



# APLICACIONES WEB CON HTML5, CSS3 Y JAVASCRIPT

LUIS BASTO DÍAZ

# ¿Qué es un Servidor Web?

---

Para publicar nuestras páginas Web necesitamos una computadora, que cuente con conexión de red permanente.

El software que se encarga de responder a la solicitud de información de algún usuario es llamado Servidor Web.

Los usuarios de Internet, o intranets, que son quienes van a buscar la información que se publica, necesitan de un software especial llamado: "cliente Web" o "navegador Web", para poder acceder al Servidor Web.

# ¿Qué es un Servidor Web?

---

En palabras simples, un servidor web es un servidor de Internet que responde a las solicitudes HTTP para entregar contenido y servicios.

Las peticiones de información que realiza “el cliente”, se llevan a cabo a través del protocolo de comunicación http.

La palabra Servidor identifica tanto al programa como a la computadora en la que dicho programa se ejecuta.

- Servidor Software.
- Servidor Hardware.

# ¿Qué es un Servidor Web?

---

En el lado del hardware, un servidor web es una computadora que almacena el software del servidor web y los archivos de componentes de un sitio web. (por ejemplo, documentos HTML, imágenes, hojas de estilo CSS y archivos JavaScript).

Un servidor web se conecta a Internet y admite el intercambio de datos físicos con otros dispositivos conectados a la web.

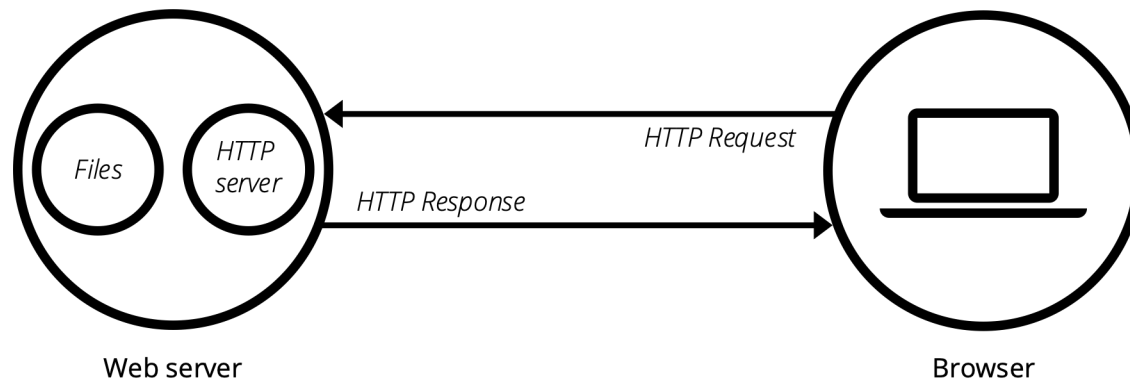
En el lado del software, un servidor web incluye varias partes que controlan cómo los usuarios web acceden a los archivos alojados.

# ¿Qué es un Servidor Web?

---

Un servidor Web se encarga de mantenerse a la espera de peticiones, llevadas a cabo por un cliente HTTP (navegador).

El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita .



[https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server)

