

Generalidades del protocolo HTTP

Cristian Adair Ramirez Rodriguez*

14 de Marzo del 2022

1 Introducción

Hablábamos antes de que en Internet tenemos una máquina que envía datos a otra mediante una red que sigue el protocolo IP. Y también sabemos que se pueden utilizar varios protocolos a la vez, como IP para la entrega de los datos y XMPP para chatear.

Como bien sabemos el protocolo de transferencia de hipertexto su objetivo general es la transmisión de datos y archivos entre los navegadores y los servidores web. Este proceso comienza cuando un cliente en este caso el navegador solicita los archivos a los servidores web, todo y cada uno de los procesos que se llevara acabo se verán en este temario, en principio analizar los generalidades del protocolo así como el mensaje, el método de petición, código de respuesta, cabeceras y ejemplos de dialogo del HTTP.

HTTP fue ideado para intercambiar hipertexto (texto que permite enlazar otros textos mediante enlaces, o sea, que hipertexto es el nombre rimbombante que reciben las páginas web). Normalmente las páginas web están alojadas en un servidor web. Un servidor web es un ordenador que está encendido todo el día esperando que alguien le envíe un mensaje pidiendo información

2 Generalidades del protocolo HTTP

2.1 Mensajes

Los mensajes HTTP, son los medios por los cuales se intercambian datos entre servidores y clientes. Hay dos tipos de mensajes: peticiones, enviadas por el cliente al servidor, para pedir el inicio de una acción; y respuestas, que son la respuesta del servidor”.(Mensajes HTTP - HTTP — MDN, 2022) Al igual que la interacción entre los humanos, siempre hay un emisor y un remitente, y de igual manera cuando se hace la retroalimentación cambian de situación el remitente pasa a ser el emisor y el emisor pasa a ser el remitente, de igual manera en las redes, existe el servidor y el cliente, que lleva acabo este intercambio de datos.

Estos datos o mensajes, son por lo tanto los textos, que son las lineas de código. Estas lineas forman el mensaje que sera enviado, mediante archivos de configuración y otros medios.

2.2 Métodos de petición

Los métodos de petición son todas aquellas peticiones que se le hace a los recursos del HTTP. Al igual que cualquier otro lenguaje este lleva un orden en la cual se van a ir ejecutando y haciendo diferentes tipos de acciones. Algunos de esos métodos son:

GET: estos métodos como su nombre lo indica solo obtienen o recuperan los datos. Con una particular diferencia, que esta espera una respuesta.

HEAD: este método tiene el mismo comportamiento que el método get, solo que esta no espera una respuesta.

PUT: este método de manera simple elimina las peticiones antiguas y agrega las nuevas peticiones.

POST: este método envía una entidad a un recurso en especifico, cabe mencionar que al hacer uso de este método por lo regular causa cambios en el estado del servidor.

DELETE: Como su nombre lo indica este método borra un recurso en especifico.

CONNECT: este método funciona como un canal entre el servidor y el recurso.

OPTIONS: con este método se identifican las opciones de comunicación para un recurso destino.

TRACE: este método realiza las correcciones de bucles o de errores en el transcurso del mensaje.

PATCH: este metodo es utilizado para hacer algunas modificaciones de manera parcial a los recursos que se encuentran disponibles.

2.3 Código de respuesta

Cabe mencionar el ejemplo anterior, entre la comunicación entre los humanos, cuando el receptor recibe el

mensaje este le envi  un nuevo mensaje al emisor, de igual manera en HTTP a ese nuevo mensaje se le conoce como respuesta, o c digo de respuesta la cual en HTTP indica si una solicitud ha sido completada correctamente. Esa respuestas se clasifican en las siguientes clases:

Respuesta informativa (100-199),
Respuesta satisfactoria (200-299),
Redirecciones (300-399),
Errores de los clientes (400-499),
Errores de los servicios (500- 599).

2.4 Cabeceras

Las Cabeceras HTTP son los par metros que se env an en una petici n o respuesta HTTP al cliente o al servidor para proporcionar informaci n esencial sobre la transacci n en curso. Estas cabeceras proporcionan informaci n mediante la sintaxis 'Cabecera: Valor' y son enviadas autom ticamente por el navegador o el servidor web. ("Anexo:Cabeceras HTTP - Wikipedia, la enciclopedia libre", 2022). Es decir al visitar una pagina web, cuando se hace una solicitud al servidor en ese instante, el navegador a la cual se esta accediendo hace la solicitud. En el intercambio de las solicitudes y respuestas ah  es en donde se presenta la cabecera presentando informaci n mas resumida y detallada acerca de la solicitud.

2.5 Ejemplos de di logo

Estos se establecen al momento de obtener recursos a trav s de un URL. De manera de ejemplo se tiene la siguiente estructura: Se abre una conexi n en el puerto 80 del host www.example.com. El puerto 80 es el puerto predefinido para HTTP. Si se quisiera utilizar el puerto XXXX habr a que codificarlo en la URL de la forma http://www.example.com:XXXX/index.html.

Se env a un mensaje en el estilo siguiente:

```
GET /index.html HTTP/1.1
Host: www.example.com
Referer: www.google.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Connection: keep-alive
[L nea en blanco]
```

La respuesta del servidor est  formada por encabezados seguidos del recurso solicitado, en el caso de una p gina web:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221
<html lang="eo">
<head>
<meta charset="utf-8">
<title>T tulo del sitio</title>
</head>
<body>
<h1>P gina principal de tuHost</h1>
(Contenido)
.
.
```

```
</body>
</html>
```

3 Conclusión

Gracias al uso de este protocolo podemos tener una mejor manera de comunicarnos entre los usuarios del internet, tomando en cuenta todo los paso que se toman desde la petición que manda el cliente con las normas y requisitos hasta las respuestas que da el servidor.

Los paso que se deben de tomar para enviar archivos y aue esto lleguen a su detno me parecio muy laborioso pero efectivo ya que con eso no esta el riesgo de perdes información.

References

- [1] Developer.mozilla.org. 2022. Mensajes HTTP - HTTP — MDN. [online] Available at: <https://developer.mozilla.org/es/docs/Web/HTTP/Messages>; [Accessed 14 March 2022].
- [2] Developer.mozilla.org. 2022. Métodos de petición HTTP - HTTP — MDN. [online] Available at: <https://developer.mozilla.org/es/docs/Web/HTTP/Methods>; [Accessed 14 March 2022].
- [3] Developer.mozilla.org. 2022. Códigos de estado de respuesta HTTP - HTTP — MDN. [online] Available at: <https://developer.mozilla.org/es/docs/Web/HTTP/Status>; [Accessed 14 March 2022].
- [4] Anexo:Cabeceras HTTP - Wikipedia, la enciclopedia libre. (2022). Retrieved 14 March 2022, from https://es.wikipedia.org/wiki/Anexo:Cabeceras_HTTP.
- [5] Protocolo de transferencia de hipertexto - Wikipedia, la enciclopedia libre. (2022). Retrieved 14 March 2022, from https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto *Ejemplo de la C3. A1 logo HTTP*