



Instituto tecnológico superior de jerez

Ingeniería en sistemas computacionales

Programación Web

Mapa conceptual y cuestionario

HTTP

Docente: M.T.I. Salvador Acevedo Sandoval

Alumna: Viviana Michel Chávez Juárez

Numero de control: 14070013

Fecha: 06/09/2019



1. **¿Qué es el protocolo HTTP?y para qué sirve?**

El http (del inglés *HyperText Transfer Protocol* o Protocolo de Transferencia de Hiper Textos) es el **protocolo de transmisión de información de la World Wide Web**, es decir, el código que se establece para que el computador solicitante y el que contiene la información solicitada puedan “hablar” un mismo idioma a la hora de transmitir información por la red.

El http, como se ha dicho, es un lenguaje que **media entre las peticiones del cliente y las respuestas del servidor en la Internet**, para permitir una comunicación fluida y en un mismo “lenguaje”. Este protocolo establece las pautas a seguir, los métodos de petición (llamados “verbos”) y cuenta con cierta flexibilidad para incorporar nuevas peticiones y funcionalidades, en especial a medida que se avanza en sus versiones.

Considerando que la Internet es poco más que una compleja red de intercambio de información entre computadores a distancia, este tipo de herramientas digitales son clave en establecer las bases para ordenar y facilitar la transmisión de la información.

2. **¿Cuál es la versión mas actual de dicho protocolo?**

HTTP/2

optimización

3. **¿Qué información tienen un encabezado HTTP?**

Una cabecera de petición esta compuesta por su nombre (no sensible a las mayusculas) seguido de dos puntos ':', y a continuación su valor (sin saltos de línea). Los espacios en blanco a la izquierda del valor son ignorados

4. **¿Qué información muestra un CODIGO DE ESTADO en una respuesta HTTP?**

Los códigos de estado de respuesta HTTP indican si se ha completado satisfactoriamente una solicitud HTTP específica. Las respuestas se agrupan en cinco clases: respuestas informativas, respuestas satisfactorias, redirecciones, errores de los clientes y errores de los servidores

Códigos de estado 1XX

Informan al navegador de algunas acciones que se van a realizar:

- **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 (WebDAV)*), 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** (*Checkpoint*), se va a reanudar una petición POST o PUT que fue abortada previamente.

Códigos de estado 2XX

Indican que la petición del navegador se ha recibido, procesado y respondido correctamente:

- **200** (*Ok*), la petición del navegador se ha completado con éxito.
- **201** (*Created*), la petición del navegador se ha completado con éxito y como resultado, se ha creado un nuevo recurso (la respuesta incluye la URI de ese recurso).
- **202** (*Accepted*), la petición del navegador se ha aceptado y se está procesando en estos momentos, por lo que todavía no hay una respuesta (se utiliza por ejemplo cuando un proceso realiza una petición muy compleja a un servidor y no quiere estar horas esperando la respuesta).
- **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 (la respuesta sí que puede incluir información en sus cabeceras HTTP).
- **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 (este código es útil por ejemplo para páginas con formularios cuyo contenido debe borrarse después de que el usuario lo envíe).
- **206** (*Partial Content*), La respuesta de esta petición sólo tiene parte de los contenidos, tal y como lo solicitó el propio navegador (se utiliza por ejemplo cuando se descarga un archivo muy grande en varias partes para acelerar la descarga).
- **207** (*Multi-Status (WebDAV)*), la respuesta consiste en un archivo XML que contiene en su interior varias respuestas diferentes (el número depende de las peticiones realizadas previamente por el navegador).
- **208** (*Already Reported (WebDAV)*), el listado de elementos DAV ya se notificó previamente, por lo que no se van a volver a listar.



