemente se resolverá después de cierto tiempo. El servidor puede enviar un encabezado *Retry-After* para sugerir una cantidad adecuada de tiempo para que el cliente espere antes de volver a intentar la solicitud. La existencia del código de estado 503 no implica que un servidor deba usarlo cuando esté sobrecargado. Algunos servidores pueden simplemente rechazar la conexión.

504 Gateway Timeout

"Esta respuesta de error es dada cuando el servidor está actuando como una puerta de enlace y no puede obtener una respuesta a tiempo." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor, mientras actuaba como una puerta de enlace o proxy, no recibió una respuesta oportuna de un servidor ascendente al que necesitaba acceder para completar la solicitud.

505 HTTP Version Not Supported

"La versión de HTTP usada en la petición no está soportada por el servidor." (*Códigos de Estado de Respuesta HTTP - HTTP | MDN*, 2022b)

Es decir, este código de respuesta indica que el servidor no soporta, o se niega a soportar, la versión principal de HTTP que se utilizó en el mensaje de solicitud. El servidor está indicando que no puede o no desea completar la solicitud utilizando la misma versión principal que el cliente, excepto con este mensaje de error. El servidor debe generar una representación para la respuesta 505 que describa por qué esa versión no es compatible y qué otros protocolos son compatibles con ese servidor.

Consideraciones de ciberseguridad en las respuestas HTTP

Las cabeceras de seguridad HTTP son una herramienta clave para mejorar la protección de los sitios web frente a amenazas digitales. Aunque no garantizan una seguridad absoluta, sí representan una barrera eficaz contra numerosos ataques. Su activación refuerza los servidores mediante la limitación de comportamientos sospechosos, lo que convierte esta práctica en una de las más recomendables para las organizaciones que operan en el entorno online.

Estas cabeceras forman parte de las comunicaciones que se producen entre el navegador y el servidor cuando se accede a una página web. A diferencia de las cabeceras HTTP estándar, las de seguridad incluyen directrices específicas orientadas a proteger la integridad de la información y prevenir vulnerabilidades. Se convierten así en una capa adicional que actúa en favor de la privacidad y la defensa frente a ataques como el Cross-Site Scripting o el clickjacking.

Entre las cabeceras más utilizadas destacan: HTTP Strict Transport Security (HSTS), que obliga al uso de HTTPS; X-XSS Protection, que protege contra scripts maliciosos; X-Content-Type-Options, que evita la interpretación incorrecta del tipo de archivo; X-Frame-Options, que impide la carga de contenidos maliciosos mediante iframes; y Content Security Policy, que filtra recursos no autorizados. Otras cabeceras útiles incluyen Referrer Policy, Expect-CT y mecanismos como CORS, que controla el acceso desde dominios cruzados.

No obstante, estas configuraciones no son eternas. Las cabeceras pueden quedar obsoletas si los navegadores dejan de soportarlas o si surgen nuevas amenazas. Por ello, es esencial mantenerse actualizado en cuanto a tendencias y buenas prácticas en ciberseguridad.

Contar con especialistas en la materia permite a las empresas implementar y ajustar correctamente estos elementos críticos.

Por último, la colaboración con técnicos especializados, como los de SysAdminOK, garantiza una implementación eficaz de estas medidas. Esta compañía ofrece asesoramiento, configuración y mantenimiento de cabeceras de seguridad HTTP como parte de su compromiso con la seguridad de sus clientes. Gracias a este tipo de apoyo profesional, las organizaciones pueden fortalecer su infraestructura tecnológica y reducir significativamente su exposición a ciberataques.

Protección de información y gestión de errores

La Política de Seguridad del Contenido (CSP) es una capa adicional de defensa que protege las aplicaciones web contra ataques como Cross-Site Scripting (XSS) y otras formas de inyección de código. Funciona permitiendo al servidor especificar, mediante cabeceras HTTP, qué fuentes de contenido son válidas para cargar y ejecutar recursos como scripts, estilos, imágenes, etc. De esta forma, los navegadores modernos que la soportan aplican restricciones sobre qué contenido se puede mostrar o ejecutar, basándose en una lista blanca definida por el desarrollador.

