

DESARROLLO WEB FULL STACK

---

# Introducción a HTTP

Comunicando clientes con servidores

DWFS COR



# HTTP

Creado por **Tim Berners-Lee** en 1989, es el **protocolo** utilizado para acceder y publicar en la Web. Significa **Protocolo de Transmisión de Hipertexto**.

Actualmente el World Wide Web Consortium (W3C) es la entidad encargada de publicar la especificación del protocolo HTTP, entre otras cosas.

<https://www.w3.org/Protocols/>



# HTTP

HTTP está basado en una **arquitectura cliente-servidor** en la que se intercambian peticiones (requests) por parte del cliente y respuestas (responses) por parte del servidor.

## Características:

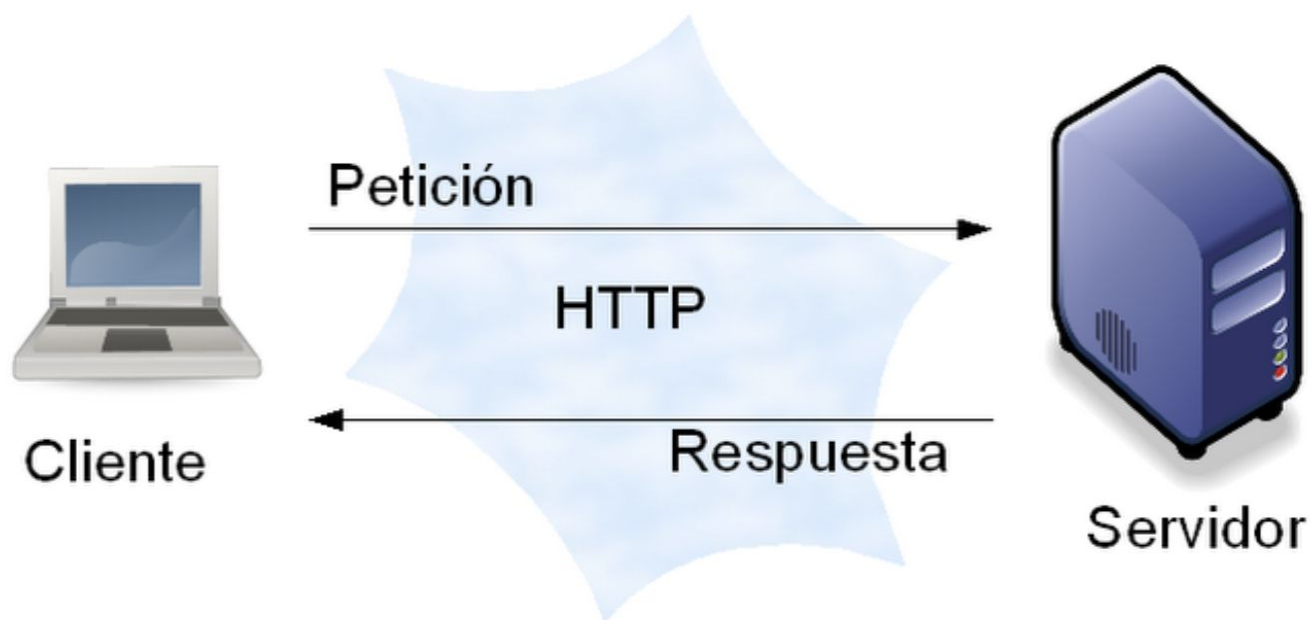
- **Sin estado.** Es decir, que cada una de las transacciones request/response que se realizan no afectan al estado del cliente o del servidor.
- **Independiente del contenido.** Aún cuando es muy común que un servidor HTTP entregue documentos HTML, pero no existe restricción en el tipo de recurso al que se pueda acceder.
- **Sin conexión.** Una vez que la transacción request/response es terminada, la conexión entre cliente y servidor es destruida.



# ¿Qué es un protocolo?

Es un conjunto de reglas usadas por computadoras para comunicarse a través de una red.





# Uniform Resource Identifiers (URI)

Como su nombre lo indica, los Identificadores Recursos Uniformes siempre apuntan hacia un recurso al que se puede acceder.

