

# HTTP/2

Bryan Ivan Montiel Ortega

Noviembre 2023

# Índice

<b>Índice.....</b>	<b>1</b>
<b>¿Qué es HTTP/2?.....</b>	<b>2</b>
<b>¿Qué Es un Protocolo?.....</b>	<b>2</b>
<b>El Objetivo de Crear HTTP/2.....</b>	<b>3</b>
<b>La necesidad de HTTP/2.....</b>	<b>5</b>
<b>¿Cómo HTTP/2 Funciona con HTTPS?.....</b>	<b>6</b>
<b>Características principales de HTTP/2.....</b>	<b>7</b>
Server Push.....	7
Multiplexación.....	7
Control de flujo y priorización de flujo.....	8
Compresión de encabezado.....	8
Protocolo binario.....	9
<b>Beneficios de rendimiento de HTTP/2.....</b>	<b>10</b>
<b>Compatibilidad y adopción.....</b>	<b>10</b>
<b>Implementación y soporte.....</b>	<b>10</b>
<b>Limitaciones de HTTP/2.....</b>	<b>11</b>
<b>Análisis Comparativo de Rendimiento de HTTPS, SPDY y HTTP/2.....</b>	<b>12</b>
<b>Configuración básica en Apache.....</b>	<b>13</b>
<b>Bibliografía.....</b>	<b>14</b>

# ¿Qué es HTTP/2?

HTTP fue propuesto originalmente por Tim Berners-Lee, el pionero de la Red Mundial que diseñó el protocolo de aplicación teniendo en mente la simplicidad de alto nivel para realizar las funciones de comunicación de datos entre servidores y clientes Web.

La primera versión documentada de HTTP fue lanzada en 1991 como HTTP0.9, que luego condujo a la introducción oficial y reconocimiento de HTTP1.0 en 1996. HTTP1.1 seguida en 1997 y desde entonces ha recibido pocas mejoras iterativas.

En febrero de 2015, el IETF (Grupo de Trabajo de Ingeniería de Internet) un Grupo de Trabajo HTTP revisó y desarrolló la segunda versión de la aplicación en forma de protocolo HTTP/2. En mayo de 2015, la especificación HTTP/2 aplicación oficialmente fue estandarizada en respuesta al protocolo SPDY compatible con HTTP.



El protocolo HTTP/2 hace que Internet sea más rápido y eficiente al comprimir información y ejecutar múltiples solicitudes simultáneamente, lo que reduce la velocidad de carga . Esto significa que los sitios web cargan más rápido y puedes acceder al contenido más rápido.

Además, con la versión 2.0 de HTTP, los servidores te envían los recursos que necesitas antes de que los solicites , como imágenes o archivos. Todas estas mejoras funcionan juntas para un rendimiento web más rápido y una experiencia en línea suave como la seda.

De acuerdo con RFC-7540 by the Internet Engineering Task Force (IETF), HTTP/2 permite un uso más eficiente de los recursos de red y reduce la percepción de latencia de la red al

- introduciendo la compresión del campo del encabezado
- permitiendo múltiples intercambios concurrentes en la misma conexión
- permitiendo control de flujo y priorización de arroyos
- introduciendo envío no solicitado de representaciones desde servidores a clientes .

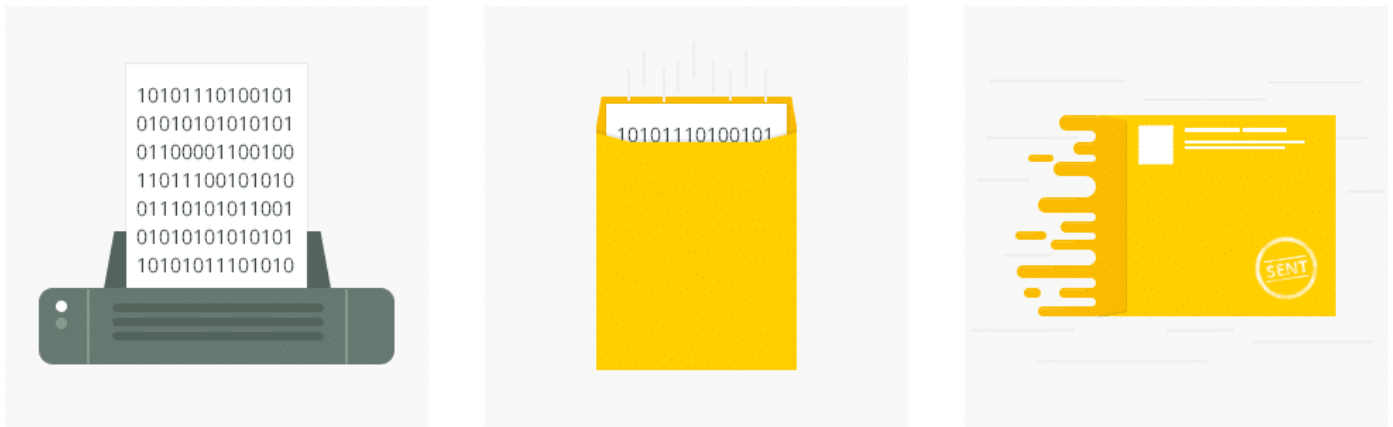
## ¿Qué Es un Protocolo?