Códigos de estado 3XX

Indican que el navegador debe realizar alguna acción adicional para que la petición se complete (como por ejemplo redirigirse a otra página):

- **300** (*Multiple Choices*), existe más de una variante para el recurso solicitado por el navegador (por ejemplo si la petición se corresponde con más de un archivo).
- **301** (*Moved Permanently*), el recurso solicitado por el navegador se encuentra en otro lugar y este cambio es permanente. El navegador es redirigido automáticamente a la nueva localización de ese recurso (este código es muy importante para tareas relacionadas con el SEO de los sitios web).
- **302** (*Moved Temporarily*), el recurso solicitado por el navegador se encuentra en otro lugar, aunque sólo por tiempo limitado. El navegador es redirigido automáticamente a la nueva localización de ese recurso.
- **303** (*See Other*), el recurso solicitado por el navegador se encuentra en otro lugar. El servidor no redirige automáticamente al navegador, pero le indica la nueva URI en la que se puede obtener el recurso.
- **304** (*Not Modified*), cuando el navegador pregunta si un recurso ha cambiado desde la última vez que se solicitó, el servidor responde con este código cuando el recurso no ha cambiado.
- **305** (*Use Proxy*), el recurso solicitado por el navegador debe obtenerse a través del proxy cuya dirección se indica en la cabecera Location de esta misma respuesta.
- **306** (*Switch Proxy*), este código se utilizaba en las versiones antiguas de HTTP pero ya no se usa (aunque está reservado para usos futuros).
- **307** (*Temporary Redirect*), el recurso solicitado por el navegador se puede obtener en otro lugar, pero sólo para esta petición. Las próximas peticiones pueden seguir utilizando la localización original del recurso.
- **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 (así por ejemplo, si envías un formulario a un recurso que ha cambiado de lugar, todo seguirá funcionando bien).

Códigos de estado 4XX

Indican que se ha producido un error cuyo responsable es el navegador:

- **400** (*Bad Request*), el servidor no es capaz de entender la petición del navegador porque su sintaxis no es correcta.



- **401 (*Unauthorized*)**, el recurso solicitado por el navegador requiere de autenticación. La respuesta incluye una cabecera de tipo WWW-Authenticate para que el navegador pueda iniciar el proceso de autenticación.
- **402 (*Payment Required*)**, este código está reservado para usos futuros.
- **403 (*Forbidden*)**, la petición del navegador es correcta, pero el servidor no puede responder con el recurso solicitado porque se ha denegado el acceso.
- **404 (*Not Found*)**, el servidor no puede encontrar el recurso solicitado por el navegador y no es posible determinar si esta ausencia es temporal o permanente.
- **405 (*Method Not Allowed*)**, el navegador ha utilizado un método (GET, POST, etc.) no permitido por el servidor para obtener ese recurso.
- **406 (*Not Acceptable*)**, el recurso solicitado tiene un formato que en teoría no es aceptable por el navegador, según los valores que ha indicado en la cabecera Accept de la petición.
- **407 (*Proxy Authentication Required*)**, es muy similar al código 401, pero en este caso, el navegador debe autenticarse primero con un proxy.
- **408 (*Request Timeout*)**, el navegador ha tardado demasiado tiempo en realizar su petición y el servidor ya no espera esa petición. No obstante, el navegador puede realizar nuevas peticiones cuando quiera.
- **409 (*Conflict*)**, la petición del navegador no se ha podido completar porque se ha producido un conflicto con el recurso solicitado. El caso más habitual es el de las peticiones de tipo PUT que intentan modificar un recurso que a su vez ya ha sido modificado por otro lado.
- **410 (*Gone*)**, no es posible encontrar el recurso solicitado por el navegador y esta ausencia se considera permanente. Si existe alguna posibilidad de que el recurso vuelva a estar disponible, se debe utilizar el 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 (esta condición se produce en muy raras ocasiones y casi siempre porque el navegador envía como GET una petición que debería ser POST).
- **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 navegador ha solicitado una porción inexistente de un recurso. Este error se produce cuando el navegador descarga por partes un archivo muy grande y calcula mal el tamaño de algún trozo.
- **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.
- **422** (*Unprocessable Entity (WebDAV)*), la petición del navegador tiene el formato correcto, pero sus contenidos tienen algún error semántico que impide al servidor responder.
- **423** (*Locked (WebDAV)*), el recurso solicitado por el navegador no se puede entregar porque está bloqueado.
- **424** (*Failed Dependency (WebDAV)*), la petición del navegador ha fallado debido al error de alguna petición anterior (por ejemplo una petición con el método PROPPATCH).
- **426** (*Upgrade Required*), el navegador debe cambiar a un protocolo diferente para realizar las peticiones (por ejemplo TLS/1.0).
- **428** (*Precondition Required*), el servidor requiere que la petición del navegador sea condicional (este tipo de peticiones evitan los problemas producidos al modificar con PUT un recurso que ha sido modificado por otra parte).
- **429** (*Too Many Requests*), el navegador ha realizado demasiadas peticiones en un determinado período de tiempo (se utiliza sobre todo para forzar los límites de consumo de recursos de las APIs).
- **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.

