

# Introducción a HTTP

DigitalHouse >  
Coding School



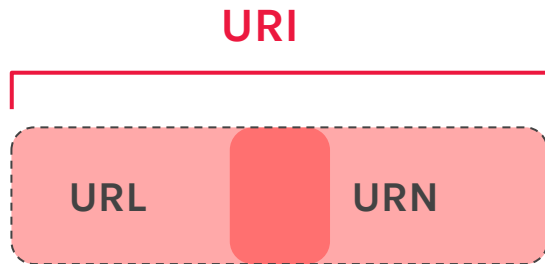
**HTTP** (*Hyper Text Transfer Protocol*)  
es el **protocolo** que gestiona las  
**transacciones** web entre clientes  
y servidores.



# ¿Qué es una **URI**?

El protocolo HTTP permite la transferencia de información en la web a través de **direcciones web**, técnicamente llamadas **URI**.

Una **URI** (identificador de recursos uniformes) es un bloque de texto que se escribe en la barra de direcciones de un navegador web y está compuesto por dos partes: la **URL** y la **URN**.



# Componentes de una URI

## URL

Indica **dónde** se encuentra el recurso que deseamos obtener y siempre comienza con un **protocolo**. En este caso HTTP.

The diagram shows the URL 'http://www.digitalhouse.com/preguntas-frecuentes'. An orange curly bracket is positioned above the 'http://' part. A teal curly bracket is positioned below the 'www.digitalhouse.com/preguntas-frecuentes' part.

**http://**www.digitalhouse.com/preguntas-frecuentes

## URN

Es el **nombre exacto** del recurso uniforme. El nombre del dominio y, en ocasiones, el nombre del recurso.

“

Dentro de esta estructura de comunicación, hablamos de ***request*** cada vez que el cliente le solicita un recurso al servidor, y de ***response*** cada vez que el servidor le devuelve una respuesta al cliente.

”

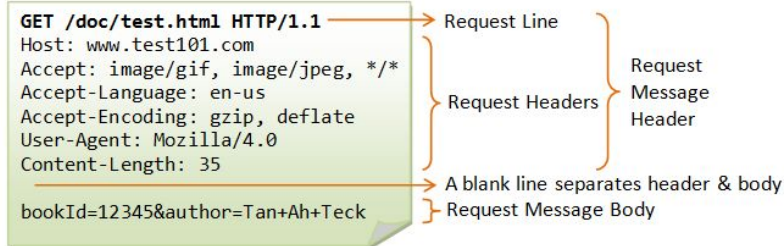


# Cómo viaja la información

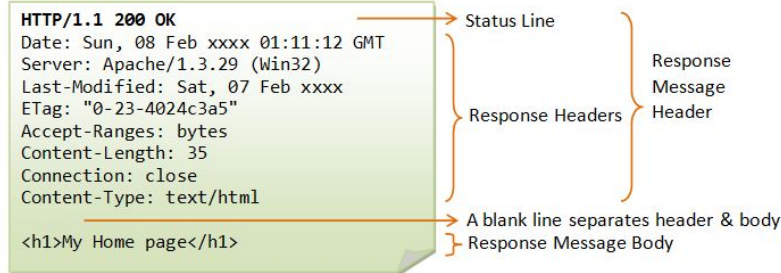
Cada vez que hacemos un pedido o recibimos una respuesta utilizando el protocolo HTTP, se envía información importante sobre ellos.

La información viaja a través de los **headers** o **cabeceras**, que son **porciones de texto** conteniendo la información requerida por el cliente o el servidor.

## PEDIDO / REQUEST



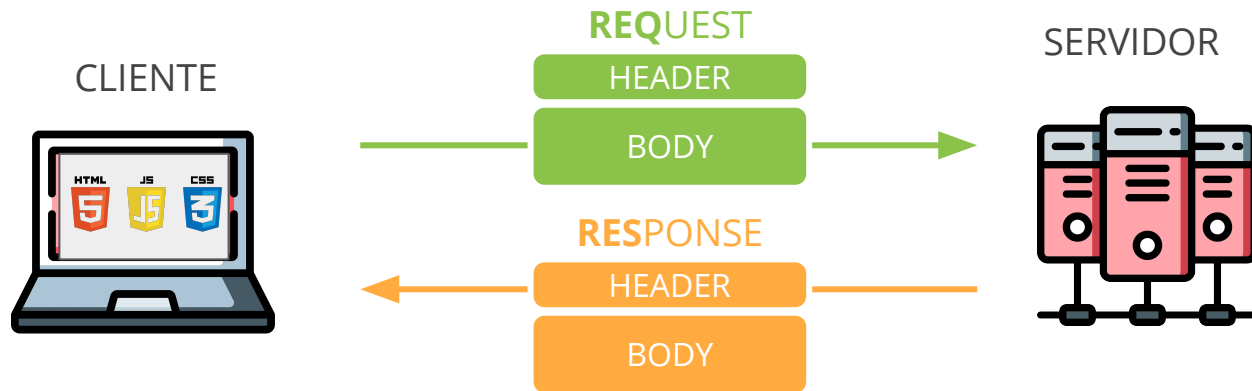
## RESPUESTA / RESPONSE



# Cómo viaja la información

Además de las cabeceras, tenemos el **body** o **cuerpo** del mensaje, que es donde estará la información en sí del pedido o su respuesta.

En el caso del pedido, puede ser el contenido de un formulario. En el caso de una respuesta, puede ser el código de la página que solicitamos.