El debate HTTP/2 vs HTTP1 debe proceder con una breve guía sobre el término protocolo. Un protocolo es un conjunto de reglas que rigen los mecanismos de comunicación de datos entre los 'clientes' (por ejemplo, navegadores web utilizados por los usuarios de internet para solicitar información) y 'servidores' (las máquinas que contengan la información solicitada).

Los protocolos generalmente consisten de tres partes principales: Encabezado, carga útil y pie de página.

- El Encabezado colocado antes de la carga útil contiene información tal como las direcciones de origen y destino, así como otros detalles (como el tamaño y tipo) con respecto a la carga útil.
- La carga útil es la información transmitida mediante el protocolo.
- El pie sigue la carga y funciona como un campo de control para enrutar las solicitudes de cliente-servidor a los destinatarios, junto con el encabezado para asegurar que la carga útil se transmita libres de errores.

El sistema es similar al 'servicio de correo'. La carta (Carga útil) se inserta en un sobre (Encabezado) con dirección de destino escrita en ella y sellada con pegamento y sello (pie de página) antes de que se expidan.



Excepto que la transmisión de información digital en forma de 0s y 1s no es tan sencillo y requiere una nueva dimensión de innovación en respuesta a aumentar los avances tecnológicos que surgen con el crecimiento explosivo del uso de internet.

Protocolo HTTP constaba inicialmente de comandos básicos: GET, para recuperar la información del servidor y POST, para entregar la información solicitada por el cliente. Este sencillo y aparentemente aburrido conjunto de algunos comandos para obtener datos y enviar una respuesta esencialmente establecieron las bases para la construcción de otros protocolos de red. El protocolo es otra iniciativa destinada a mejorar la experiencia y eficacia del usuario de internet, necesitándose aplicación HTTP/2 para mejorar la presencia en línea.

## El Objetivo de Crear HTTP/2

Desde sus inicios a principios de los años noventa, HTTP ha visto sólo unas pocas grandes revisiones. La versión más reciente, HTTP1.1 ha servido al mundo cibernético durante más de 15 años. Las páginas Web en la actual era de actualizaciones en información dinámica, formatos de contenido multimedia intensivos en recursos y excesiva inclinación hacia el rendimiento web han colocado a las tecnologías de

protocolo antiguas en la categoría de legado. Estas tendencias exigen importantes HTTP/2 cambios para mejorar la experiencia de internet.

El objetivo principal de la investigación y el desarrollo de una nueva versión de HTTP se centra alrededor de tres cualidades raramente asociadas con un único protocolo de red sin requerir tecnologías de redes adicionales de alto rendimiento, simplicidad y solidez. Estos objetivos se logran mediante la introducción de funciones que reducen la latencia en la tramitación de las solicitudes del navegador con técnicas como la multiplexación, la compresión, la priorización y la solicitud de inserción de servidor.

Mecanismos tales como el control de flujo, manejo de errores y actualización funcionan como mejoras para el protocolo HTTP para desarrolladores para asegurar un alto rendimiento y flexibilidad de aplicaciones basadas en web.

El sistema colectivo permite a los servidores responder eficientemente con más contenido que originalmente solicitado por los clientes, eliminando la intervención del usuario para solicitar información continuamente hasta que el sitio web esté completamente cargado en el navegador web. Por ejemplo, la capacidad de inserción del servidor con HTTP/2 permite a los servidores responder con una página llena de contenidos distintos de la información ya disponible en la cache del navegador. La compresión eficiente de archivos de encabezado HTTP minimiza el exceso de protocolo para mejorar el rendimiento con cada solicitud del explorador y la respuesta del servidor.