Códigos de estado 5XX

Indican que se ha producido un error cuyo responsable es el servidor:

- **500** (*Internal Server Error*), la solicitud del navegador no se ha podido completar porque se ha producido un error inesperado en el servidor.
- **501** (*Not Implemented*), el servidor no soporta alguna funcionalidad necesaria para responder a la solicitud del navegador (como por ejemplo el método utilizado para la petición).
- **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 (WebDAV)*), el servidor no puede crear o modificar el recurso solicitado porque no hay suficiente espacio de almacenamiento libre.
 - **508** (*Loop Detected (WebDAV)*), la petición no se puede procesar porque el servidor ha encontrado un bucle infinito al intentar procesarla.
 - **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 (se utiliza por ejemplo con los portales cautivos que te obligan a autenticarte antes de empezar a navegar).
5. **¿Cuáles son los METODOS de comunicación de este protocolo?**
- Abre una conexión TCP: la conexión TCP se usará para hacer una petición, o varias, y recibir la respuesta. El cliente puede abrir una conexión nueva, reusar una existente, o abrir varias a la vez hacia el servidor.
- Hacer una petición HTTP: Los mensajes HTTP (previos a HTTP/2) son legibles en texto plano. A partir de la versión del protocolo HTTP/2, los mensajes se encapsulan en franjas, haciendo que no sean directamente interpretables, aunque el principio de operación es el mismo.
- Leer la respuesta enviada por el servidor
- Cierre o reuso de la conexión para futuras peticiones
6. **¿Qué es el protocolo HTTPS y para qué sirve?**
- HTTPS (Protocolo **Seguro** de Transferencia Hipertexto) funciona desde el puerto 443 y utiliza un cifrado basado en SSL/TLS con el fin de crear un canal cifrado entre el cliente y el servidor. SSL (Secure Sockets Layer) y TLS (Transmission Layer Security) son dos protocolos utilizados para enviar paquetes cifrados a través de Internet. Se pueden utilizar para más de un protocolo, no sólo con HTTP. **HTTP + SSL/TLS = HTTPS.**



Inicialmente, las conexiones HTTPS utilizaban el protocolo de seguridad SSL (Secure Socket Layer) para proteger las conexiones de los usuarios al utilizar plataformas muy concretas, como PayPal o webs bancarias. Hoy en día, este protocolo ha cambiado SSL por TLS, (Transport Layer Security), protocolo que nos brinda mayor seguridad y mayor privacidad en nuestras conexiones.

Esta seguridad es, sin duda, la principal ventaja de las conexiones seguras HTTPS. Mientras que cualquier usuario puede interceptar el contenido de los paquetes HTTP, e incluso modificarlo en ataque MITM, las conexiones HTTPS no pueden ser espiadas, por lo que nadie sabrá qué tipo de información estamos intercambiando con el servidor. Además, estas conexiones no pueden ser modificadas en ataques MITM, por lo que tendremos la certeza de que los datos que intercambiamos con el servidor son legítimos, rechazando cualquier dato que no coincida con la firma del certificado de la web.

