

# HTTP: El protocolo detrás de las páginas Web.

## Resumen

Desde el nacimiento del World Wide Web y del protocolo HTTP en 1990 en el CERN (Centro Europeo de Física de Partículas) hasta este día, el mundo se encuentra conectado o "hipervinculado" a través de millones de páginas Web.

Muchas de estas páginas han sido realizadas desconociendo los mecanismos de negociación entre un navegador y un servidor Web.

El presente trabajo pretende dar una introducción al protocolo HTTP y de esta manera, conocer las bases del intercambio de información entre clientes y servidores Web.

Palabras claves: Métodos, URL, WWW.

## Introducción.

Siempre ha existido un génesis en todas las cosas, y la tecnología no escapa a ello.

Las tecnologías relacionadas con el nacimiento del WWW y HTTP se remontan con la creación del ARPA (Advanced Research Projects Agency) a finales de la Segunda Guerra Mundial, donde Vannevar Bush publica un artículo llamado "As We May Think" donde expone un dispositivo llamado Memex, que permite hacer y seguir ligas entre documentos, muy similar al hipertexto que conocemos actualmente [2].

El inicio de Internet se produjo con la creación del ARPANET, dentro de la cual se realizaron diferentes inventos como el packet-switching por Paul Baran, Ethernet por Bob Metcalf, TCP por Vint Cerf y Bob Khan [2].

La implementación de TCP permite la conexión de diversas redes formando una verdadera Internet [2].

Sentadas las bases de una verdadera interconexión entre redes, se empieza a desarrollar la concepción del WWW y HTTP, con el invento del Mouse por Doug Englebart, la concepción del término hipertexto por Ted Nelson y la creación de Enquire por Tim Berners-Lee para su uso personal el cual le ayuda a recordar conexiones entre varias personas y proyectos [1] [2] [3].

En 1990 Tim Berners-Lee inventa el nombre de "World Wide Word" como proyecto, así como el HTTP, define también como URI a la dirección de los documentos a acceder.

De esta forma nacen el primer servidor Web, con su respectivo navegador Web (cliente) a base de caracteres y el esquema de nombramiento para los documentos en el CERN.

La dirección de este servidor era `http://nxox01.cern.ch/hypertext/WWW/TheProject.html` [1][2][3]

Posteriormente Marc Andreessen escribe el primer navegador web gráfico llamado Mosaic [4]

Actualmente el mundo se encuentra virtualmente conectado por medio de Internet y con sólo un clic podemos obtener información desde cualquier parte del planeta.

En un principio sólo fue utilizado para la distribución de documentos pero el contenido era solo estático, es decir, no cambiaba a menos que el documento fuera modificado.

Después con la necesidad de tener información actualizada, se empezaron a realizar contenidos dinámicos, los cuales eran obtenidos de bases de datos y mostrados al momento.

Usar el WWW como entorno de aplicación se fue desarrollando poco a poco, ya que cada etapa de tecnologías servía como plataforma para la aparición de nuevas ideas [6].

Pero aún cuando las tecnologías y aplicaciones utilizadas en el entorno Web sean muy sofisticadas, el conocer como operan los diferentes mecanismos de negociación entre diferentes entidades no es algo esotérico ni mucho menos trivial.

Estos mecanismos representan las bases de negociación entre navegadores y servidores Web, y de esta forma intercambiar información sin importar donde se localice.

## La especificación HTTP

Las normas establecidas por Internet se especifican en un documento RFC (Request For Comments).

Las RFC se enumeran y nunca cambian una vez que se publican, pero si alguna queda obsoleta, se publica nuevamente.

Existen varios RFC para HTTP:

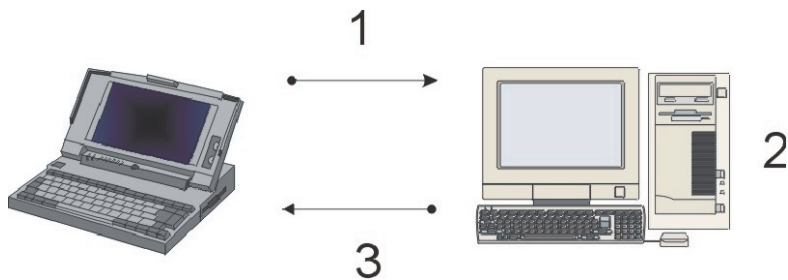
- RFC 1945.- Descripción de HTTP versión 1.0
- RFC 2068.- Descripción inicial de la versión 1.1
- RFC 2616.- Versión actualizada de la versión 1.1 [7]

HTTP es un protocolo de nivel de aplicación para sistemas de información distribuida y colaborativa [5].