Los cambios a HTTP/2 están diseñados para mantener la interoperabilidad y compatibilidad con las ventajas HTTP1.1. HTTP/2 se espera aumenten con el tiempo sobre la base de experimentos del mundo real y su capacidad para abordar cuestiones relacionadas con el rendimiento real en comparación en el mundo real con HTTP1.1 impactará en gran medida su evolución a largo plazo.

La aplicación de HTTP/2 no permitirá soporte automático para todos los tipos de cifrado disponibles con HTTP1.1, pero sin duda abre la puerta a mejores alternativas de cifrado adicionales o actualizaciones de compatibilidad en el futuro cercano. Sin embargo, comparaciones de características tales como HTTP/2 vs HTTP1 y SPDY vs HTTP/2 presentan sólo el último protocolo de aplicación como el ganador en términos de rendimiento, seguridad y confiabilidad por igual.



# La necesidad de HTTP/2

El auge de los dispositivos móviles, el aumento del uso de contenido multimedia y la demanda de experiencias web más rápidas y eficientes implicaron el desarrollo de un nuevo protocolo.

HTTP/1.1 tenía ciertas limitaciones que dificultaban el rendimiento, como su incapacidad para manejar las solicitudes concurrentes de los clientes de manera eficiente.

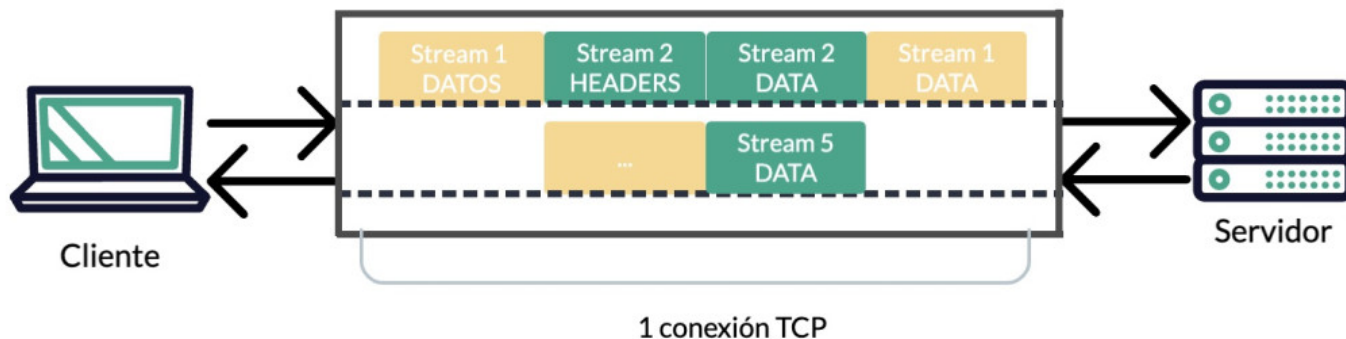
HTTP/1.1 procesa cada solicitud y respuesta secuencialmente, lo que genera retrasos potenciales. Además, cada solicitud requiere una nueva conexión TCP al servidor, causando tráfico de red innecesario por las múltiples conexiones TCP. Además, con cada solicitud HTTP 1.1, los encabezados se envían junto con la carga útil, lo que resulta en transferencias de datos redundantes.

Estas limitaciones llevaron a la necesidad de un protocolo más avanzado y optimizado. Estandarizado en mayo de 2015, HTTP/2 fue adoptado como una revisión importante del protocolo HTTP que incorpora muchas mejoras introducidas en SPDY.

## Conexión HTTP 1.1



## Conexión HTTP 2.0



Es un protocolo binario que permite la multiplexación de múltiples solicitudes en paralelo a través de una sola conexión TCP, lo que resulta en menos conexiones TCP en general. HTTP/2 también introduce un

mecanismo de compresión de campo de encabezado, tecnología push y otras optimizaciones para mejorar el rendimiento, la seguridad y la eficiencia de la comunicación web.

## ¿Cómo HTTP/2 Funciona con HTTPS?

HTTPS se utiliza para establecer una red ultra-segura conectando computadoras, servidores y máquinas para procesar información confidencial de empresas y consumidores. Procesar las transacciones financieras de los bancos y de las instituciones de salud, mantener registros de pacientes son el objetivo principal de delitos cibernéticos criminales. HTTPS funciona como una capa eficaz contra amenazas a la ciberdelincuencia persistentes, aunque no es la única implementación de seguridad utilizada para evitar ataques sofisticados cibernéticos vulnerando redes corporativas de alto valor.