7. **Códigos de error de una respuesta HTTP**

Códigos de estado 4XX

Indican que se ha producido un error cuyo responsable es el navegador:

- **400** (*Bad Request*), el servidor no es capaz de entender la petición del navegador porque su sintaxis no es correcta.
- **401** (*Unauthorized*), el recurso solicitado por el navegador requiere de autenticación. La respuesta incluye una cabecera de tipo WWW-Authenticate para que el navegador pueda iniciar el proceso de autenticación.
- **402** (*Payment Required*), este código está reservado para usos futuros.
- **403** (*Forbidden*), la petición del navegador es correcta, pero el servidor no puede responder con el recurso solicitado porque se ha denegado el acceso.
- **404** (*Not Found*), el servidor no puede encontrar el recurso solicitado por el navegador y no es posible determinar si esta ausencia es temporal o permanente.
- **405** (*Method Not Allowed*), el navegador ha utilizado un método (GET, POST, etc.) no permitido por el servidor para obtener ese recurso.
- **406** (*Not Acceptable*), el recurso solicitado tiene un formato que en teoría no es aceptable por el navegador, según los valores que ha indicado en la cabecera Accept de la petición.
- **407** (*Proxy Authentication Required*), es muy similar al código 401, pero en este caso, el navegador debe autenticarse primero con un proxy.



- **408** (*Request Timeout*), el navegador ha tardado demasiado tiempo en realizar su petición y el servidor ya no espera esa petición. No obstante, el navegador puede realizar nuevas peticiones cuando quiera.
- **409** (*Conflict*), la petición del navegador no se ha podido completar porque se ha producido un conflicto con el recurso solicitado. El caso más habitual es el de las peticiones de tipo PUT que intentan modificar un recurso que a su vez ya ha sido modificado por otro lado.
- **410** (*Gone*), no es posible encontrar el recurso solicitado por el navegador y esta ausencia se considera permanente. Si existe alguna posibilidad de que el recurso vuelva a estar disponible, se debe utilizar el 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 (esta condición se produce en muy raras ocasiones y casi siempre porque el navegador envía como GET una petición que debería ser POST).
- **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 navegador ha solicitado una porción inexistente de un recurso. Este error se produce cuando el navegador descarga por partes un archivo muy grande y calcula mal el tamaño de algún trozo.
- **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.
- **422** (*Unprocessable Entity (WebDAV)*), la petición del navegador tiene el formato correcto, pero sus contenidos tienen algún error semántico que impide al servidor responder.
- **423** (*Locked (WebDAV)*), el recurso solicitado por el navegador no se puede entregar porque está bloqueado.
- **424** (*Failed Dependency (WebDAV)*), la petición del navegador ha fallado debido al error de alguna petición anterior (por ejemplo una petición con el método PROPPATCH).
- **426** (*Upgrade Required*), el navegador debe cambiar a un protocolo diferente para realizar las peticiones (por ejemplo TLS/1.0).
- **428** (*Precondition Required*), el servidor requiere que la petición del navegador sea condicional (este tipo de peticiones evitan los problemas



producidos al modificar con PUT un recurso que ha sido modificado por otra parte).

- **429** (*Too Many Requests*), el navegador ha realizado demasiadas peticiones en un determinado período de tiempo (se utiliza sobre todo para forzar los límites de consumo de recursos de las APIs).
- **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.