# Métodos de petición

El protocolo HTTP define **métodos de petición**. Cada método representa una **acción** y, si bien comparten algunas características, implementan funcionalidades diferentes entre sí. Los métodos más utilizados por este protocolo son:

## GET

Se utiliza para pedirle información al servidor de un recurso específico. Cada vez que escribimos una dirección en el navegador o accedemos a un enlace, estamos utilizando el método GET. En caso de querer enviar información al servidor usando este método, la misma viajará a través de la URL.



# Métodos de petición

## **POST**

Se utiliza para enviar datos al servidor. Este método es más seguro que GET ya que la información no viaja a través de la URL.

## **DELETE**

Borra un recurso presente en el servidor. Cuando eliminamos un posteo en Facebook, por ejemplo, estamos utilizando este método.

# Métodos de petición

## PUT

Es muy parecido a POST. Se usa para reemplazar toda la información actual de un recurso presente en el servidor.

## PATCH

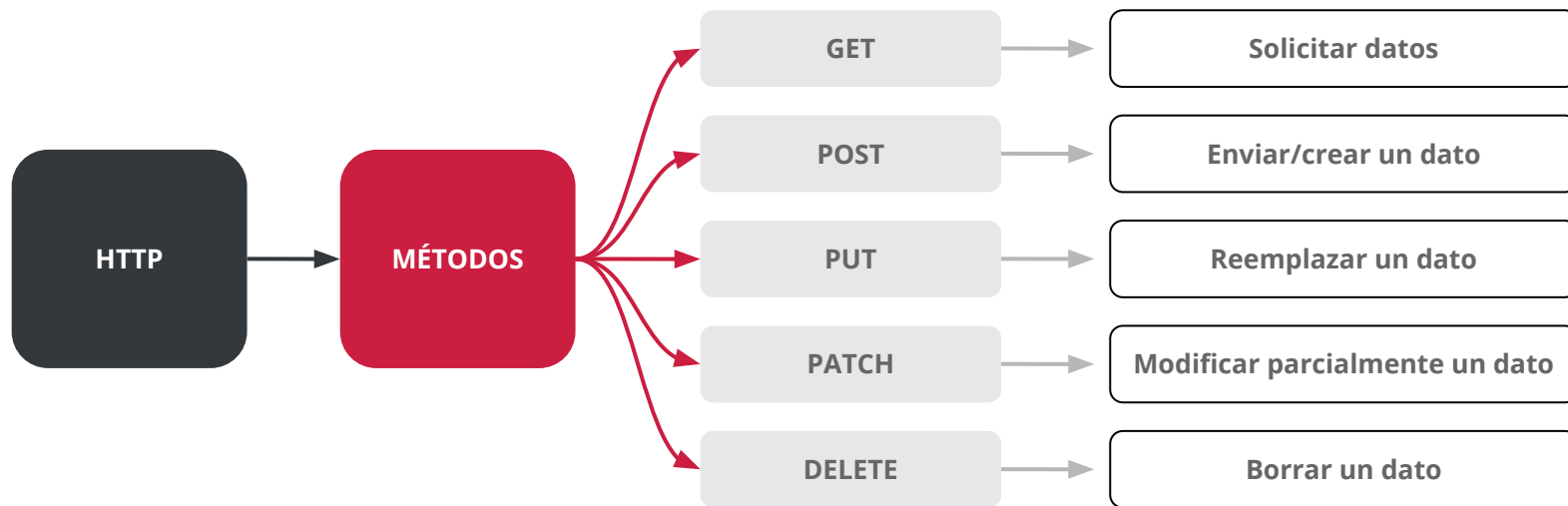
Similar a PUT. Es utilizado para aplicar modificaciones parciales a un recurso en el servidor.



PUT y PATCH suelen ser lo mismo. Elegir entre uno y otro va a depender del contexto y lo que queramos implementar en nuestra aplicación. Al editar un posteo o un perfil estaremos usando alguno de estos dos métodos.

# Métodos de petición

En resumen, tenemos 5 métodos y cada uno de ellos tiene un propósito.



# Códigos de estado HTTP

Cada vez que el **servidor** recibe una petición o **request**, este emite un código de estado que indica, de forma abreviada, el **estado** de la respuesta HTTP. El código tiene tres dígitos. El primero representa uno de los 5 tipos de respuesta posibles:

- **1\_\_** Respuestas informativas
- **2\_\_** Respuestas exitosas
- **3\_\_** Redirecciones
- **4\_\_** Errores del cliente
- **5\_\_** Errores de servidor

# Códigos de estado HTTP

Algunos de los códigos más usados son:

- **200:** *OK* → La petición se realizó con éxito.
- **301:** *Moved Permanently* → El recurso se ha movido.
- **302:** *Found* → El recurso fue encontrado.
- **304:** *Not Modified* → El recurso no cambió, se cargará desde el caché.
- **400:** *Bad Request* → El pedido está mal.
- **401:** *Unauthorized* → No estás autorizado, seguramente debes autenticarte.
- **403:** *Forbidden* → El pedido está prohibido y no debería repetirse.
- **404:** *Not Found* → El recurso no fue encontrado.
- **500:** *Internal Server Error* → Hubo un error en el servidor.
- **503:** *Service Unavailable* → El servicio solicitado no está disponible.
- **550:** *Permission denied* → Permiso denegado.



**HTTPS** es un protocolo mejorado de **HTTP**. Usando este protocolo, el servidor **codifica** la sesión con un **certificado digital**.



DigitalHouse>  
Coding School