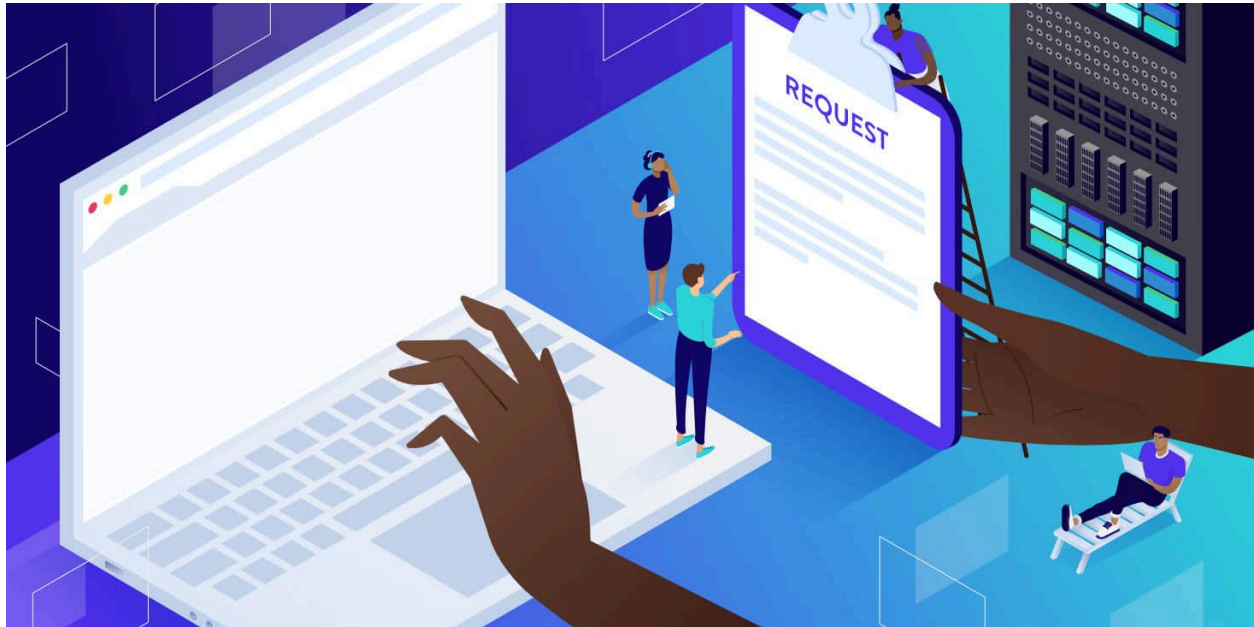


# HTTP



HTTP

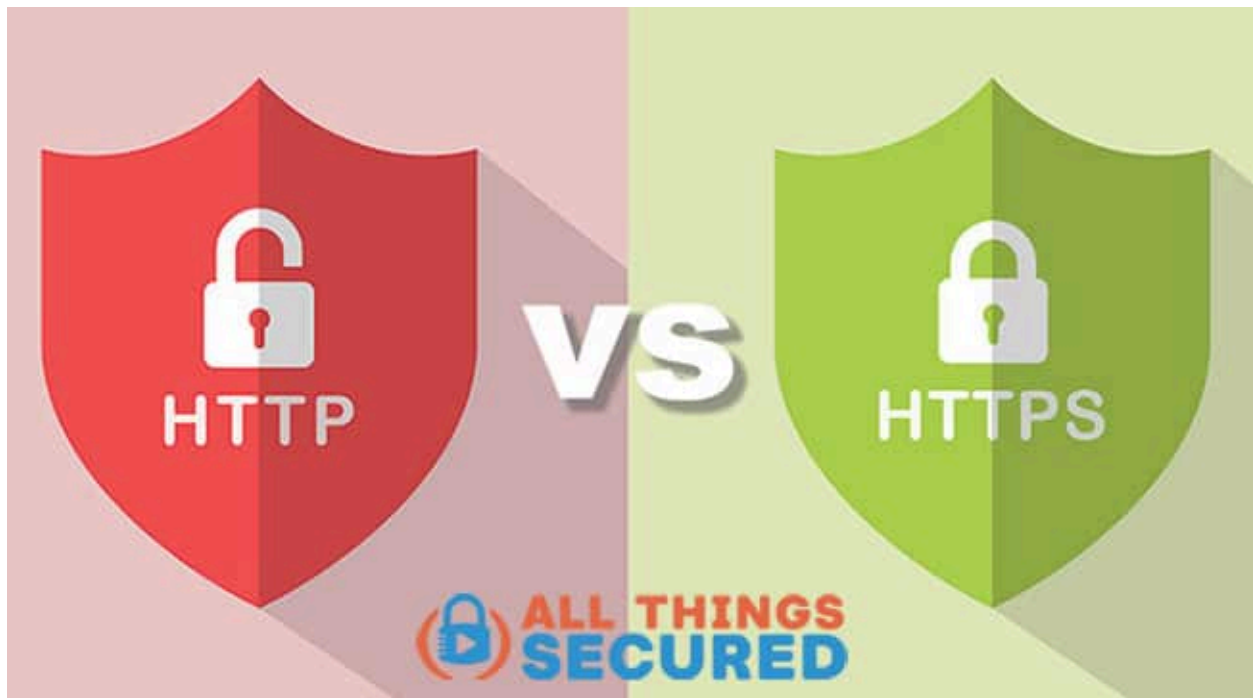
## ¿Qué es HTTP?