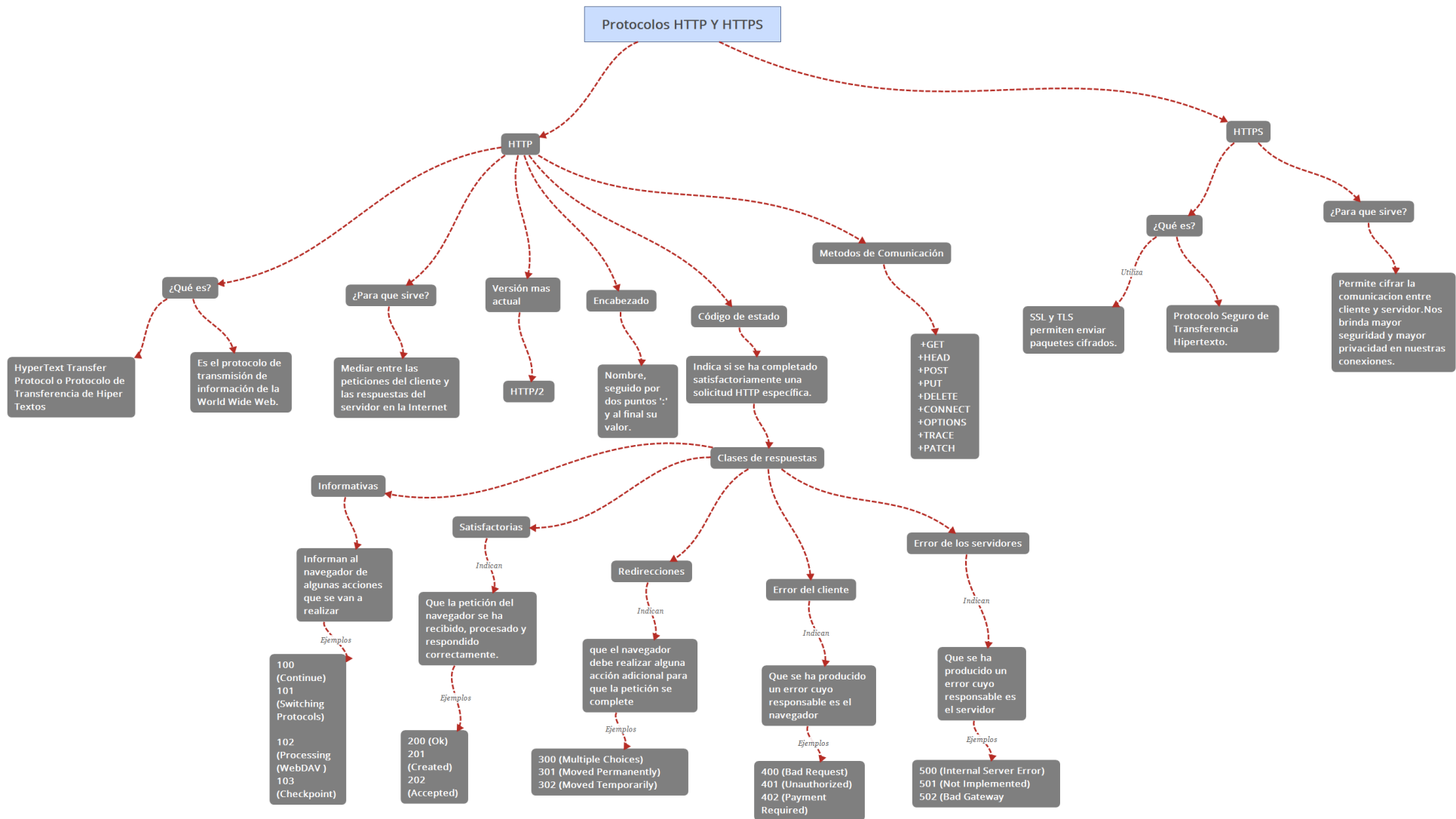
Códigos de estado 5XX

Indican que se ha producido un error cuyo responsable es el servidor:

- **500** (*Internal Server Error*), la solicitud del navegador no se ha podido completar porque se ha producido un error inesperado en el servidor.
- **501** (*Not Implemented*), el servidor no soporta alguna funcionalidad necesaria para responder a la solicitud del navegador (como por ejemplo el método utilizado para la petición).
- **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 (WebDAV)*), el servidor no puede crear o modificar el recurso solicitado porque no hay suficiente espacio de almacenamiento libre.
- **508** (*Loop Detected (WebDAV)*), la petición no se puede procesar porque el servidor ha encontrado un bucle infinito al intentar procesarla.
- **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 (se utiliza por ejemplo con los portales cautivos que te obligan a autenticarte antes de empezar a navegar).





Bibliografía

mdnwebdocs-bot, S. C. (23 de marzo de 2019). *MDN Web Documento*. Obtenido de https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP

Raffino, M. E. (29 de Agosto de 2019). *Concepto.de*. Obtenido de <https://concepto.de/http/>

<https://developer.mozilla.org/es/docs/Web/HTTP/Status>

<https://uniwebsidad.com/tutoriales/los-codigos-de-estado-de-http>

<https://developer.mozilla.org/es/docs/Web/HTTP/Overview>