

# PROTOCOLO HTTP

LUIS LENNIN

VILLARREAL CASTRO

UNIVERSIDAD DE LA  
SIERRA SUR

14/MAR/2022

**Introducción:**

Para dar inicio con este tema de protocolos de HTTP, debemos conocer que un protocolo no es más que un conjunto de reglas, en la red los protocolos son considerados como estándares y políticas formales que están conformados por restricciones, procedimientos y formatos que definen el intercambio de paquetes de información para lograr la comunicación entre servidores o dispositivos que se conectan a través de una red. Así que podríamos decir que un protocolo HTTP brinda al usuario una forma de interactuar con los recursos que brinda la web, como lo son los archivos HTML, mediante la transmisión de hipertexto entre el cliente y el servidor. Por lo tanto en este documento vamos a describir los métodos de petición, códigos de respuesta, cabeceras y algún ejemplo de diálogo HTTP.

## GENERALIDADES DEL PROTOCOLO HTTP:

### ¿QUÉ ES HTTP?

HTTP es un protocolo de transferencia de hipertexto, por decir, es un protocolo de aplicación para sistemas de información hipermedia distribuidos y colaborativos que permite a los usuarios comunicar datos en la World Wide Web, se inventó junto con HTML para crear el primer navegador web interactivo basado en texto "la original World Wide Web ". Hoy en día, el protocolo sigue siendo uno de los principales medios de uso de Internet.

Como protocolo de solicitud-respuesta, HTTP brinda a los usuarios una forma de interactuar con recursos web, como archivos HTML, mediante la transmisión de mensajes de hipertexto entre clientes y servidores. Los clientes HTTP generalmente usan conexiones de Protocolo de control de transmisión (TCP) para comunicarse con los servidores.

### ¿MENSAJE DEL PROTOCOLO HTTP?

En HTTP los mensajes son los medios por los cuales se intercambian datos entre servidores y clientes. Hay dos tipos de mensajes:

- Peticiones.
- Peticiones enviadas por el cliente al servidor.
- Para pedir el inicio de una acción.
- Respuestas (son la respuesta del servidor).

Los mensajes HTTP están compuestos de texto, codificado en ASCII, y pueden comprender múltiples líneas. En HTTP/1.1, y versiones previas del protocolo, estos mensajes eran enviados de forma abierta a través de la conexión. En HTTP/2.0 los mensajes, que anteriormente eran legibles directamente, se conforman mediante tramas binarias codificadas para aumentar la optimización y rendimiento de la transmisión.

Los desarrolladores de páginas Web, o administradores de sitios Web, desarrolladores... raramente codifican directamente estos mensajes HTTP. Normalmente especifican estos mensajes HTTP, mediante archivos de configuración (para proxies, y servidores), APIs (para navegadores) y otros medios.

### ¿MÉTODOS DE PETICIÓN?

MDN contributors(12 Mar 2022)"Se define como un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado segun los protocolos de HTTP, unque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados HTTP verbs. Cada uno de ellos implementan una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos: ej. un request method puede ser safe, idempotent, o cacheable"(p.2).

GET

El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

HEAD

El método HEAD pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

POST

El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

PUT

El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

DELETE

El método DELETE borra un recurso en específico.

CONNECT

El método CONNECT establece un túnel hacia el servidor identificado por el recurso.

OPTIONS

El método OPTIONS es utilizado para describir las opciones de comunicación para el recurso de destino. TRACE

El método TRACE realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.

PATCH

El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.

### ¿CÓDIGOS DE RESPUESTA?

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 (100–199)

Respuestas satisfactorias (200–299)

Redirecciones (300–399)

Errores de los clientes (400–499)

y errores de los servidores (500–599).

Los códigos de estado se definen en la sección 10 de RFC 2616. Puedes obtener las especificaciones actualizadas en RFC 7231.

Respuestas informativas: 100 Continue (indica que todo está bien y que el cliente debe continuar con la solicitud ).

101 Switching Protocol (envía en respuesta, indica que el servidor acepta el cambio de protocolo).

102 Processing (WebDAV (en-US)(indica que el servidor ha recibido la solicitud, aún se encuentra procesandola).

103 Early Hints (en-US)( usado con el encabezado Link,el usuario empieza a pre-cargar recursos mientras el servidor prepara una respuesta).