Creado en 1991 por Tim Berners-Lee, **HTTP** puede ser descrito como un conjunto de reglas las cuales permiten que distintos dispositivos logren comunicarse entre si. Esto es un protocolo, y más específicamente **HTTP** se enfoca en la comunicación entre dispositivos a través de la World Wide Web (Comúnmente conocida como la Web) con el modelo Cliente-Servidor.

Es el protocolo estándar que utilizan los navegadores y servidores web para lograr un intercambio de datos entre dispositivos, estos datos son comúnmente imágenes, archivos, documentos, páginas web, entre otros.

**HTTP** está dividido en versiones que han ido cambiando a lo largo de los años, comenzando por **HTTP 0.9**, hasta llegar a **HTTP 3**, la cual es la versión utilizada actualmente por la mayoría de los navegadores.

Existe una variante conocida como **HTTPS**, que en esencia es lo mismo, pero con la diferencia de que **HTTPS** incluye un cifrado que permite que los datos no sean robados. Cada uno de estos ocupa un puerto que comúnmente es 80 para **HTTP** y 443 para **HTTPS**.



https-http

## ¿Cual es la estructura de una petición?

La petición es el mensaje que envía el cliente hacia el servidor en forma de solicitud, para así obtener los datos que se requieran.

La estructura de una petición se compone de tres partes esenciales, las cuales son:

### Línea de petición

Es la primera línea que se coloca en la petición y su objetivo es indicar lo que se va a hacer, esta tiene tres elementos.

<MÉTODO> <RUTA/URL> <VERSIÓN DE HTTP>

Ya estructurado:

```
GET /index.html HTTP/1.1
```

### Encabezados

Son elementos que dan datos adicionales sobre la petición que se quiere realizar como el Host, el tipo de contenido, la autenticación, el agente, entre otros.

### Cuerpo

En este último se incluyen los métodos que enviarán los datos, pueden ser en distintos formatos de archivos, como los JSON. Algunos ejemplos de métodos que se utilizan son POST, PUT y PATCH.

## ¿Cuales son los métodos mas usados?

Los métodos son las acciones que quiere realizar el cliente respecto al servidor al cual envia la solicitud con el fin de recibir los datos necesarios. Existen varios métodos para esto, los más comunes son:



## GET

Consta de pedir algo al servidor, este no modifica el servidor, solo busca datos. No contienen un cuerpo de mensaje y los parámetros que se requieren se envían mediante la consulta en la URL.



## POST

Este busca actualizar los datos en el servidor, esto con el fin de realizar un cambio en los datos, como actualizar o crear algún recurso. Los datos de este si se incluyen en el cuerpo de la solicitud.



## PUT

Al igual que POST, sirve para crear algún recurso, pero se diferencian en que PUT tiene la posibilidad de reemplazar al recurso dado en los datos de la solicitud.

Su principal característica es que por más veces que mandes el put, no habrá diferencia entre la primera vez que lo enviaste y la última, esto suponiendo que no hayas cambiado la estructura de este.



## PATCH

Sirve para aplicar cambios a un recurso existente en el servidor, esto siendo modificaciones parciales, básicamente solo manda los cambios especificados en vez de reemplazar completamente el recurso.



## DELETE

Su función principal es borrar algún recurso del servidor, deshaciéndose completamente de él, este recurso se suele especificar en la URL y el código de respuesta del servidor indica si se eliminó el recurso.

