

# Bootcamp Java Developer

Fase 2 - Java Web Developer  
Módulo 19



# Introducción a HTTP

## ¿Qué es HTTP?

El *HyperText Transfer Protocol* es un protocolo para el intercambio de información en la red; es decir, que define las reglas de ese intercambio.

Es un protocolo para transferencias de red bajo el modelo Cliente/Servidor. El Cliente envía una petición y recibe una respuesta. El Servidor recibe una petición y envía una respuesta asociada. Por defecto, el Cliente inicia la comunicación y el Servidor responde. Esto se puede invertir gracias a los WebSockets.

Una característica importante de HTTP es que se trata de un **protocolo sin estado** ¿Qué significa esto? En lo que concierne a las reglas de este protocolo, **toda la información está contenida en el intercambio.**

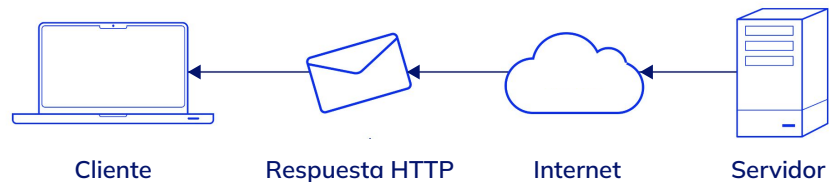
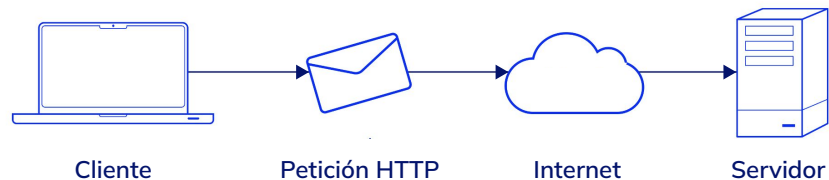
Esto quiere decir que este protocolo no regula lo que sucede dentro del servidor ni dentro del cliente (le es indiferente), sino únicamente la comunicación. En otras palabras, **regula las entradas y salidas entre los agentes, pero no el proceso interno de cada agente.**

## ¿Qué es un mensaje HTTP?

Un mensaje, desde HTTP, es la forma genérica de englobar peticiones y respuestas.

**Los mensajes (peticiones y respuestas) de HTTP son, en esencia, textos.** HTTP envía y recibe texto formateado.

- Una **Petición HTTP** es un mensaje enviado del cliente al servidor.
- Una **Respuesta HTTP** es un mensaje enviado del servidor al cliente.



## ¿Cuáles son las partes más importantes de una petición HTTP?

Veamos un ejemplo de un texto de Petición HTTP:

```
GET /productos HTTP/1.1
Content-type: application/json

{
  "buscar": "productos"
}
```

La primera línea contiene la denominada **Request-Line**.

Es, básicamente, el método y la URI. El método puede ser cualquier verbo HTTP como **GET**, **POST**, **DELETE**, y otros.

Debajo de esa línea, van las **cabeceras de la petición**. Una cabecera es un metadato. Nos permiten añadir información extra que deba tener en cuenta el servidor.

Finalmente, se escribe el **cuerpo de la petición**, que está formado por los datos que envía el cliente como información para la petición.

# ¿Cuáles son las partes más importantes de una respuesta HTTP?

## Ejemplo

```
HTTP/1.1 200 OK
Date: Sat, 20 Nov 2021 16:33:35 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1Set-Cookie:
1P_JAR=2021-11-20-16; expires=Mon, 20-Dec-2021 16:33:35
GMT; path=/; domain=.google.com;
Connection: close

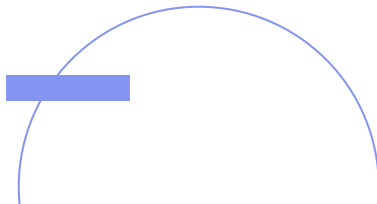
<div>Hola mundo</div>
```

## Descripción del ejemplo

En el bloque de código anterior, tenemos la línea inicial que, en el caso de la respuesta, es la **línea de estado**. El **estado HTTP** es un número que **resume toda la respuesta**. En este caso, 200 representa que el servidor pudo generar una respuesta de forma satisfactoria.

Luego, debajo, siguen una a una las **cabeceras HTTP**, con la misma lógica que la petición.

Finalmente, un **texto representando el contenido de la respuesta**. Técnicamente, el texto puede ser, justamente, cualquier texto. Por esta razón existe HTML: es un lenguaje para representar el cuerpo de respuestas HTTP.



## Links de interés

- [Especificación de HTTP.](#)
- [Métodos de petición HTTP.](#)
- [Tipos de cuerpo disponibles.](#)
- [Códigos de estado HTTP.](#)
- [Cabeceras HTTP disponibles.](#)





**¡Sigamos  
trabajando!**