Respuestas satisfactorias:

GET (El recurso se ha obtenido y se transmite en el cuerpo del mensaje).

HEAD (Los encabezados de entidad están en el cuerpo del mensaje).

PUT o POST: (describe el resultado de la acción se transmite en el cuerpo del mensaje).

TRACE: (contiene el mensaje de solicitud recibido por el servidor).

200 OK (La solicitud ha tenido éxito).

201 Created (se ha creado un nuevo recurso como resultado de ello).

202 Accepted(La solicitud se ha recibido, pero aún no se ha actuado).

203 Non-Authoritative Information (La petición se ha completado con éxito, pero su contenido no se ha obtenido de la fuente originalmente solicitada).

204 No Content (en-US)(La petición se ha completado con éxito pero su respuesta no tiene ningún contenido).

205 Reset Content (en-US)(La petición se ha completado con éxito, pero su respuesta no tiene contenidos y además).

206 Partial Content(La petición servirá parcialmente el contenido solicitado).

207 Multi-Status (WebDAV (en-US)(transmite información sobre varios recursos en situaciones en las que varios códigos de estado podrían ser apropiados).

208 Multi-Status (WebDAV (en-US))(El listado de elementos DAV ya se notificó previamente, por lo que no se vuelven a listar).

302 Found(la URI solicitada ha sido cambiado temporalmente).

303 See Other (en-US)(El servidor envía una respuesta para dirigir al cliente a otra dirección usando una petición GET).

304 Not Modified(Esta es usada para propósitos de caché).

305 Use Proxy (Fue definida en una versión previa de la especificación del protocolo HTTP para indicar que una respuesta solicitada debe ser accedida desde un proxy).

306 unused(Este código de respuesta ya no es usado más).

307 Temporary Redirect (en-US)(El servidor envía esta respuesta para dirigir al cliente a obtener el recurso solicitado a otra URI).

308 Permanent Redirect (en-US)(Significa que el recurso ahora se encuentra permanentemente en otra URI).

400 Bad Request(el servidor no pudo interpretar la solicitud dada una sintaxis inválida).

401 Unauthorized(Es necesario autenticar para obtener la respuesta solicitada).

402 Payment Required(Este código de respuesta está reservado para futuros usos).

- 403 Forbidden (El cliente no posee los permisos necesarios para cierto contenido).
- 404 Not Found (El servidor no pudo encontrar el contenido solicitado).
- 405 Method Not Allowed (en-US)(El método solicitado es conocido por el servidor pero ha sido deshabilitado y no puede ser utilizado).
- 406 Not Acceptable (en-US)(no encuentra ningún contenido seguido por la criteria dada por el usuario).
- 407 Proxy Authentication Required (en-US)(la autenticación debe estar hecha a partir de un proxy).
- 408 Request Timeout(Esta respuesta es enviada en una conexión inactiva).
- 409 Conflict (en-US) (puede ser enviada cuando una petición tiene conflicto con el estado actual del servidor).
- 410 Gone (en-US)(el contenido solicitado ha sido borrado del servidor).
- 411 Length Required (en-US)(El servidor rechaza la petición porque el campo de encabezado Content-Length no esta definido y el servidor lo requiere).
- 412 Precondition Failed (en-US)(El cliente ha indicado pre-condiciones en sus encabezados la cual el servidor no cumple).
- 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).
- 414 URI Too Long (en-US)(La URI solicitada por el cliente es más larga de lo que el servidor está dispuesto a interpretar).
- 415 Unsupported Media Type (en-US)(El formato multimedia de los datos solicitados no está soportado por el servidor, por lo cual el servidor rechaza la solicitud).
- 416 Requested Range Not Satisfiable (en-US)(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).
- 417 Expectation Failed (en-US)(Significa que la expectativa indicada por el campo de encabezado Expect solicitada no puede ser cumplida por el servidor).
- 418 I'm a teapot(El servidor se rehúsa a intentar hacer café con una tetera).
- 421 Misdirected Request(La petición fue dirigida a un servidor que no es capaz de producir una respuesta).
- 422 Unprocessable Entity (en-US) (WebDAV (en-US))(La petición estaba bien formada pero no se pudo seguir debido a errores de semántica).
- 423 Locked (WebDAV (en-US))(El recurso que está siendo accedido está bloqueado).
- 424 Failed Dependency (WebDAV (en-US))(La petición falló debido a una falla de una petición previa).
- 426 Upgrade Required (en-US)(El servidor se rehúsa a aplicar la solicitud usando el protocolo actual pero puede estar dispuesto a hacerlo después que el cliente se actualice a un protocolo diferente).
- 428 Precondition Required (en-US)(El servidor origen requiere que la solicitud sea condicional).
- 429 Too Many Requests (en-US)(El usuario ha enviado demasiadas solicitudes en un periodo de tiempo dado).
- 431 Request Header Fields Too Large (en-US)(El servidor no está dispuesto a procesar la solicitud porque los campos de encabezado son demasiado largos).
- 451 Unavailable For Legal Reasons (en-US)(El usuario solicita un recurso ilegal, como alguna página web censurada por algún gobierno).

