



# Arquitectura de Información en Web

Cardeña Matamoros Antonio Carlos

## HTTP

Castro Bouquet Ildfonso  
Fragoso Oseguera Juan Pablo

Fecha Entrega: 28 Enero 2022

Arquitectura de Información en Web	1
Resumen	3
Desarrollo	4
Protocolo HTTP	4
Funcionalidad	4
Transacción HTTP	5
Métodos	6
Ventajas Versiones HTTP/1.X, HTTP/2.X, HTTP/3.X	6
Conclusiones	10
Referencias	11

## Resumen

El protocolo HTTP es una forma de comunicación entre la interfaz de usuario o cliente y el servidor. Permite la comunicación entre dos sistemas obteniendo, desplegando y modificando datos en cualquiera de los dos extremos por medio de un sistema llamado transacción. Utiliza métodos para acceder a esta información, los más conocidos son GET, POST, PUT y DELETE.

Conforme el tiempo se fue adaptando a las necesidades y junto a estas su velocidad de comunicación. Esto incorpora 3 versiones principales y varias versiones de arreglo de funcionamiento, las cuales tienen ventajas mientras evoluciona.

# Desarrollo

## Protocolo HTTP

*“Hypertext transfer Protocol”* es un conjunto de reglas para hacer solicitudes de datos, es la base para el desarrollo WEB, este se comunica entre el servidor y la página web, en el cual normalmente se divide en distintos documentos. Propuesta por Tim Berners-Lee, para cubrir necesidades poniendo como ejemplo World Wide Web.

Se basa en solicitudes y respuestas, donde el cliente establece conexión con el servidor junto a una solicitud de datos y el servidor manda una respuesta con un mensaje similar, que contiene la función, objeto o recurso en el que se actúa.

## Funcionalidad

HTTP es *“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”*, (Editorial Etecé, 2021). Comúnmente las páginas web tienen contenido que representa información del mundo real, ya sea artículos como ropa o calzado en E-Commerce, o datos en tiempo real sobre deportes, entre otros. Para poder mostrar estos datos y que estén actualizados de forma automática y no teniendo que introducirlos de una forma ‘hardcoded’ se necesita acceder a una base de datos.

El almacén con estos datos, o bases de datos, permiten almacenar información en diferentes estructuras, y se requiere solicitar esta información. Entonces una computadora

que sirve a un puerto específico, 80 para HTTP (TCP por defecto), va a poder tener disponible información a compartir con el cliente que lo necesite para su página web, sin embargo se necesitan establecer ciertos criterios para la comunicación entre estos elementos, el cliente-servidor y los proxies. Para eso fue creado HTTP.

Se tiene que considerar también que no todos deberían tener acceso a los datos, por lo que se necesitan las credenciales correctas para acceder a este servicio web, sin olvidar que la comunicación entre dominios, esquemas, o puertos diferentes no es permitida directamente y va a estar sujeta a observación por CORS. "CORS es un mecanismo que utiliza cabeceras HTTP adicionales para permitir que un usuario obtenga permiso para acceder a recursos seleccionados desde un servidor, en un origen distinto (dominio) al que pertenece.

Un agente crea una petición HTTP de origen cruzado cuando solicita un recurso desde un dominio distinto, un protocolo o un puerto diferente al del documento que lo generó", (MDN contributors, 2022).

## Transacción HTTP

1. Ingreso a una URL, accediendo a un documento HTML o ser redireccionados a Location del cliente WEB
2. El cliente WEB descodifica la URL, obteniendo el protocolo de acceso, dirección DNS o IP del servidor, puerto opcional y el objeto del servidor.
3. Se establece conexión TCP/IP con el servidor, llamando al puerto TCP<sup>1</sup>. Realizando una petición. Para ello se usa GET, POST, HEAD entre otros, la URL, la versión del protocolo HTTP, y un conjunto de información (Datos del navegador para el servidor).

---

<sup>1</sup>TCP (Transmission Control Protocol) es un protocolo de red que permite una conexión e intercambio de data streams entre 2 hosts.

4. Sale respuesta del servidor al cliente, formado por un código de estado y el tipo de dato MIME<sup>2</sup> de la información de retorno, junto a la información requerida.
5. Cierre de la conexión.

## Métodos

Un método es una instrucción para indicar la acción a realizar. También llamados HTTP verbs.

- GET solicita una representación de un recurso, **sólo** recupera datos.
- HEAD solicita una respuesta como GET pero **sin** el cuerpo de la respuesta.
- POST enviar una entidad a un recurso en específico, **normalmente** afecta el servidor.
- PUT reemplaza representaciones **actuales** del recurso de destino con la carga en la petición.
- DELETE borra un recurso en **específico**.
- CONNECT establece un túnel con el servidor **identificado** por el recurso.
- OPTIONS describe las opciones de comunicación para el recurso de **destino**.<sup>2</sup>
- TRACE **prueba** el retorno de mensaje a lo largo de la ruta al recurso de destino.
- PATCH aplicar modificaciones **parciales** a un recurso.

## Ventajas Versiones HTTP/1.X, HTTP/2.X, HTTP/3.X

**Tabla 1**

*Evolución HTTP con características y ventajas*

---

<sup>2</sup>La manera estándar de mandar contenido a través de la red. Especifican tipo de datos como por ejemplo texto, imagen, audio, etc.