El soporte del navegador HTTP/2 incluye cifrado HTTPS y realmente complementa el rendimiento general de la seguridad de las implementaciones de HTTPS. Características como menos apretones de manos de TLS, bajo consumo de recursos tanto en el lado del cliente como en el servidor y capacidades mejoradas en la reutilización de sesiones web existente mientras elimina las vulnerabilidades relacionadas con HTTP1.x presentan a HTTP/2 como un factor clave para asegurar la comunicación digital sensible en entornos de red.



HTTPS no está limitado a organizaciones de alto perfil y la seguridad cibernética es igualmente valiosa para los dueños de negocios en línea, blogueros casuales, comercio electrónico, comerciantes e incluso los usuarios de medios sociales. El HTTP/2 exige intrínsecamente la más reciente y segura versión de TLS y todas las comunidades en línea, propietarios de negocios y webmasters deben asegurarse de que sus sitios web utilicen HTTPS de forma predeterminada.

Procesos habituales para configurar HTTPS incluyen el uso de planes de alojamiento web, comprar, activar e instalar un certificado de seguridad y, finalmente, la actualización de la página web para que utilice HTTPS.

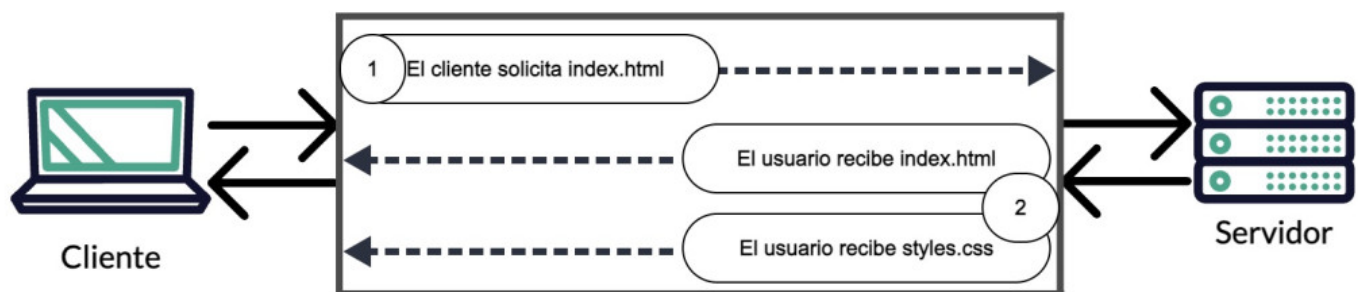
## Características principales de HTTP/2

HTTP/2 introduce varias características que abordan las limitaciones de HTTP/1.1 y mejoran el rendimiento de la comunicación web.

### Server Push

Una de las características notables de HTTP/2 es el servidor push. Con él, el servidor puede enviar recursos de forma proactiva para satisfacer las futuras solicitudes del cliente. Esta función reduce el número de viajes de ida y vuelta necesarios entre ambos, el cliente y el servidor, lo que resulta en una velocidad de carga de la página más rápida.

## Envío de servidor HTTP 2.0



Ante una sola solicitud del cliente, el servidor inserta 2 activos: el index.html junto con styles.css. Gracias a ese impulso preventivo de los activos del sitio web, HTTP/2 facilita y acelera el procesamiento de la página.

### Multiplexación

HTTP/2 usa multiplexación para permitir múltiples solicitudes y respuestas concurrentes a través de una sola conexión TCP. A diferencia de HTTP/1.1, donde cada solicitud tenía que esperar una respuesta antes de que se pudiera enviar la siguiente, HTTP/2 habilita el paralelismo. Mejorando así la eficiencia global.



Con la multiplexación, las solicitudes y respuestas se dividen en unidades más pequeñas llamadas tramas. Estas tramas se intercalan y envían sin bloqueo. Esto significa que si una solicitud se retrasa debido a la congestión de la red, otras solicitudes pueden continuar siendo procesadas.

La multiplexación también elimina el problema del bloqueo de cabecera , donde una solicitud lenta puede retrasar la entrega de solicitudes HTTP posteriores. Como resultado, esta función ayuda a reducir la latencia y promueve una mayor satisfacción del cliente.