#### Errores de servidor

- 500 Internal Server Error(El servidor ha encontrado una situación que no sabe cómo manejarla).
- 501 Not Implemented (en-US)(El método solicitado no está soportado por el servidor y no puede ser manejado).
- 502 Bad Gateway(respuesta de error que significa que el servidor, mientras trabaja como una puerta de enlace para obtener una respuesta necesaria para manejar la petición, obtuvo una respuesta inválida).

503 Service Unavailable(El servidor no está listo para manejar la petición).

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).

505 HTTP Version Not Supported(La versión de HTTP usada en la petición no está soportada por el servidor). 506 Variant Also Negotiates (en-US)(El servidor tiene un error de configuración interna: negociación de contenido transparente para la petición resulta en una referencia circular).

507 Insufficient Storage (en-US)(El servidor tiene un error de configuración interna: la variable de recurso escogida está configurada para acoplar la negociación de contenido transparente misma, y no es por lo tanto un punto final adecuado para el proceso de negociación).

508 Loop Detected (en-US) (WebDAV (en-US))(El servidor detectó un ciclo infinito mientras procesaba la solicitud).

510 Not Extended (en-US)(Extensiones adicionales para la solicitud son requeridas para que el servidor las cumpla).

511 Network Authentication Required (en-US)(El código de estado 511 indica que el cliente necesita autenticar para obtener acceso a la red).

## CABECERAS

Los campos de encabezado HTTP brindan información requerida sobre la solicitud o respuesta, o sobre el objeto enviado en el cuerpo del mensaje. Hay cuatro tipos de encabezados de mensajes HTTP:

Encabezado general: estos campos de encabezado tienen aplicabilidad general tanto para mensajes de solicitud como de respuesta.

Encabezado de solicitud del cliente: estos campos de encabezado tienen aplicabilidad solo para mensajes de solicitud.

Encabezado de respuesta del servidor: estos campos de encabezado tienen aplicabilidad solo para mensajes de respuesta.

Encabezado de entidad: estos campos de encabezado definen la metainformación sobre el cuerpo de la entidad o, si no hay ningún cuerpo presente, sobre el recurso identificado por la solicitud.

## EJEMPLO DE DIÁLOGO HTTP

El elemento HTML <dialog>representa una caja de diálogo u otro componente interactivo, como inspector o ventana.

```
<dialog open>
  <p>Greetings, one and all!</p>
</dialog>
```

**Bibliografía:**

HTTP Protocol - Definition How it Works - ExtraHop | ExtraHop. (2022). Retrieved 15 March 2022, from <https://www.extrahop.com/resources/protocols/http/>

HTTP - Header Fields. (2022). Retrieved 15 March 2022, from <https://www.tutorialspoint.com/http/httpheaderfields.htm>

Códigos de estado de respuesta HTTP - HTTP | MDN. (2022). Retrieved 15 March 2022, from <https://developer.mozilla.org/es/docs/Web/HTTP/Status>

Métodos de petición HTTP - HTTP | MDN. (2022). Retrieved 15 March 2022, from <https://developer.mozilla.org/es/docs/Web/HTTP/Methods>

<dialog>- HTML: Lenguaje de etiquetas de hipertexto | MDN. (2022). Retrieved 15 March 2022, from <https://developer.mozilla.org/es/docs/Web/HTML/Element/dialog>