Generalidades del Protocolo *HTTP*.

Guillermo Daniel Cruz Ortega
guillermo.daniel.cruz.ortega@gmail.com
Universidad de la Sierra Sur.

2022/03/14

1 Introducción.

En este ensayo se buscara indagar y enetender que es el protocolo *HTTP*, las partes que lo componen, y el como funciona, y un poco de su historia a lo largo de los años de su existencia. *HTTP* es un protocolo de comunicación implementado desde la epoca de los años noventa, con el fin de transferir informacion en la www (*Word Wide Web*). *HTTP* es la abreviatura de protocolo de transferencia de hipertextos, la cual es utiliza en gran mayoria por todos los sitios de internet de todo el mundo. *HTTP* es de lo que mas pasa desapercibido al momento de navegar en la internet, siento que solo ponemos en nombre del sitio a ocupar y automaticamente hace el proceso de redireccionamiento, y todo esto posible en unos cuantos segundos (dependiendo de la velocidad dell internet) y mientras que el protocolo HTTP se encarga de hacer todo esto posible, en cualquier motor de busqueda al dar *click* en la *url* del sitio se puede ver el protocolo *HTTP* al incio de de cada *url* de culquier sitio.

2 ¿Que es *HTTP*?

"Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. El cliente (se le suele llamar "agente de usuario", del inglés *user agent*) realiza una petición enviando un mensaje, con cierto formato al servidor. El servidor (al que es común llamarle servidor *web*) le envía un mensaje de respuesta. Ejemplos de cliente son los navegadores web y las arañas web (también conocidas por su término inglés, *webcrawlers*)."(*wikipedia*, 2022)
HTTP igual es un protocolo sin estado, con esto se refiere a que no guardara informacion sobre conexiones anteriores, pero como los sitios necesitan guardar información sobre visitas pasadas, hacen uso de las *cookies*, la cual es información que puede guardar el sistema del cliente.

3 Mensajes *HTTP*.

"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. 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." (*Mdn Web Docs*, 2022)
Estos mensajes de *HTTP* estan compuestos de texto plano, lo cual tiene un inconveniente, el cual es que son textos largos y tediosos de hacer, asi que los desarrolladores web optan por definir estos mensajes en archivos de configuracion, *APIs* y otros medios. La estructura de estos mensajes es la siguiente: en primer lugar esta la linea inicial, la cual consiste de la accion requerida, la *url* del recurso, la versión *HTTP* que soporta el cliente, eso seria para las peticiones, para la respuesta se compone de la versión *HTTP*, el codigo de respuesta y el mensaje, despues viene las cabeceras y el cuerpo del mensaje el cual es opcional.

4 Métodos de Petición *HTTP*.

"*HTTP* define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado. Aunque 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* (en-US), o cacheable."(*Mdn Web Docs*, 2022)
En la tabla 1 se muestran los metodos de peticion y su significado.

4.1 Métodos de Petición.

Metodo	Acción
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.

Table 1: Tabla Métodos de Petición.

5 Códigos de Respuesta *HTTP*.

El codigo de respuesta es un número que indica el estado de la petición solicitada, el demas contenido de la respuesta dependera del resto del codigo. Hay cientos de codigos en esta categoria, pero se pueden englobar en cinco formatos como vemos en la tabla 2. (*wikipedia*, 2022)

5.1 Códigos de Respuesta.

Formato de código	Significado
1xx	Respuestas informativas. Indica que la petición ha sido recibida y se está procesando.
2xx	Respuestas correctas. Indica que la petición ha sido procesada correctamente.
3xx	Respuestas de redirección. Indica que el cliente necesita realizar más acciones para finalizar la petición.
4xx	Errores causados por el cliente. Indica que ha habido un error en el procesado de la petición a causa de que el cliente ha hecho algo mal.
5xx	Errores causados por el servidor. Indica que ha habido un error en el procesado de la petición a causa de un fallo en el servidor.

Table 2: Tabla Códigos de Respuesta.

6 Caberas *HTTP*.

"Las cabeceras HTTP contienen información de metadatos como, por ejemplo, la información de autenticación de seguridad, el agente de usuario que se utiliza y los metadatos de control de memoria caché. Las cabeceras HTTP estándar se definen en la especificación HTTP; sin embargo, puede utilizar cabeceras HTTP personalizadas, si es necesario." (*IBM*, 2021)
"Las cabeceras HTTP son demasiado utiles para los programas de intermedio y cliente a comprender la información sobre las solicitudes y las respuestas para las aplicaciones. Las cabeceras HTTP contienen información de metadatos. Los códigos de estado HTTP proporcionan información de estado sobre la respuesta." (*IBM*, 2021)
Las cabeceras se puden clasificar segun su funcionalidad y hay ocho clasificiones, las primeras son las cabecera que indican las capacidades aceptadas por el que envía el mensaje, la segunda es la que describe el contenido, la tercera son las que hacen referencia a URIs, las cuartas son las que permiten ahorrar trasmisiones, la quinta es para el control de cookies, la sexta son las de autenticación, las septimas son para describir la comucación y las octabas se clasifican en dos las Range y Max-Forward.

7 Ejemplo de Dialogo HTTP.

"Para obtener un recurso con el URL http://www.example.com/index.html
1.-Se abre una conexión 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 habría que codificarlo en la URL de la forma http://www.example.com:XXXX/index.html
2.-Se envía 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
[Línea en blanco]
```

La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página 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>Título del sitio</title>
</head>
<body>
<h1>Página principal de tuHost</h1>
(Contenido)
  .
  .
  .
</body>
</html>
```

(*wikipedia*, 2022)

8 Conclusiones.

En conclusión HTTP es un protocolo de súper utilidad, que permite las conexiones con sitios web de manera rápida y que al momento de haber un error en la comunicación saltaran mensajes de error que dependiendo de la situación enviaran uno u otro mensaje, permitiendo al usuario identififcar el problema ya sea para que pueda solucionarse por cuenta del usuario o para avisar al administrador del sitio web. Al igual que hay una leve diferencia entre HTTP y HTTPS, ambos cumplen con las mismas responsabilidades pero con la diferencia de que HTTPS cuenta con mayor seguridad. HTTP tiene una gran planeación de por medio al punto de que es el protocolo mas ocupado a nivel mundial para acceder a sitios web.

9 Referencias.

Protocolo de transferencia de hipertexto. (2022, 19 de febrero). Wikipedia, La enciclopedia libre. Fecha de consulta: 21:42, marzo 13, 2022 desde https://es.wikipedia.org/w/index.php?title=Protocolo_de_transferencia_de_hipertextooldid=141779019.
Mensajes HTTP - HTTP — MDN. (2022, 12 marzo). Mdn Web Docs. Recuperado 13 de marzo de 2022, de https://developer.mozilla.org/es/docs/Web/HTTP/Messages
Métodos de petición HTTP - HTTP — MDN. (2022, 12 marzo). Mdn Web Docs. Recuperado 13 de marzo de 2022, de https://developer.mozilla.org/es/docs/Web/HTTP/Methods
Definición de cabeceras HTTP y códigos de respuesta para aplicaciones RESTful. (s. f.). IBM. Recuperado 13 de marzo de 2022, de https://www.ibm.com/docs/es/was/9.0.5?topic=applications-defining-http-headers-response-codes-restful