Las políticas CSP se implementan a través de la cabecera HTTP Content-Security-Policy, o mediante etiquetas <meta> en el HTML. La estructura de estas políticas incluye directivas como default-src, script-src, style-src, entre otras, que determinan qué fuentes están permitidas para cada tipo de recurso. También se puede usar report-uri o Content-Security-Policy-Report-Only para registrar violaciones sin aplicar directamente la política, facilitando la depuración antes de una implementación completa.

CSP también contribuye a la protección contra ataques de sniffing, al exigir el uso exclusivo de HTTPS para todos los recursos, especialmente en entornos sensibles como la banca en línea. Esto se complementa con cabeceras como Strict-Transport-Security para reforzar la conexión segura. Además, se pueden definir políticas específicas por tipo de recurso, como permitir imágenes de cualquier origen, pero restringir scripts o formularios a ciertos dominios de confianza.

Finalmente, cuando se produce una violación de la política CSP, los navegadores pueden enviar informes en formato JSON a un endpoint definido, detallando la URL afectada, el recurso bloqueado, y la directiva incumplida. Esto permite a los desarrolladores monitorear e identificar vulnerabilidades en tiempo real. Sin embargo, hay que tener en cuenta ciertas inconsistencias de compatibilidad en navegadores como Safari, que pueden generar falsos positivos si no se combinan adecuadamente con otras políticas como Same-Origin.

Seguridad en autenticación, redirecciones y encabezados

La autenticación básica HTTP es un mecanismo simple de desafío y respuesta en el que el servidor solicita credenciales al cliente (usuario y contraseña), las cuales son enviadas mediante la cabecera Authorization, codificadas en Base64. Aunque es fácil de implementar, solo debe utilizarse sobre conexiones seguras (HTTPS), ya que de lo contrario, las credenciales pueden ser interceptadas. El servidor responde con un código 401 Unauthorized y la cabecera WWW-Authenticate, que inicia el proceso de autenticación. Los clientes modernos almacenan estas credenciales temporalmente para facilitar futuras solicitudes.

En cuanto a la implementación de cabeceras HTTP de seguridad, estas son fundamentales para proteger aplicaciones web frente a ataques como XSS, clickjacking y

sniffing de contenido. Entre las cabeceras más importantes están Strict-Transport-Security (HSTS), Content-Security-Policy (CSP), X-XSS-Protection, X-Frame-Options y X-Content-Type-Options. Se recomienda configurarlas adecuadamente para cada respuesta HTTP. Además, plataformas como Azure Application Gateway permiten añadir estas cabeceras automáticamente y aplicar reglas globales a todas las respuestas del servidor.

La reescritura de encabezados y URLs es otra herramienta clave dentro de la seguridad y el enrutamiento inteligente. Mediante reglas de reescritura, se pueden modificar encabezados, eliminar información confidencial (como datos sobre el servidor o sistema operativo) y personalizar URLs sin alterar lo que el usuario ve. Application Gateway también permite reescribir parámetros de consulta basándose en la estructura de la URL, facilitando la navegación amigable y el procesamiento interno eficiente.

Una parte importante de la configuración es la personalización de la Directiva de Seguridad de Contenido (CSP). Esta cabecera controla los orígenes desde donde se pueden cargar recursos como scripts, imágenes o estilos. Puede establecerse una política general (default-src) y luego ajustarse con directivas más específicas como img-src, script-src, etc. AD FS, por ejemplo, utiliza CSP personalizada para permitir JavaScript esencial durante el proceso de autenticación, aunque esto implica usar valores como 'unsafe-inline' y 'unsafe-eval', los cuales deben manejarse con precaución.

Finalmente, la compatibilidad de estas cabeceras de seguridad varía según el navegador, por lo que es vital conocer qué navegadores soportan qué cabeceras antes de implementarlas en producción. Además, AD FS permite definir encabezados personalizados, lo que otorga flexibilidad adicional. Todo esto destaca la importancia de realizar

configuraciones conscientes y mantenerse actualizado con las mejores prácticas para minimizar vulnerabilidades y garantizar una experiencia de usuario segura.