Sirve como base al modelo Cliente - Servidor Web, ya que describe como se envían documentos por Internet.

## Interacción cliente/servidor

El protocolo HTTP es un protocolo de solicitud/respuesta, dicha interacción se muestra a continuación:



1. El cliente envía una solicitud HTTP al servidor. Un cliente -usualmente un navegador Web - envía una solicitud al servidor.
2. El servidor procesa la solicitud. El servidor espera en el puerto especificado (de forma predeterminada 80), la solicitud enviada por el cliente (navegador web) localiza el recurso especificado y lo envía de vuelta al cliente. Esta interacción se realiza utilizando sockets como conexiones.
3. El navegador Web muestra el contenido. El cliente recibe lo que el servidor le envió y si estos datos son comandos HTML se muestran, en caso contrario, se llama a una aplicación que le auxilia en la interpretación con un recurso en particular y le cede los datos.

Los recursos (documentos, imágenes, etc.) deben de tener una forma de nombrado especial para poder acceder a ellos, a través de un navegador Web.

## Identificadores de Recursos

Un URL (Uniform Resource Locator) representa una forma de nombrado dentro de Internet para acceder a recursos específicos y consta de cuatro partes:



Figura 2. Estructura de un URL.

- El protocolo, Indica al navegador Web que protocolo de Internet va a utilizar al acceder el recurso. HTTP es el protocolo nativo del Web. Existen otros protocolos como son: FTP, Gopher, Mailto, News.
- El nombre del servidor. El cual es un nombre de dominio de Internet que identifica la ubicación del servidor, pueden utilizar las direcciones IP.
- Número de puerto. De forma predeterminada es el número 80, aunque puede especificarse otro número de puerto, el cual es accedido por medio de dos puntos (:).
- Acceso al recurso. Representa la dirección al recurso que se va a acceder y comienza con una diagonal después del servidor y/o puerto. [5] [6]

## Mensajes HTTP

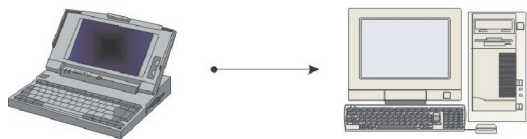
Los mensajes HTTP consisten en solicitudes del cliente al servidor y respuestas desde el servidor al cliente. Todos ellos con un formato específico: Línea de comienzo, cero o más campos de encabezados, una línea en blanco que indica el final de los encabezados y posiblemente un cuerpo de mensaje.

## Solicitud HTTP

El primer paso dentro de este intercambio la realiza el cliente, invocando una solicitud al servidor.

Se compone de una línea de solicitud, uno o más campos de encabezado de solicitudes y un cuerpo de entidad opcional.

Las líneas se separan por medio de retorno de carro y avance de línea (crLf -carriage-return/line feed-).



**Sintaxis de solicitud HTTP**

```
<method> <resource identifier> <HTTP version> <crf>
[<header>: <value>] <crf>
...
[<header>: <value>] <crf>
  blank line <crf>
[Entity body]
```

**Ejemplo**

```
GET /path/file.html HTTP/1.1
Accept: text/html
Accept: audio/x
```

Figura 3. Formato de solicitud HTTP

- La línea de solicitud se compone de tres campos, separados por espacios en blanco. El primer campo indica el método de HTTP, el segundo método indica el nombre del recurso a obtener y el tercer campo indica la versión de HTTP.
- Los campos de encabezado de solicitudes, indican al servidor la información que el cliente puede manejar. Estos campos consisten de un nombre, dos puntos y el valor del campo.
- El cuerpo de entidad se utiliza en raras ocasiones para informarle al servidor de forma masiva.