Versión	Año	Características	Ventajas
0.9	1991	<p>Peticiones sin especificar el verbo (solo GET).</p> <p>Los errores y respuestas se devuelven en la respuesta de tipo archivo HTML.</p>	
1.0	1996	<p>Soporte de verbos como GET, POST y HEAD.</p> <p>La versión del protocolo está incluida en la petición.</p> <p>Implementación de códigos de estado HTTP.</p> <p>Cabeceras de HTTP para peticiones y respuestas.</p> <p>Transmisión de documentos no HTML.</p>	<p>Diversidad de peticiones.</p> <p>Retroalimentación por medio de códigos de estado.</p> <p>Transmisión de la metadata gracias a las cabeceras.</p> <p>Variedad de contenidos para transmitir.</p>
1.1	1997	<p>Verbos PUT y DELETE.</p> <p>Web orientada a recursos REST.</p> <p>Se añadió el enrutamiento a la especificación.</p> <p>División de respuestas a peticiones en sub-partes.</p> <p>Adición de controles adicionales a los mecanismos de gestión de la caché.</p>	<p>Reutilización de conexiones para evitar abrir contenido sin cambiar.</p> <p>Reducción de latencia de la comunicación al tener llamadas de petición de datos en una sola por Pipelining.</p> <p>Acuerdo más adecuado para el cliente-servidor en el intercambio por la negociación de contenido.</p>

		Negociación de contenido añadida a la especificación. Cabecera Host.	Múltiple alojamiento de dominios en la misma dirección IP por la cabecera Host.
2.0	2015	Multiplexed streams. Server push. Compresión de headers o cabeceras. Formato binario.	Provee recursos en partes pequeñas, pasando todos ellos a través de una misma conexión. Envío de recursos a la caché del navegador sin que éste los solicite utilizando la conexión, que probablemente va a necesitar. Las cabeceras son empaquetadas en un solo bloque comprimido para enviarse, y del otro lado hay una decodificación del bloque. Eficiente interpretación del mensaje, al tener solo una forma de interpretarlo a diferencia de HTTP/1.X con 4 formas de hacerlo.
3.0	2020	Utilización del protocolo UDP en lugar de TCP, por medio del protocolo de transporte QUIC. Uso nulo de handshakes. Cifrado con TLS. Secuencias QUIC.	No se requiere de un intercambio continuo de información, mejorando la latencia de conexión. Los datos a transferir son imposibles de leer.



			Las secuencias se entregan de forma independiente, y si se pierde algún paquete, esto no afectará a los demás.
--	--	--	--

, (Angulo, 2016), (MDN contributors, 2022), y (Fernández, 2020).

## Conclusiones

Castro Bouquet Ildelfonso:

Aprendí que el protocolo tiene como función principal ser la comunicación entre dispositivos, así como cuáles son los pasos que sigue para llevar a cabo esta comunicación. Pude vincular los métodos con los vistos en bases de datos, por lo que pude entender con mayor facilidad sus usos. La aplicación de este tema será relevante para el curso ya que da inicio o me imagino a https, que es algo esencial para el desarrollo seguro de una plataforma web. Espero que de la misma manera sirva para crear APIs ya que tiene que ver con la comunicación entre servidor y cliente.

Fragoso Oseguera Juan Pablo:

Pude rescatar algunos conceptos nuevos y profundizar en temas que ya conocía anteriormente. La transacción HTTP es algo que no era de mi conocimiento, tampoco tenía claridad en qué aportaba cada versión, y también pude ver métodos que no son muy usados, pero que en su momento fueron desarrollados para propósitos específicos. Con esta investigación puedo conocer y aplicar métodos para recibir información en el Front de una forma más controlada, además de qué especificaciones del contenido a recibir en la parte de la generación de RESTful APIs en el Back. Dentro de mi carrera se puede recordar este proceso para el manejo de redes, y que para las telecomunicaciones es importante el empaquetado de datos y de protocolos para poder mantener comunicaciones de forma segura y óptima.

## Referencias

Angulo, J. (2016). ¿Qué es HTTP/2 y qué ventajas tiene sobre HTTP 1.1 ? SomosTechies.

Recuperado en enero de 2022, de <https://somostechies.com/que-es-http2/>

Editorial Etecé. (2021). HTTP - Concepto, para qué sirve y cómo funciona. Concepto.

Recuperado en enero de 2022, de <https://concepto.de/http/>

Fernández, Y. (2020). HTTP/3: qué es, de dónde viene, y qué es lo que cambia para buscar

un Internet más rápido. Xataka. Recuperado en enero de 2022, de <https://www.xataka.com/basics/http-3-que-donde-viene-que-que-cambia-para-buscar-internet-rapido>

Herramientas WEB. El protocolo HTTP. Herramientas WEB. Recuperado en enero de 2022,

de <https://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html>

MDN contributors. (2022). Control de acceso HTTP (CORS) - HTTP | MDN. MDN Web

Docs. Recuperado en enero de 2022, de <https://developer.mozilla.org/es/docs/Web/HTTP/CORS>

MDN contributors. (2022). Evolución del protocolo HTTP - HTTP | MDN. MDN Web Docs.

Recuperado en enero de 2022, de [https://developer.mozilla.org/es/docs/Web/HTTP/BasicsofHTTP/Evolution\\_of\\_HTTP](https://developer.mozilla.org/es/docs/Web/HTTP/BasicsofHTTP/Evolution_of_HTTP)

P

MDN contributors. (2022). Metodos de peticion HTTP- HTTP | MDN. MDN Web Docs.  
Recuperado en enero de 2022, de  
<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>