# Protocolo HTTP

---

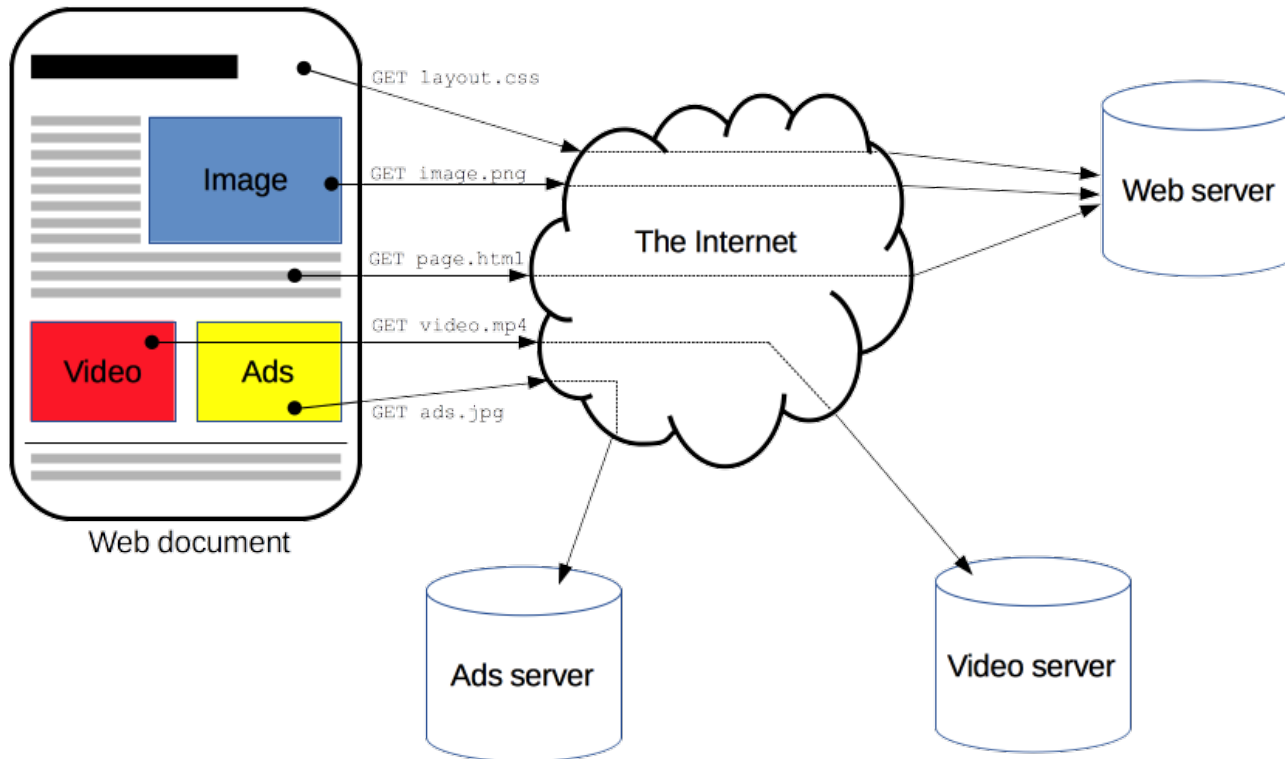
El protocolo HTTP, por sus siglas en inglés: "Hypertext Transfer Protocol", es el que nos permite realizar una petición de datos y recursos.

Una página web completa se compone de varios sub-documentos recibidos, como, por ejemplo: un documento que especifique el estilo de maquetación de la página web (CSS), el texto, las imágenes, vídeos, scripts, etc. <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>.

El navegador envía una petición de documento HTML al servidor.

Se procesa este documento, y envía más peticiones para solicitar scripts, hojas de estilo (CSS), y otros datos que necesite (vídeos, imágenes, etc.).

El navegador, une todos estos documentos y datos, y entrega la página Web completa.



# Protocolo HTTP

---

# Protocolo HTTP

---

Entre el cliente y el servidor, pueden existir distintos dispositivos que administran los mensajes HTTP.

La mayoría de estos dispositivos solamente gestionan estos mensajes en los niveles de protocolo inferiores: capa de transporte, capa de red o capa física, de manera que son transparentes para la capa de aplicación del HTTP.