Conclusión

Los códigos de respuesta HTTP desempeñan un papel esencial en la comunicación entre clientes y servidores en el entorno web. Cada código, desde los 1xx informativos hasta los 5xx de error de servidor, proporcionan información crucial sobre el estado de una solicitud y guía el comportamiento tanto de los navegadores web como de las aplicaciones. Estos códigos son fundamentales para garantizar una experiencia de usuario fluida y una interacción eficiente entre los diversos componentes de la web.

Al comprender la variedad de códigos de respuesta y sus significados, los desarrolladores pueden diagnosticar y solucionar problemas más fácilmente, lo que contribuye a una mejor calidad y fiabilidad de las aplicaciones web. Desde los códigos 2xx que indican el éxito de una solicitud hasta los 4xx que señalan errores del cliente, cada código proporciona pistas importantes sobre cómo mejorar y optimizar el funcionamiento de una aplicación web. Además, la correcta gestión de los códigos de respuesta no solo mejora la experiencia del usuario, sino que también es vital para la seguridad y la protección de los datos. Los códigos 4xx, por ejemplo, pueden ayudar a prevenir ataques malintencionados al indicar problemas de autorización o acceso no autorizado, mientras que los códigos 5xx alertan sobre posibles problemas en el servidor que deben abordarse rápidamente para evitar interrupciones del servicio.

En conclusión, los códigos de respuesta HTTP son elementos fundamentales en el desarrollo y mantenimiento de aplicaciones web de alta calidad. Su correcta comprensión y gestión son esenciales para garantizar un funcionamiento óptimo, una experiencia de usuario satisfactoria y la seguridad de los datos en el vasto panorama digital de hoy en día.

Citas y referencias

Códigos de estado de respuesta HTTP - HTTP | MDN. (2022, 26 noviembre). MDN Web Docs. *https://developer.mozilla.org/es/docs/Web/HTTP/Status*

¿Qué son las respuestas HTTP? (s. f.). aulab.es. *https://aulab.es/articulos-guias-avanzadas/109/que-son-las-respuestas-http#:~:text=Las%20respuestas%20HTTP%20son%20mensajes,solicitud%20fue%20exitosa%20o%20no.*

Fielding, R. T., Nottingham, M., & Reschke, J. (2022, 1 junio). RFC 9110: HTTP Semantics. *https://www.rfc-editor.org/rfc/rfc9110.html#status.1xx*

RFC 4918: HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV). (s. f.). IETF Datatracker. *https://datatracker.ietf.org/doc/html/rfc4918#section-11*

Masinter, L. (1998). Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0). *https://doi.org/10.17487/rfc2324*

Nottingham, M., & Fielding, R. (2012). Additional HTTP status codes. https://doi.org/10.17487/rfc6585

Bray, T. (s. f.). *RFC7725*. IETF HTTP Working Group Specifications. https://httpwg.org/specs/rfc7725.html#n-451-unavailable-for-legal-reasons

Content Security Policy (CSP) - HTTP | MDN. (2025, 21 marzo). MDN Web Docs. https://developer.mozilla.org/es/docs/Web/HTTP/Guides/CSP

Cabeceras de seguridad HTTP: una capa extra de protección para las webs. (2022, 22 junio). https://www.sysadminok.es/blog/ciberseguridad/cabeceras-de-seguridad-http/#:~:text=HTTP%20Strict%20Transport%20Security%20(HSTS,la%20aparición%20de%20posibles%20peligros.

Greg-Lindsay. (s. f.). *Reescritura de los encabezados HTTP y direcciones URL con Azure Application Gateway*. Microsoft Learn. https://learn.microsoft.com/es-es/azure/application-gateway/rewrite-http-headers-url

CICS Transaction Server for z/OS. (s. f.). https://www.ibm.com/docs/es/cics-ts/6.x?topic=concepts-http-basic-authentication

Billmath. (s. f.). *Personalizar encabezados de respuesta de seguridad HTTP con AD FS*. Microsoft Learn. https://learn.microsoft.com/es-es/windows-server/identity/ad-fs/operations/customize-http-security-headers-ad-fs