

# Protocolo HTTP

<b>Protocolo HTTP</b>	<b>1</b>
¿Qué aprenderás?	2
Introducción	2
El protocolo HTTP	3
El modelo request/response en http	3
Mensajes http-Request	4
Mensaje http-response	7
Códigos de error http	8
La URL	9



**¡Comencemos!**

## ¿Qué aprenderás?

- Utilizar protocolos HTTP.
- Entender el concepto de arquitectura web.
- Entender request y response HTTP.

## Introducción

HTTP es un protocolo basado en un principio de cliente-servidor: Cuando un servidor web genera un request hacia un cliente, por lo general envía algún tipo de contenido para que el navegador lo interprete y lo pueda mostrar al usuario. En la mayoría de los casos, el servidor envía un set de instrucciones en formato de código HTML (Hyper Text Markup Lenguaje). El HTML le indica al cliente cómo debe visualizarse el recurso web que fue solicitado.

Entender este concepto nos permitirá entender las bases de la transmisión de información vía web y conocer los criterios de sintaxis y semántica que debemos cumplir al escribir nuestro código.

## El protocolo HTTP

Sus siglas significan *Hyper Text Transfer Protocol*, o mejor dicho en español, Protocolo de transferencia de hipertexto. Toda la interacción entre cliente y servidor es mediante este protocolo de comunicación. El cliente envía un *http-request* y el servidor responde con un *http-response*.

El http es un protocolo perteneciente a la capa de aplicación y corre sobre otro protocolo llamado TCP. Es un protocolo ampliable, lo que significa que en su evolución ha sido capaz de transmitir no solo páginas web mediante html, sino que también es capaz de transmitir imágenes, audio, video, etc.

## El modelo request/response en http

El protocolo http, y todos los protocolos que están basados en él, están contruidos bajo el mismo modelo de comunicación. Este modelo de comunicación trabaja de la siguiente manera:

Un cliente, típicamente un navegador web envía un request para que el servidor le devuelva cierto recurso. Cuando el server recibe el request, lo analiza, lo gestiona para luego enviar de vuelta un objeto response. La siguiente imagen gráfica este flujo:

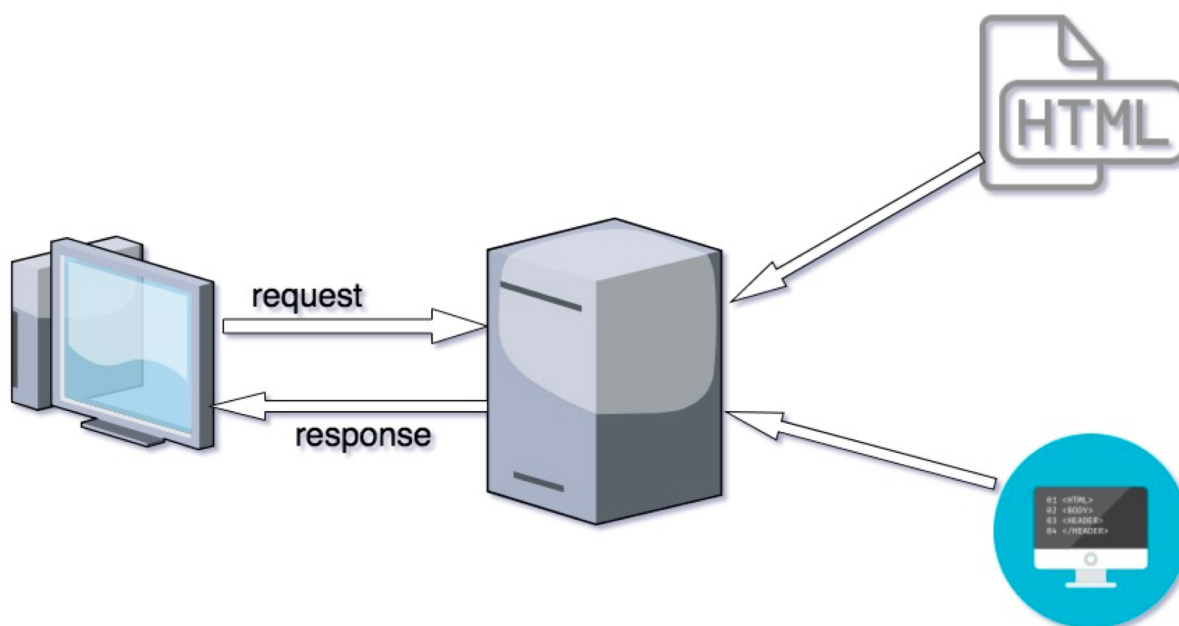


Imagen 1. Flujo request-response HTTP.  
Fuente: Desafío Latam

## Mensajes http-Request

Veamos un ejemplo de request, en este caso inspeccionamos la página de Desafío Latam, para verificar cómo se envía la petición mediante http. En este ejemplo hacemos seguimiento a un archivo .css.

Para abrir la consola del desarrollador, debes pinchar con botón derecho sobre la página web y buscar la opción Inspeccionar.

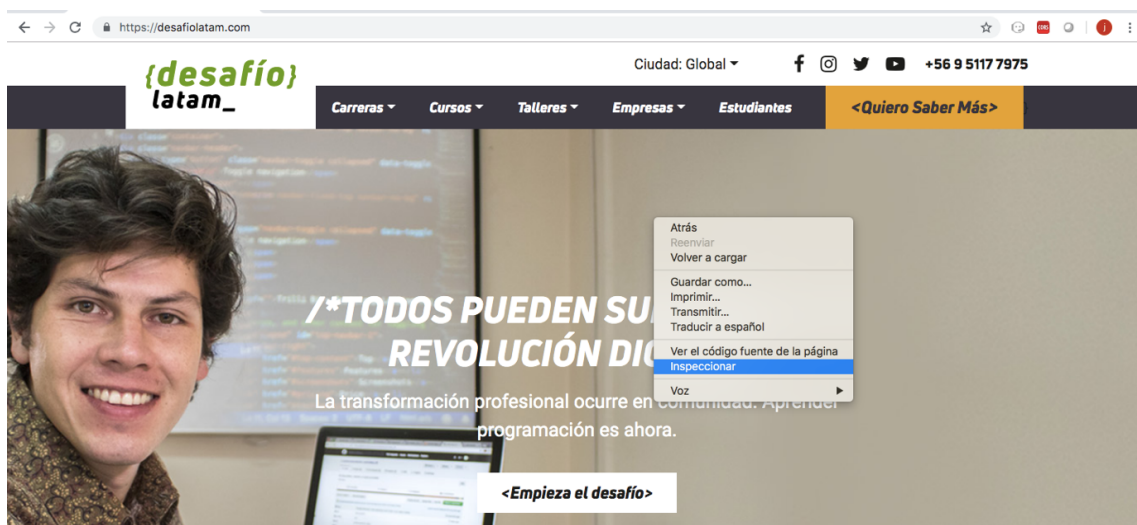


Imagen 2. Inspeccionando página de Desafío Latam.

Fuente: Desafío Latam

Al momento de presionar la opción, se abrirá un sector en la parte inferior de la pantalla, con una serie de herramientas útiles para los desarrolladores:

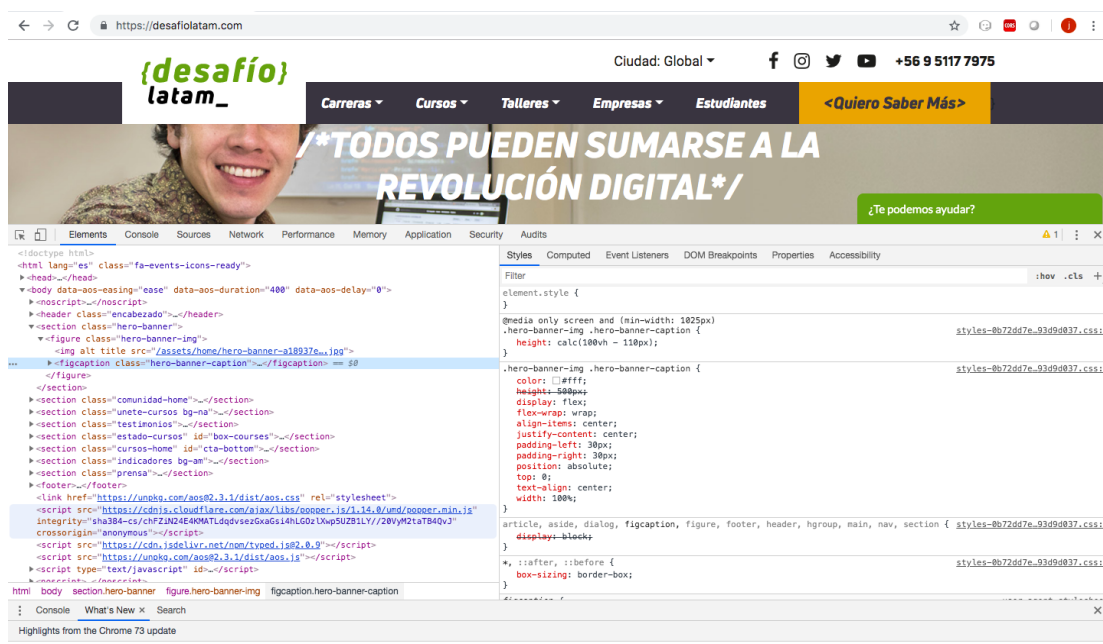


Imagen 3. Identificando la consola de desarrollador.

Fuente: Desafío Latam

En el sector izquierdo, se puede apreciar el código html que el navegador interpretó. Sirve mucho para verificar las posiciones de tus elementos e inspeccionar errores de código.

Para poder ver los contenidos de los request y response de la página web, debes pinchar la pestaña *Network* y en su pantalla principal, al lado izquierdo en la columna name, seleccionar el recurso que se desea comprobar

Al seleccionarlo, se verá toda la información del request en el lado derecho de la pantalla.

Con esos pasos es posible investigar la mayoría de elementos que hacen que la página web se despliegue.