## Control de flujo y priorización de flujo

HTTP/2 introduce el concepto de priorización de flujo, que permite al cliente asignar niveles de prioridad a diferentes recursos. Esto, junto con el control de flujo, asegura que los recursos críticos se recuperen primero, optimizando el renderizado de las páginas web y proporcionando una mejor experiencia de usuario.

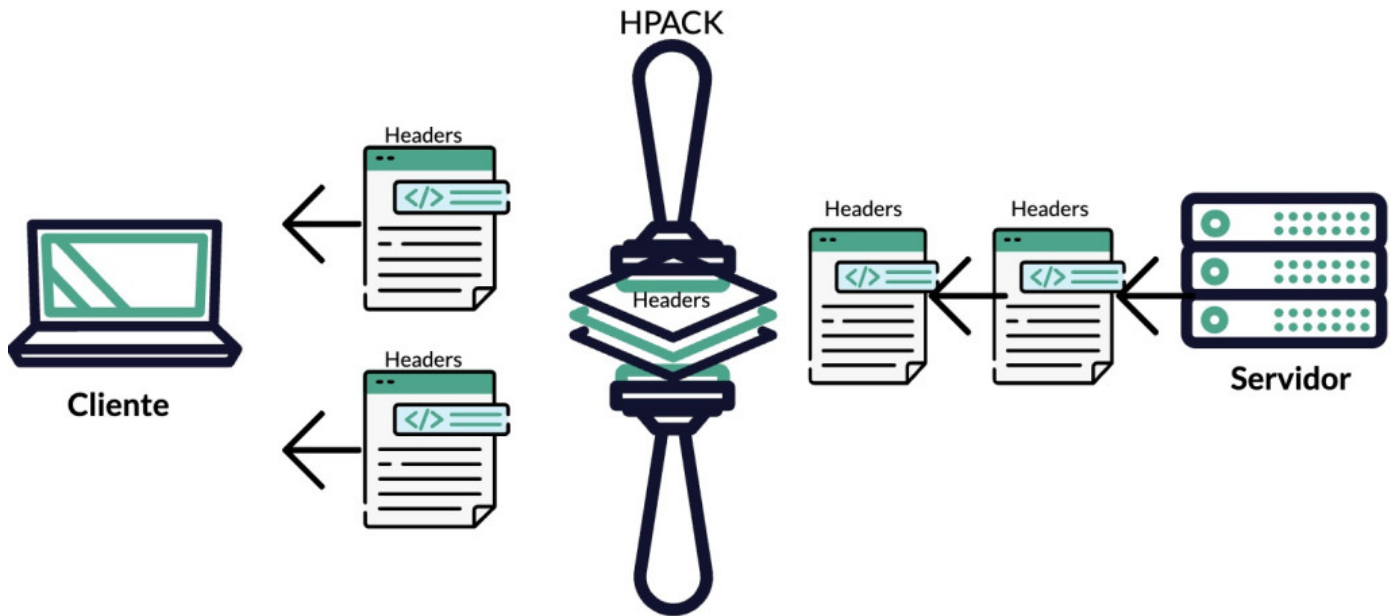
El control de flujo permite al receptor regular la tasa de transmisión de datos priorizando flujos de datos específicos. Evita sobrecargar el receptor con más datos de los que puede soportar, evitando la congestión y el desbordamiento del búfer. El control de flujo funciona a nivel de la corriente, lo que permite que cada corriente tenga su propia ventana de control de flujo.

La interacción entre el control de flujo y la priorización de la transmisión es crucial para optimizar el rendimiento general del protocolo HTTP/2. El control de flujo asegura que los datos se transmitan a una velocidad que el receptor puede manejar, mientras que la priorización de los flujos determina el orden en el que se procesan los flujos, lo que permite que los recursos críticos se entreguen de manera más eficiente.

## Compresión de encabezado

HTTP/2 utiliza un mecanismo de compresión de encabezados para reducir la sobrecarga asociada con los encabezados HTTP . Esta técnica de compresión reduce el tamaño de los encabezados, lo que resulta en un menor consumo de ancho de banda y una comunicación más rápida entre el cliente y el servidor.

