

## REQUEST Y RESPONSE



HTTP REQUEST y RESPONSE son dos conceptos básicos en el desarrollo web, pero no siempre le quedan claros a los programadores. Hoy quiero definirlos, para que veamos que es mucho más sencillo de lo que pensamos:

**HTTP** significa **H**yper**T**ext **T**ransfer **P**rotocol. Esta es la forma de comunicación de datos básica en Internet. La comunicación de datos empieza con un request enviado del cliente, y termina con la respuesta del servidor web.

Por ejemplo, si fuera un ejemplo clásico con un ser humano visitando una página Web:

- Un sitio web que empieza con la URL [¡Error! Referencia de hipervínculo no válida.](#) es entrado en un navegador web de la computadora del cliente. El navegador puede ser Chrome, Firefox, o Internet explorer, no importa.
- El navegador envía un request al servidor web que está hospedado en el website.

- El servidor web regresa una respuesta como una página de HTML, o algún otro formato de documento al navegador (puede ser un mp4, mp3, pdf, doc, entre otros soportados por el navegador)

El navegador despliega el response del servidor al usuario. Por supuesto esto dependerá de los formatos que soporte el navegador

Por supuesto lo bello de esto, es que no es siempre este mismo proceso tradicional el que se ejecuta. Mucha de la comunicación vía web, ya no se hace directamente entre humano-máquina, cada vez más son aplicaciones automatizadas de ambos lados, las que se envían datos. Es decir, por ejemplo, yo tengo un programa en PHP, APEX, o algún otro lenguaje que utilizo para enviar request y recibir response, de manera que la comunicación es de máquina a máquina

Ahora bien, ¿Qué contienen el request y el response?

## HTTP Request Structure from Client

Un simple mensaje de request de un cliente tiene los siguientes componentes:

- Una línea de request para obtener el recurso, por ejemplo, un request con el método GET /content/page1.html está requiriendo el recurso llamado /content/page1.html del servidor
- Encabezados. Indican cosas como el lenguaje, codificación, tipo de datos (XML,JSON, etc). (Por ejemplo– Accept-Language: EN).
- Una línea vacía.
- Un cuerpo del mensaje que es opcional. Entre aplicaciones esta es la parte más importante. Por ejemplo, yo puedo enviar un XML o un JSON a otra máquina, y el servidor web interpretara la información que yo le mando.

Todas las líneas terminan con un retorno de carro y nueva línea. La línea vacía sólo contiene el retorno de carro y la nueva línea sin espacios.

## HTTP Response Structure from Web Server

Un simple response del servidor web contiene lo siguiente:

- HTTP Status Code (Por ejemplo, HTTP/1.1 301 Movido Permanentemente, significa que el recurso requerido fue movido permanentemente y redirigido a otro recurso).
- Encabezados. Igual que en el request, describen el contenido del response (Example – Content-Type: html)
- Una línea vacía.
- Un cuerpo de mensaje, que es opcional. Cuando trabajamos con aplicaciones, aquí puede venir la respuesta en XML u otro formato. Cuando es una página del navegador, contiene el código HTML que forma nuestra página.

Todas las líneas terminan con un retorno de carro y nueva línea. La línea vacía sólo contiene el retorno de carro y la nueva línea sin espacios.

No hay mayor ciencia en la comunicación bajo el protocolo HTTP. Pero es poderoso y flexible porque permite desde visualizar páginas web, hasta intercambiar datos entre aplicaciones.