## ¿Cuáles son los códigos de estado?

Existen distintos códigos de estado, los cuales son formas en las que el servidor se comunica con el cliente para indicar un resultado en la solicitud. Generalmente se representan por el primer dígito siendo un número del 1 al 5 y seguido dos dígitos determinados por el contexto de la situación, estos son:

### 1xx Informativos

- 100 **Continue** - el servidor recibió la cabecera, el cliente puede enviar el cuerpo.
- 101 **Switching Protocols** - se cambió de protocolo.

### 2xx Exitosos

- 200 OK - petición exitosa.
- 201 Created - recurso creado correctamente.
- 202 Accepted - aceptado, pero aún en proceso.
- 204 No Content - petición exitosa pero sin contenido que devolver.

### 3xx Redirecciones

- 301 Moved Permanently - recurso movido de forma permanente.
- 302 Found - recurso movido temporalmente.

- 304 Not Modified - el recurso no ha cambiado.

## 4xx Errores en el cliente

- 400 Bad Request - error en la sintaxis de la petición.
- 401 Unauthorized - no autenticado.
- 403 Forbidden - autenticado pero sin permisos.
- 404 Not Found - recurso no encontrado.
- 405 Method Not Allowed - método HTTP no permitido en ese recurso.
- 429 Too Many Requests - demasiadas peticiones.

## 5xx Errores en el servidor

- 500 Internal Server Error - error genérico del servidor.
- 501 Not Implemented - el servidor no soporta ese método.
- 502 Bad Gateway - un servidor intermedio recibió respuesta inválida.
- 503 Service Unavailable - el servicio no está disponible.
- 504 Gateway Timeout - el servidor intermedio no recibió respuesta a tiempo.

## ¿Qué son los encabezados?

Estos son una línea de información adicional que se envía junto con la petición o la respuesta. Sirve para dar contexto sobre la comunicación entre cliente y servidor.

Sus principales características es que son pares de clave: valor, se colocan después de la línea de petición o la línea de estado y se separan del cuerpo.

Hay distintos tipos de encabezados como encabezado general, que es de información sin relación con los datos del cuerpo; encabezado de consulta, que es para información adicional sobre el cliente; encabezado de respuesta para información adicional sobre el origen; y encabezado de entidad para información sobre el cuerpo de la entidad.

## ¿Qué es y cuál es la Estructura de una URL?

Las siglas URL significan Uniform Resource Locator, la cual es una secuencia de caracteres cuyo objetivo es localizar datos en internet. Esta estructurado por siete elementos, los cuales veremos a continuación.

### Protocolo

El protocolo es la parte inicial de la URL que indica el método de comunicación que se utilizará entre el navegador y el servidor.

Define cómo se transmiten los datos y garantiza la manera en que se establece la conexión. Entre los más comunes se encuentran los que aseguran la transferencia de información de forma segura o no segura.

### Subdominio

El subdominio es un elemento que aparece antes del dominio principal y sirve para organizar o separar secciones de un sitio web. Se utiliza comúnmente para diferenciar servicios, áreas específicas o entornos dentro de la misma página sin necesidad de adquirir un dominio distinto.

### Dominio

El dominio es el nombre principal de la URL y representa la identidad del sitio web en internet. Es único y permite que los usuarios lo recuerden fácilmente en lugar de usar la dirección numérica del servidor. Constituye la parte central

que da reconocimiento a la página.

## Dominio de primer nivel

El dominio de primer nivel, también conocido como TLD, es la extensión que sigue inmediatamente al dominio principal. Está regulado y puede tener fines generales, geográficos o temáticos. Su función es clasificar el sitio en un contexto más amplio dentro del sistema de nombres de dominio.

## Subcarpeta

La subcarpeta se encuentra después del dominio y actúa como una ruta interna que organiza los archivos o secciones de un sitio. Representa una jerarquía en la estructura del servidor y permite que las páginas estén ordenadas por categorías o directorios.

## Slug

El slug corresponde a la parte de la URL que identifica de manera específica una página o recurso dentro de un sitio. Generalmente se forma por palabras legibles que describen el contenido al que se accede, facilitando tanto su comprensión como su posicionamiento en motores de búsqueda.

## Parámetros de la URL

Los parámetros de la URL son fragmentos que se añaden al final de esta con el objetivo de transmitir información adicional. Normalmente se emplean para filtrar, personalizar, rastrear o modificar el contenido que se mostrará. Son dinámicos y permiten mayor flexibilidad en la interacción con el servidor.