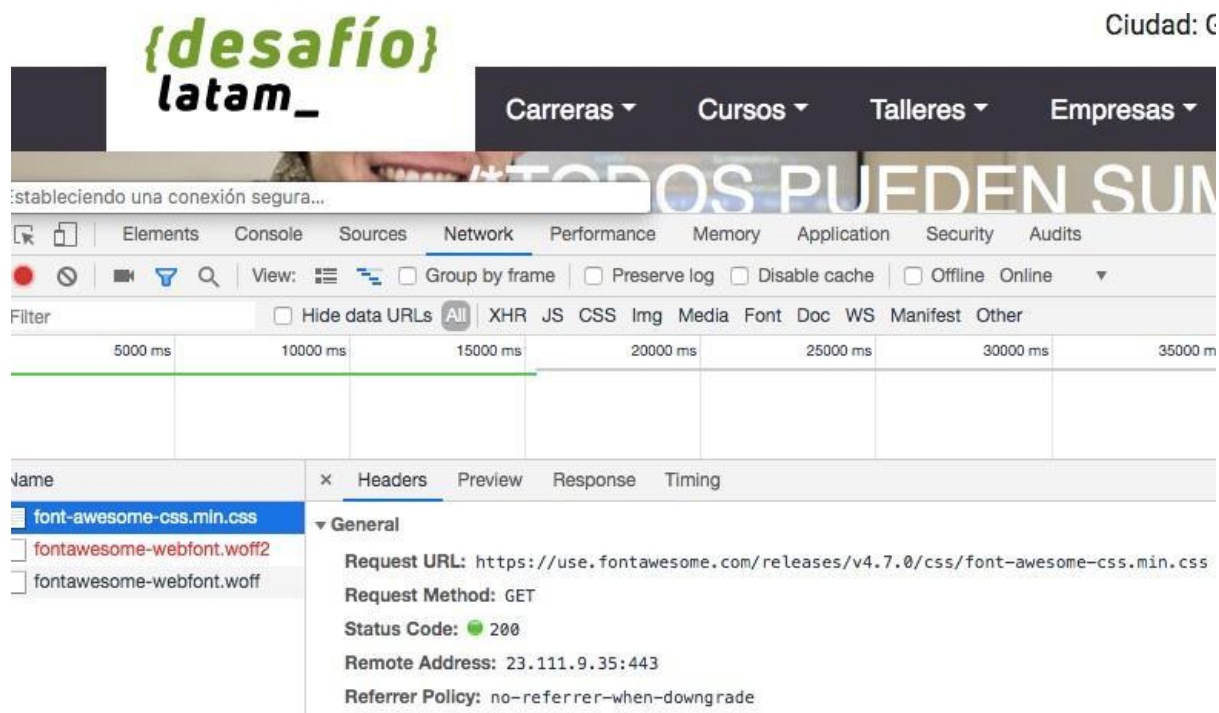


Imagen 4. Revisando la información request.  
Fuente: Desafío Latam

Vemos que la sección general, contiene una dirección URL, la cual está formada por:

- El protocolo.
- La dirección del servidor.
- La ubicación del archivo css solicitado.

Miremos la imagen 27 de nuevo, vemos un sector que tiene el título de *REQUEST METHOD* y su valor es *GET*. El valor get es un verbo *http*, y aparte del anterior existen varios más. Un verbo se utiliza para indicar de qué forma el request pide el recurso (más adelante veremos los verbos). Podemos ver también la dirección remota del servidor y la url de destino.

## Mensaje http-response

Luego de generar un request a un servidor remoto, este nos devuelve nuestro recurso. La siguiente imagen muestra como es la respuesta:

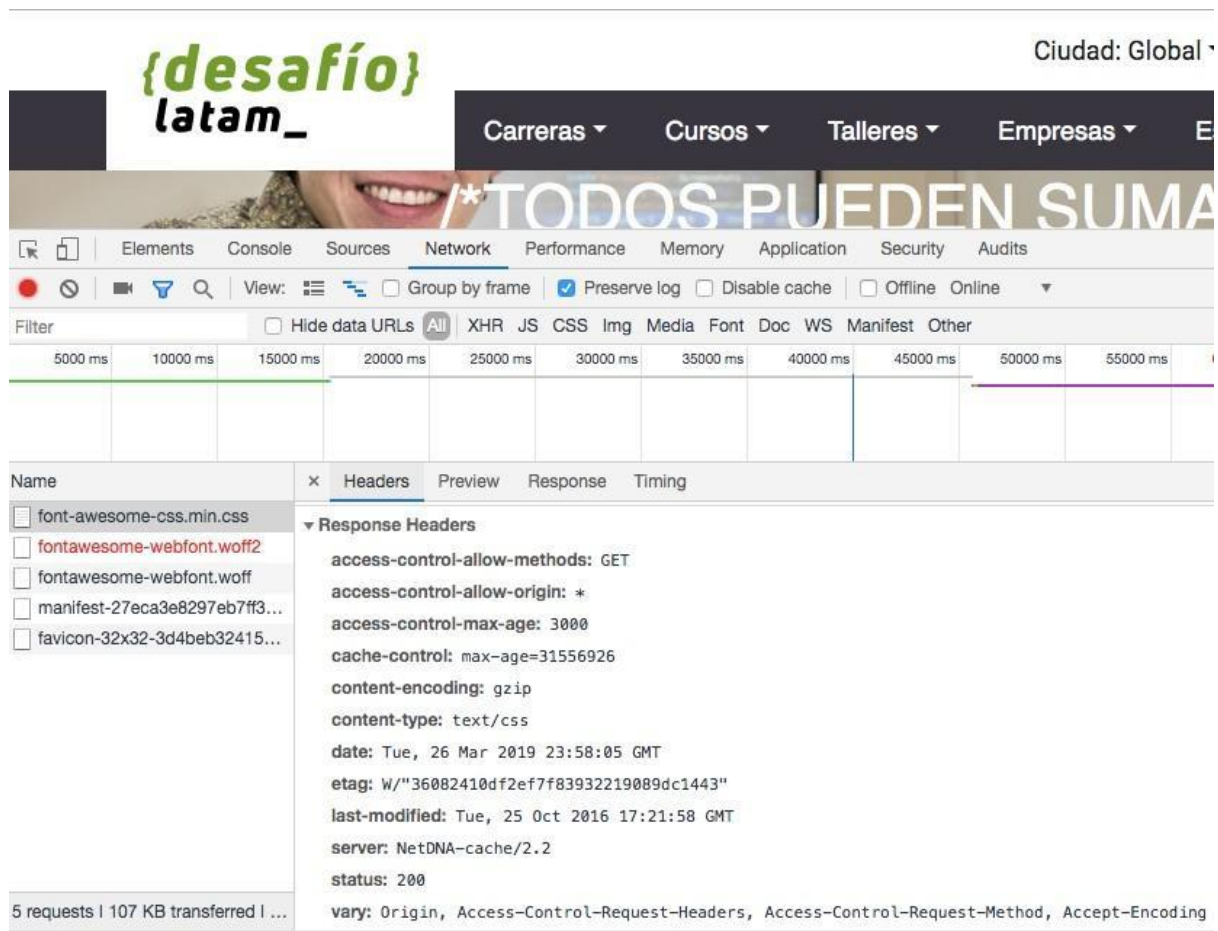


Imagen 5. Respuesta.  
Fuente: Desafío Latam

Lo importante de aquí, es que el request internamente nos entrega bastante información sobre el recurso como:

- Método de respuesta, en este caso *GET*.
- *Content Type*, en este caso un archivo *.css*.
- La fecha.

## Códigos de error http

Los request y los response vienen con un código de respuesta, el cual es generado por el servidor para indicar el resultado de los mensajes. HTTP define varios códigos de error, que podemos agrupar de la siguiente manera:

Código	Descripción	Detalle
1XX	Respuestas informáticas	Indica una respuesta provisional
2XX	Peticiones correctas	La acción solicitada por el cliente ha sido recibida, entendida, aceptada y procesada correctamente
3XX	Redirecciones	El cliente debe tomar una acción adicional para completar el request
4XX	Errores del cliente	Excepto cuando se responde a un EAD request, el servidor debe incluir una entidad que contiene una explicación del error, y su es temporal o permanente
5XX	Errores del servidor	El servidor ha fallado al completar una solicitud aparentemente válida

Tabla 1. Códigos de respuesta HTTP.

Fuente: Desafío Latam

La lista completa de los códigos podemos encontrarla en este [link](#).

Todos estos códigos de error son generados de forma automática por el servidor, y nos permite como desarrolladores saber cuál es el tipo de problema que está afectando el correcto funcionamiento de nuestros sistemas.

Por ejemplo, pensemos que nuestro sitio web tiene un banner con la imagen de nuestra empresa y necesitamos que se despliegue, pero por algún motivo ajeno a nosotros, tal imagen se borró del servidor. El servidor detectará que el recurso que estamos pidiendo no existe y automáticamente enviará un error 404. Nosotros recibimos el código de error y podemos avisar al usuario mediante algún mensaje o alguna otra acción.



## La URL

Su sigla significa *Uniform Resource Locators*, y se encarga de identificar las direcciones en donde se alojan los recursos web. Veamos su estructura:

**`https://www.desafiolatam.com:8927/java/inicio`**

- **https:** Indica al navegador el protocolo que utiliza en el requerimiento. En este ejemplo es https.
- **www.desafiolatam.com:** Es la ubicación física de la máquina en la cual está alojado el sitio.
- **8927:** Puerto de escucha del servidor.
- **/java/inicio:** (Path) Es la ubicación física de los documentos que estamos solicitando, en el servidor.