# Compresión de encabezado HTTP/2



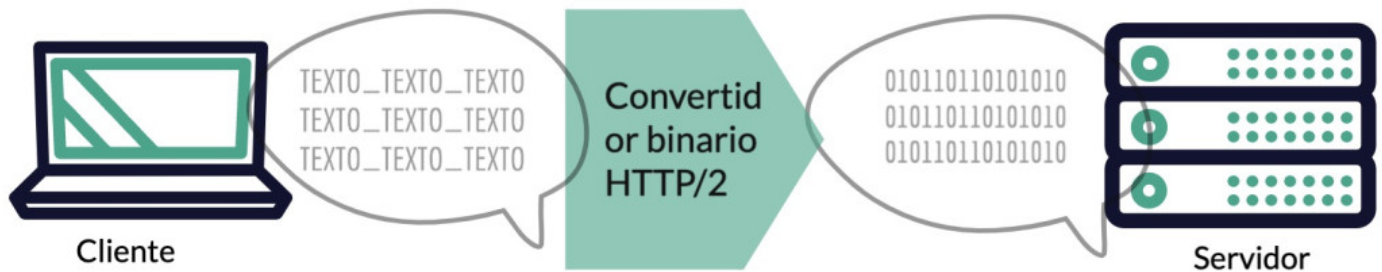
El mecanismo usa el algoritmo HPACK (HTTP/2 Static Table and Huffman Encoding) . HPACK utiliza técnicas como la codificación, indexación y administración dinámica de tablas de Huffman para minimizar el tamaño de los encabezados sin perder información crítica.

## Protocolo binario

Mientras que HTTP/1.1 usaba texto plano para la comunicación, HTTP v2 emplea un protocolo binario, lo que permite un análisis más eficiente y una transmisión de datos más rápida . El formato binario reduce la cantidad de datos que se deben transmitir, mejorando el rendimiento.

En HTTP/2, los datos se dividen en unidades más pequeñas llamadas tramas, que son paquetes binarios que llevan tipos específicos de información. Cada fotograma tiene un formato específico, incluido un encabezado de fotograma que indica su tipo y longitud. Estas tramas se envían luego a través de una sola conexión cliente-servidor, lo que permite la comunicación concurrente e intercalada.

# Protocolo binario HTTP2



## Beneficios de rendimiento de HTTP/2

La adopción de HTTP/2 trae beneficios de rendimiento significativos. La combinación de características como la inserción en el servidor, la multiplexación, la priorización de la transmisión y la compresión del campo del encabezado da como resultado tiempos de carga de la página más rápidos, una latencia reducida y un rendimiento web mejorado. Los sitios web y aplicaciones que usan HTTP/2 a menudo experimentan una mejor participación del usuario y mayores tasas de conversión.

## Compatibilidad y adopción

HTTP/2 está diseñado para ser compatible con HTTP/1.1, lo que garantiza que la infraestructura web existente pueda realizar la transición sin problemas al último protocolo. La mayoría de los navegadores y servidores modernos son compatibles con HTTP/2, lo que permite a los propietarios y desarrolladores de sitios web aprovechar sus beneficios sin mayores problemas de compatibilidad.

## Implementación y soporte

Para habilitar HTTP/2 en un sitio web, se requiere configuración del lado del servidor. Los pasos específicos para la implementación dependen del servidor web que se utilice. Los servidores web populares como Apache y Nginx tienen soporte integrado para HTTP v2.

HTTP/2 incluye un mecanismo de respaldo llamado HTTP/2 sobre TLS (HTTP/2 con TLS). Esto permite a los clientes que no soportan HTTP/2 recurrir a HTTP/1.1 cuando se comunican con servidores que soportan ambos protocolos.

Además, la adopción de HTTP/2 puede requerir algunos ajustes y consideraciones. Por ejemplo, el requisito de cifrado de HTTP/2 (seguridad de la capa de transporte) puede requerir que los administradores del servidor obtengan e instalen Certificados SSL/TLS .

Por último, pero no menos importante, también hay soporte CDN para HTTP/2. Content Delivery Networks (CDN) también han adoptado HTTP/2, lo que permite la entrega de contenido a través del protocolo. Los CDN juegan un papel crucial en acelerar la adopción de HTTP/2 al asegurar que el contenido en caché y distribuido se sirva utilizando los últimos protocolos.

## Limitaciones de HTTP/2

Si bien HTTP/2 ofrece numerosas ventajas, es esencial tener en cuenta sus limitaciones. Aquí describiremos algunos de ellos, así que sigue leyendo.