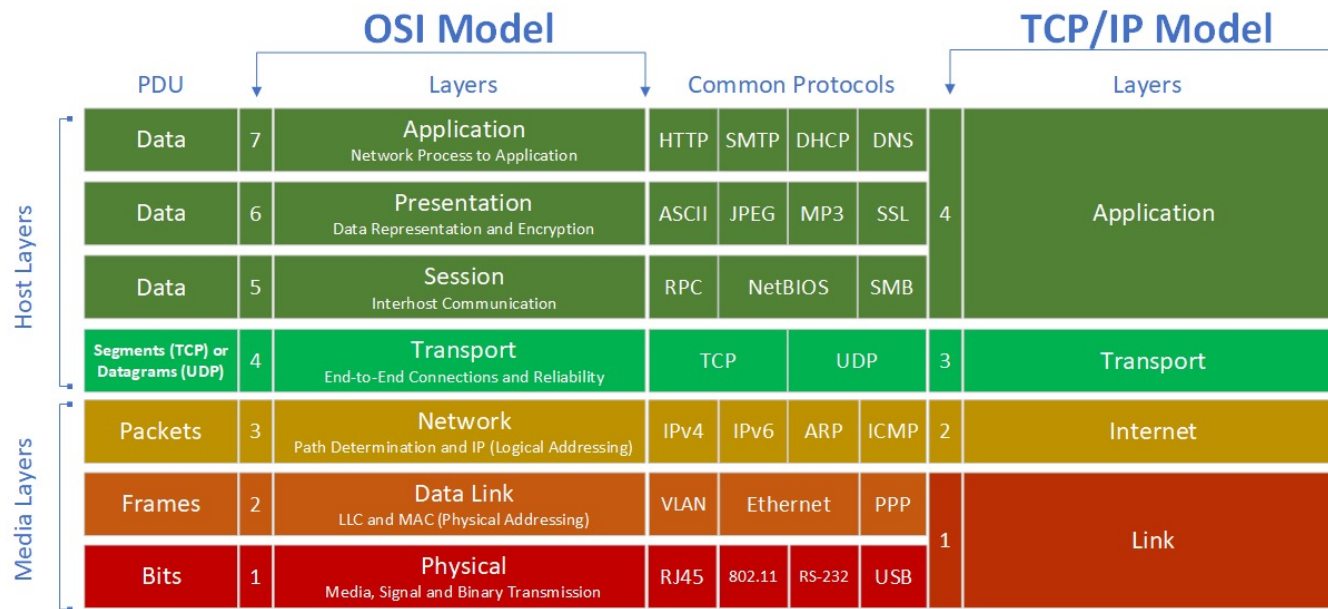
Los dispositivos que sí operan procesando la capa de aplicación se conocen como proxies, los cuales pueden ser transparentes, o no (modificando las peticiones que pasan por ellos), y realizan varias funciones:

- caching
- filtrado
- balanceo de carga de peticiones
- autenticación
- registro de eventos



# Protocolo HTTP

HTTP es un protocolo de la capa de aplicación y se transmite sobre el protocolo TCP.



# Características del Protocolo HTTP

---

Es sencillo

Es extensible

Con sesiones, pero sin estado

# Métodos HTTP

---

Cuando el cliente quiere comunicarse con el servidor, tanto si es directamente con él, o a través de un proxy intermedio, realiza los siguientes pasos:

1. Abre una conexión TCP: El cliente puede abrir una conexión nueva, reutilizar una existente, o abrir varias a la vez hacia el servidor.
2. Hace una petición HTTP
3. Leer la respuesta enviada por el servidor
4. Cierre o reutilización de la conexión para futuras peticiones.

# Práctica

---

visitar con el google Chrome [www.uady.mx](http://www.uady.mx), la herramienta de desarrollador inspeccionar.

En la opción Network, seleccionar algún recurso e identificar sus elementos.

# Peticiones HTTP

---

Una petición de HTTP está formada por los siguientes elementos:

- Un método HTTP, normalmente pueden ser un verbo, como: GET, POST o un nombre como: OPTIONS o HEAD, que defina la operación que el cliente quiera realizar.
- La URL del recurso solicitado
- La versión del protocolo HTTP.
- Cabeceras HTTP opcionales, que pueden aportar información adicional a los servidores.
- Cuerpo de mensaje, en algún método, como puede ser POST, en el cual envía la información para el servidor.

# Peticiones HTTP

---

## ▼ General

**Request URL:** `https://www.uady.mx/`

**Request Method:** GET

**Status Code:**  200 OK

**Remote Address:** `13.92.255.123:443`

**Referrer Policy:** `strict-origin-when-cross-origin`

# Respuestas HTTP

---

Las respuestas están formadas por los siguientes campos:

- La versión del protocolo HTTP que están usando.
- Un código de estado, indicando si la petición ha sido exitosa, o no, y debido a que.
- Un mensaje de estado, una breve descripción del código de estado.
- Cabeceras HTTP, como las de las peticiones.
- Opcionalmente, el recurso que se ha pedido.