## Respuesta HTTP

Una vez que el servidor ha recibido una solicitud, esta es procesada y vuelta al cliente. La sintaxis es la siguiente:

Consiste en una línea de encabezado respuesta, uno o mas campos de respuesta opcionales y un cuerpo de entidad opcional.

Al igual que la solicitud, las líneas de la respuesta se separan por medio de retorno de carro y avance de línea. (crfl)



**Sintaxis de respuesta HTTP**

```
HTTP/1.1 200 OK
Content-type: text/html
Content-length: 1234
```

**Ejemplo**

```
HTTP/1.1 200 OK
Content-type: text/html
Content-length: 1234
```

Figura 4.- Formato de respuesta HTTP.

- Línea de encabezado de respuesta, en ella se envía la versión de HTP, el estado de la respuesta y una explicación de este estado.
- Campos de encabezado de respuesta, envían información que describen los atributos del servidor asi como del documento HTML que es enviado al cliente.
- Cuerpo de entidad, contiene el recurso que el cliente solicita, por lo general en HTML.

## Métodos HTTP.

El método indica la forma de operación sobre el recurso dado. Se muestra a continuación una cuadro de los encabezados soportados en la versión 1.1.

OPTIONS	Permite a los clientes determinar las capacidades de un servidor sin recuperar el recurso.
GET	Recupera el URL solicitado.
HEAD	Idéntico a Get, sólo que el servidor solo envía los encabezados.
POST	Envía datos a un URL específico.
PUT	Almacena los datos en el URL especificado.
DELETE	Borra el recurso indentificado por el URL.
TRACE	Notifica todo lo que se reciba del cliente.
CONNECT	Reserva para uso con un proxy que puede ser dinámicamente cambiado a tunnel.

Cuadro 1. Métodos en HTTP 1.1

## Campos de Encabezados

Los campos de encabezados son datos que tanto el cliente y el servidor utilizan para poder informarse-solicitarse uno al otro ciertas capacidades.

El formato de los campos es: nombre del campo seguido del signo = y el valor del campo.

Message header = field-name : [field-value]

Estos campos se dividen en tres clases: Generales, de Solicitud y de Respuesta.

Los campos generales se ocupan en ambas direcciones (solicitudes y respuestas).

Cache-Control	Establece como los servidores proxy deberán almacenar temporalmente las páginas.
Connection	Indica si el cliente puede manejar conexiones persistentes HTTP.
Date	Representa la fecha y hora en la que el mensaje fue originado.
Pragma	Indica que se debería reenviar la petición aun si cuenta con una copia local.
Trailer	Identifica los campos de encabezado que están presentes en la parte inicial de un mensaje que se envía por secciones.
Transfer Encoding	Indica una codificación de transferencia por secciones.
Upgrade	Permite al cliente especificar un protocolo diferente a HTTP.
Via	Muestra los sitios intermedios por las que pasó la petición.
Warning	Permite a los clientes prevenir errores de almacenamiento temporales.

Cuadro 2. Campos de encabezados generales

En los encabezados de solicitud, el cliente le indica al servidor las capacidades que puede tener.

Accept	Estable los tipos MIME que el navegador puede manejar.
Accept-Charset	Indica el conjunto de caracteres que utiliza el navegador.
Accept-Encoding	Tipos de codificación que el cliente puede manejar.
Accept-Language	Establece los idiomas preferidos por el cliente.
Authorization	Los clientes utilizan este encabezado para identificarse cuando acceden a páginas protegidas por contraseñas.
Expect	Se indica al servidor que tipos de comportamientos espera
From	Devuelve la dirección de correo electrónico de la persona responsable de la petición HTTP.
Host	Indica el Host y puerto que se otorgan en el URL.

If-Match	Cliente proporciona lista de etiquetas de entidad y la operación se realiza sólo si una de ellas coincide.
If-Modified-Since	El cliente desea la página solo si ha sido modificada desde la fecha indicada.
If-None-Match	Funciona igual que Match, excepto que la operación se realiza sólo si no coincide ninguna etiqueta.
If-Range	Permite una copia parcial de un documento.
If-Unmodified-Since	Es como If-Modified-Since pero al revés.
Proxy-Authorization	Permite que los clientes se identifiquen ante los proxy que lo requieran.
Range	Permite que el cliente cuente con una copia parcial de un documento
Referer	Indica el URL de la página Web a la que hace referencia
User-Agent	Identifica al cliente que hace la petición.