## **Sintaxis:**

```
"http:" "://" host [ ":" port ] [ abs_path [ "?" query ] ]
```



# Peticiones, respuestas y sesiones

Las comunicaciones entre el cliente y el servidor consisten en un serie de **intercambios de datos**.

- Un cliente por lo general envía una petición (request) a un servidor atendiendo a una dirección específica.
- El servidor recibe la petición y procesa los datos. Dependiendo de la petición, el servidor puede enviar distintos mensajes que incluyen un estado específico.
- A este intercambio de peticiones y respuestas entre un cliente y un servidor se conocen como sesión.



# Métodos HTTP

El protocolo HTTP define **métodos o "verbos"**, los cuales permiten realizar peticiones específicas entre un cliente y un servidor. Algunos de los métodos más utilizados son:





## Métodos HTTP

- **GET:** Es el método más usado y nos permite obtener información desde un server http

**Por ejemplo:** al entrar desde un navegador a

<https://www.google.com/>

<https://api.mercadolibre.com/currencias/ARS>

automaticamente hacemos un GET. Por Postman también lo podemos hacer. Probemos.

**Otro ejemplo:**

[https://api.mercadolibre.com/sites/MLA/search?nickname=nuestro\\_usuario\\_de\\_mercadolibre](https://api.mercadolibre.com/sites/MLA/search?nickname=nuestro_usuario_de_mercadolibre)



## Métodos HTTP

- **POST:** Nos permite enviarle información al server para guardar un nuevo recurso.

Ejemplo usando POSTMAN



## Métodos HTTP

- **PUT:** Nos permite modificar un recurso ya existente.

Ejemplo usando POSTMAN



## Métodos HTTP

- **DELETE:** Es el verbo que le comunica a la aplicación que queremos borrar un recurso.

Ejemplo usando POSTMAN



# Mensajes de estado

Los **mensajes de estado son respuestas de un servidor con respecto a una consulta o búsqueda de recursos**. Está conformado por un número entero de 3 dígitos.

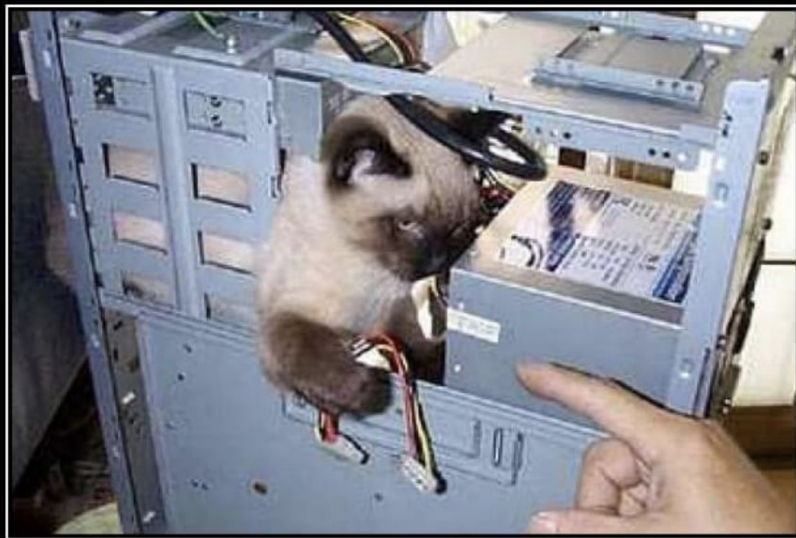
Quizás el mensaje de estado más conocido es el 404 que se emite cuando un recurso no es encontrado.

## Tipos por el número inicial:

- 1xx Información.
- 2xx Éxito.
- 3xx Redireccionamiento.
- 4xx Error del cliente.
- 5xx Error del servidor.



<https://http.cat/>



500

Internal Server Error



<https://httpstatusdogs.com/>



200

OK



---

## Idempotencia

Un método es 'idempotente' cuando **no importan las veces que se envíe la misma petición, ésta dará el mismo resultado.**





**Veamos algunas apis divertidas:**

<https://medium.com/@vicbergquist/18-fun-apis-for-your-next-project-8008841c7be9>



## Ejercicio:

Hacer con **Postman** un POST de alguna información en formato JSON a la api proporcionada para luego comprobar con un GET si la información se guardó correctamente.

<https://acamica-92509.firebaseio.com/>