# Respuestas HTTP

---

## ▼ Response Headers [View source](#)

**Connection:** Keep-Alive

**Content-Encoding:** gzip

**Content-Length:** 12964

**Content-Type:** text/html; charset=UTF-8

**Date:** Mon, 18 Oct 2021 17:46:56 GMT

**Keep-Alive:** timeout=5, max=100

**Server:** Apache

**Vary:** Accept-Encoding



# Códigos de retorno

---

1xx – Mensaje de información

2xx – Éxito

3xx - Redirección

4xx - Error de cliente

5xx - Error del servidor

# Métodos HTTP

---

Método	Descripción
OPTIONS	Una petición de información sobre las opciones de comunicación disponibles. No recupera un recurso.
GET	Petición para recuperar un recurso como un HTML o una imagen.
POST	Petición para que el servidor acepte los datos enviados desde el cliente para modificar los datos existentes en el servidor
PUT	Petición para que el servidor acepte los datos enviados desde el cliente para insertar nuevos datos.
DELETE	Petición para que el servidor elimine un recurso específico

# Sitios y Aplicaciones Web

---

## Sitio Web

- Un sitio Web Contiene información posiblemente estática.
- Es actualizado por diseñadores o programadores.
- Posiblemente necesite solamente tecnología HTML para desplegar páginas Web.

# Sitios y Aplicaciones Web

---

## Aplicación Web

- Contiene información dinámica.
- Es actualizado por los propios usuarios (u otros programas) a través de formularios.
- Utiliza un lenguaje de programación para su desarrollo.
- Necesita de una combinación de diversas tecnologías para desplegar las páginas Web al navegador.

# Aplicaciones del lado del Cliente

---

El cliente Web (navegador) es el encargado de ejecutarlas en la máquina del usuario.

Son las aplicaciones tipo Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta.

Es necesario, que el cliente cuente con capacidad para ejecutar estas aplicaciones (también llamadas scripts).

# Aplicaciones del lado del Servidor

---

Suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones Web.

El código de las aplicaciones Web, se ejecutan en el servidor y no en la máquina del cliente.

El cliente no necesita ninguna capacidad adicional, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java.

Cualquier cliente que cuente con un navegador web básico puede utilizar este tipo de aplicaciones.

# Aplicaciones del lado del Servidor

---

Entre las Tecnologías para programar aplicaciones por el lado del servidor se encuentran:

PHP (Pre Hypertext - Processor )

JSP (Java Server Pages)

ASP.net (active Server Pages)

Python

NodeJS

C#

# Servidores Web

---

Hay varios servidores web disponibles en el mercado, algunos son:

## Servidor HTTP Apache.

- Desarrollado por Apache Software Foundation, es un servidor web gratuito y de código abierto para Windows, Mac OS X, Unix, Linux, Solaris y otros sistemas operativos;
- necesita la licencia de Apache (<https://httpd.apache.org/>)

## Servicios de información de Internet de Microsoft (IIS).

- Desarrollado por Microsoft para plataformas Microsoft;
- no es de código abierto, pero se utiliza ampliamente (<https://www.iis.net/>).



# Servidores Web

---

Hay varios servidores web disponibles en el mercado, algunos son:

Nginx.

- Un popular servidor web de código abierto para administradores debido a su escasa escalabilidad y utilización de recursos.
- Puede manejar muchas sesiones concurrentes debido a su arquitectura impulsada por eventos.
- También se puede utilizar como servidor proxy y equilibrador de carga (<https://www.nginx.com/>).

Lighttpd.

- Un servidor web gratuito que viene con el sistema operativo FreeBSD.
- Se considera rápido y seguro, y consume menos energía de la CPU (<https://www.lighttpd.net/>)

# Instalación de Herramientas

---

Visual studio code (<https://code.visualstudio.com/>)

Extensiones de visual Studio Code

- Lenguaje español
- Auto close tag
- Auto rename tag
- Better comment
- HTML5 snippets
- HTML CSS support
- IntelliSense for CSS
- HTMLHint
- Live Server
- Live preview
- Image preview