Cuadro 3. Campos de encabezados de solicitudes

En los encabezados de respuesta el servidor le indica al cliente, acerca de sus capacidades así como del acceso al recurso solicitado por el cliente.

Accept-Ranges	Indica al cliente si puede aceptar encabezados de petición Range.
Age	Hace cuanto el documento fue generado por el servidor original.
Etag	Da nombre a los documentos devueltos.
Location	Notifica al cliente la dirección del documento.
Retry-After	Indica al cliente que tan pronto puede repetir la operación.
Server	Identifica al Servidor Web.
Vary	Indica al cliente cuáles encabezados pueden utilizarse para determinar si el documento puede ser almacenado en cache.
WWW-Authenticate	Indica al cliente que tipo de autorización y área debe proporcionar al cliente.

Cuadro 4. Campos de encabezados de respuesta

Una entidad consiste de metainformación en la forma de encabezados [5].

Los mensajes de solicitudes y respuestas pueden transferir una entidad si no es restringido por el método de solicitud o el código de respuesta.

Esta información define como será el cuerpo del mensaje y si el cuerpo no esta presente, indica cómo es identificado el recurso. Algunas son opcionales.

Allow	Indica los métodos de petición que permite el servidor.
Content-Encoding	Indica cómo fue codificada la página durante la transmisión.
Content-Language	Indica el lenguaje en el que fue escrito el lenguaje.
Content-Length	Devuelve el tamaño en bytes de los datos POST.
Content-Location	Proporciona la dirección alternativa del documento solicitado
Content-MD5	Permite una verificación de integridad del mensaje para clientes que deseen confirmar que recibieron el documento completo.
Content-Range	Enviado como respuesta de documentos parciales.
Content-Type	Indica el tipo MIME del documento que se responde.
Expires	Indica el momento en que deberá ser considerado como caducado.
Last-Modified	Indica el momento en que el documento fue modificado por última vez.

Cuadro 5. Campos de encabezados utilizados en solicitudes y respuestas

## Códigos de Estado

Cuando el servidor Web responde a un navegador Web, la respuesta por lo general consta de: una línea de estado, algunos encabezados, una línea en blanco y el documento.

Esta línea de estado se forma a su vez de la versión de HTTP, un código de estado y un mensaje de dicho código.

Los códigos se organizan en cinco categorías:

- 100-199. Informativos
- 200-299. Petición exitosa

- 300-399. Acción de redirección
- 400-499. Error del cliente.
- 500-599. Error del servidor.

## Conclusiones

Conocer el protocolo HTTP, nos brinda una ventaja en el desarrollo de las aplicaciones basadas en ambiente Web.

Esta ventaja se muestra claramente al proporcionar aplicaciones eficientes y robustas en el uso sistemas cliente servidor utilizando el paradigma Web.

Las aplicaciones pueden ser desde sencillas páginas Web hasta sistemas complejos Cliente Servidor utilizando como medio de transporte el protocolo HTTP

## Referencias

- [1] A little history of the World Wide Web  
<http://www.w3.org/History.html>
- [2] Internet Pioneers  
<http://www.ibiblio.org/index.html>
- [3] BERNERS-LEE, TIM  
Internet Pioneers  
<http://www.ibiblio.org/pioneers/lee.html>
- [4] ANDRESEN, MARC  
Internet Pioneers. <http://www.ibiblio.org/pioneers/andresen.html>
- [5] RFC 2616.
- [6] ORFALI ROBERT, HARKEY DAN, EDWARDS JERI.  
1998 "Cliente/Servidor Guía de supervivencia" 2da Edición Ed. McGraw-Hill.
- [7] HANNA PHIL  
2002 "JSP Manual de referencia" Ed. McGraw-Hill.

Roberto Hernández Sampedro.  
Universidad Virtual.  
Universidad Tecnológica de la Mixteca.