- **Soporte de navegador limitado.** Es posible que algunos sistemas heredados y navegadores más antiguos no sean totalmente compatibles con HTTP/2, lo que podría causar problemas de compatibilidad.
- **Complejidad de la actualización.** El proceso de actualización puede requerir cambios en la configuración del servidor, la implementación de nuevas características y consideraciones de compatibilidad con versiones anteriores.
- **Latencia e inicio lento.** El protocolo puede enfrentar desafíos en conexiones de red lentas y entornos de alta latencia. La configuración de la conexión inicial (también conocida como la fase de inicio lento ) puede introducir latencia al establecer nuevas conexiones.
- **Bloqueo de cabecera.** En HTTP/2, se pueden enviar múltiples solicitudes y respuestas simultáneamente a través de una sola conexión. Sin embargo, si una solicitud se retrasa o se bloquea debido a una respuesta lenta del servidor, puede llevar a un bloqueo de cabecera.
- **Mayor uso de recursos del servidor.** HTTP/2 requiere más recursos del servidor en comparación con HTTP/1.1 debido a la mayor complejidad de manejar flujos multiplexados y administrar solicitudes concurrentes. En ciertos escenarios, esto puede resultar en una mayor utilización de recursos del servidor, especialmente cuando se atienden un gran número de solicitudes simultáneas.

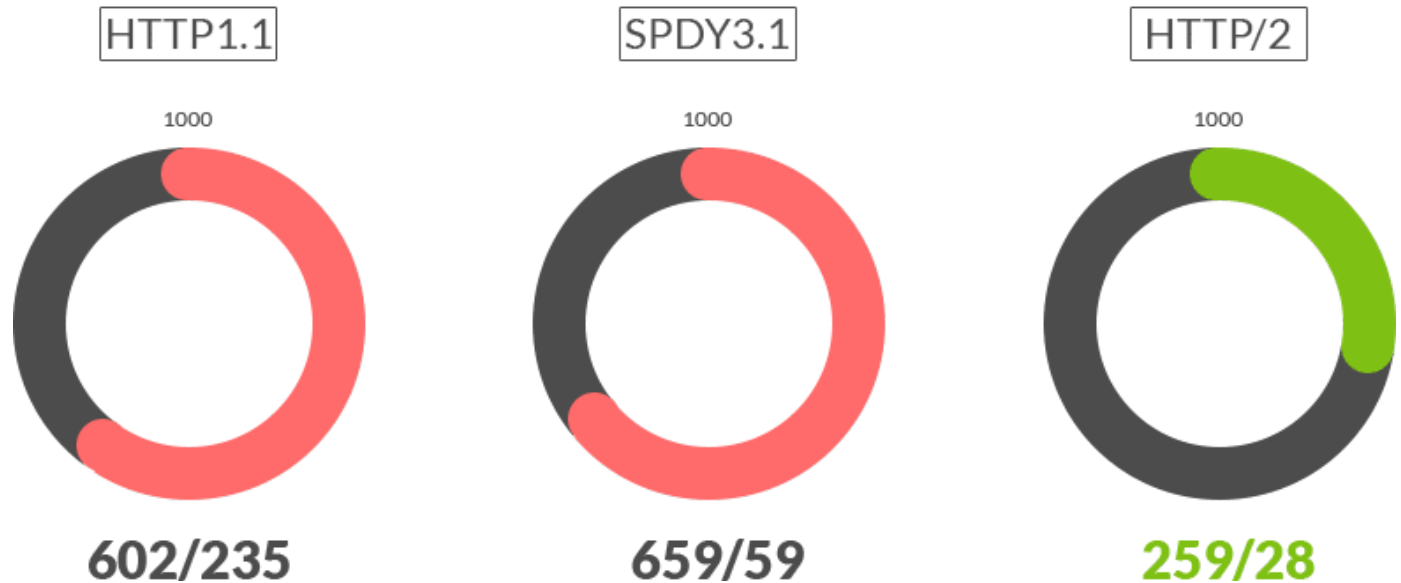
Además, ciertas configuraciones de red y proxies pueden no manejar el tráfico HTTP/2 correctamente, lo que lleva a un rendimiento degradado.

A pesar de estas limitaciones, los beneficios de HTTP/2 superan estos inconvenientes en la mayoría de los casos. Es importante que los propietarios y desarrolladores de sitios web consideren estas limitaciones y aseguren la implementación y las pruebas adecuadas al adoptar HTTP/2 para sus sitios web.

# Análisis Comparativo de Rendimiento de HTTPS, SPDY y HTTP/2

Las siguientes comparaciones entre el rendimiento de puntos de referencia de HTTPS, SPDY y HTTP/2 pintan una imagen clara mejoras en el rendimiento de la web con el último protocolo de aplicación.

## REQUEST SENT / RECEIVED



Los resultados de puntos de referencia HTTP/2 confirman las ideas que la compresión de encabezados, server push y otros mecanismos utilizados específicamente para mejorar la velocidad de la página y la experiencia del usuario entregan sistemáticamente en el mundo real:

Detalles de la prueba: Esta prueba comparando HTTPS, SPDY3.1 y HTTP/2 presenta los siguientes resultados:

- El tamaño de la solicitud del cliente y los encabezados de respuesta del servidor: Los puntos de referencia HTTP/2 demuestran cómo el uso del mecanismo de compresión del encabezado reducen significativamente el tamaño del encabezado, mientras que SPDY sólo encoge el encabezado utilizado en la respuesta del servidor para esta solicitud en particular. HTTPS no reduce el tamaño del encabezado, tanto en la solicitud como en la respuesta de comandos.
- Tamaño del mensaje de respuesta del servidor: Aunque la respuesta del servidor HTTP/2 fue mayor en tamaño, proporciona un cifrado más sólido para la mejora de la seguridad como una de las principales desventajas.
- Número de conexiones TCP utilizadas: HTTP/2 y SPDY utilizan menos recursos de red mediante el procesamiento de varias solicitudes simultáneas (multiplexación) y por lo tanto reduciendo la latencia.

- La velocidad de carga de la página: HTTP fue consistentemente más rápida que SPDY. HTTPS fue significativamente más lento debido a la falta de compresión de encabezados y capacidades de inserción de servidor.

## Configuración básica en Apache

Cuando tiene un `httpd` compilado con `mod_http2` necesita una configuración básica para activarlo. Lo primero, como con cualquier otro módulo de Apache, es que necesita cargarlo:

```
LoadModule http2_module modules/mod_http2.so
```

La segunda directiva que necesita añadir a la configuración de su servidor es:

```
Protocols h2 http/1.1
```

Esto permite `h2`, la variante segura, para ser el protocolo preferido de las conexiones en su servidor. Cuando quiera habilitar todas las variantes de HTTP/2, entonces simplemente configure:

```
Protocols h2 h2c http/1.1
```

Dependiendo de dónde pone esta directiva, afecta a todas las conexiones o solo a las de ciertos host virtuales. La puede anidar, como en:

```
Protocols http/1.1
<VirtualHost ...>
    ServerName test.example.org
    Protocols h2 http/1.1
</VirtualHost>
```

Esto solo permite HTTP/1, excepto conexiones SSL hacia `test.example.org` que ofrecen HTTP/2.

El orden de los protocolos mencionados también es relevante. Por defecto, el primero es el protocolo preferido. Cuando un cliente ofrece múltiples opciones, la que esté más a la izquierda será la escogida. En

```
Protocols http/1.1 h2
```

el protocolo preferido es HTTP/1 y siempre será seleccionado a menos que el cliente sólo soporte `h2`. Puesto que queremos hablar HTTP/2 con clientes que lo soporten, el orden correcto es:

```
Protocols h2 h2c http/1.1
```

Hay algo más respecto al orden: el cliente también tiene sus propias preferencias. Si quiere, puede configurar su servidor para seleccionar el protocolo preferido por el cliente:

```
ProtocolsHonorOrder Off
```

Hace que el orden en que usted escribió los `Protocols` sea irrelevante y sólo el orden de preferencia del cliente será decisivo.

# Bibliografía

1. Guía HTTP/2. (s/f). Apache.org. Recuperado el 14 de noviembre de 2023, de <https://httpd.apache.org/docs/2.4/es/howto/http2.html>
2. Hosting, S. W. (s/f). ¿Qué es HTTP/2? (Fundamentos, Evolución, Características y Beneficios). Recursos de conocimiento de SiteGround. Recuperado el 14 de noviembre de 2023, de <https://es.siteground.com/kb/que-es-http2/>
3. Qué es HTTP/2 – La Guía Definitiva. (2016, abril 28). Kinsta®; Kinsta. <https://kinsta.com/es/aprender/que-es-http2/>