Portafolio

Arquitecturas Web

Colegio Montessori 2 DAW

Ricardo Sorin Almajan

# Índice

[Índice 1](#_Toc178616170)

[Definición Arquitecturas Web: 2](#_Toc178616171)

[Tecnologías de desarrollo web 2](#_Toc178616172)

[ ASP: 2](#_Toc178616173)

[ JSP: 2](#_Toc178616174)

[ PHP: 2](#_Toc178616175)

[Plataformas web libres y propietarias. 3](#_Toc178616176)

[LAMP 3](#_Toc178616177)

[Servidor WEB 3](#_Toc178616178)

[Protocolos de comunicación. 3](#_Toc178616179)

[OSI 3](#_Toc178616180)

[TCP/IP 5](#_Toc178616181)

[Servidores 5](#_Toc178616182)

[Tipos de servidores 6](#_Toc178616183)

[Tipos de servidores web 6](#_Toc178616184)

[ Apache 7](#_Toc178616185)

[ NginX 7](#_Toc178616186)

[ GWS 7](#_Toc178616187)

[ Microsoft IIS 7](#_Toc178616188)

[ Lite Speed 7](#_Toc178616189)

[ Tomcat 8](#_Toc178616190)

[Mapa Conceptual: 8](#_Toc178616191)

[Fundamentación teórica 9](#_Toc178616192)

[Reflexión personal 11](#_Toc178616193)

# 

# Definición Arquitecturas Web:

La arquitectura de la World Wide Web es un modelo flexible que utiliza estándares para la presentación y localización de aplicaciones y contenidos. Los navegadores web envían peticiones a servidores, que responden con datos codificados en formatos estándar como HTML, ECMA y JavaScript, usando protocolos como HTTP sobre TCP/IP.

La arquitectura web es un modelo compuesto de tres capas:  
Base de datos, Servidores de aplicaciones, Clientes.  
En sus inicios, la web era estática y poco interactiva. Con la llegada de la Web 2.0, se ha dado paso a aplicaciones más dinámicas y centradas en el usuario, permitiendo colaboración, redes sociales, control de la información y la transición de software de escritorio a la web.

# Tecnologías de desarrollo web

Las aplicaciones dinámicas son las que emplean webs dinámicas, estas para mostrase en el equipo del cliente que las solicita se ejecutan en un servidor web, las tecnologías mas comunes para esto son:

## ASP:

Los clientes que realizan las solicitudes recibirán una pagina que bravíamente se ha ejecutado en el lado del servidor.

## JSP:

Con JSP podemos crear aplicaciones web que se ejecuten en distintos servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. JSP aporta la ventaja de poder utilizar las clases Java, así como sus herramientas y acceso a BBDD.

## PHP:

Este lenguaje es muy similar al ASP, pero con la diferencia de que nos permite hacer páginas web dinámicas más complejas por la implantación de las bases de datos.

# Plataformas web libres y propietarias.

Una plataforma web es el entorno de desarrollo de software empleado para diseñar y ejecutar un sitio web.

* El sistema operativo, bajo el cual opera el equipo donde se hospedan las páginas web y que representa la base misma del funcionamiento del computador. En ocasiones limita la elección de otros componentes.
* El servidor web es el software que maneja las peticiones desde equipos remotos a través de la Internet.
  + En el caso de páginas estáticas, el servidor web simplemente provee el archivo solicitado, el cual se muestra en el navegador.

En el caso de sitios dinámicos, el servidor web se encarga de pasar las solicitudes a otros programas que puedan gestionarlas adecuadamente.

* El gestor de bases de datos se encarga de almacenar sistemáticamente un conjunto de registros de datos relacionados para ser usados posteriormente.
* Un lenguaje de programación interpretado que controla las aplicaciones de software que corren en el sitio web.

## LAMP

Plataforma que trabaja con software libre para no tener restricciones/limitaciones, sus siglas proviene de:

* Linux: Sistema operativo.
* Apache: Servidor web.
* MySQL: Gestor de bases de datos.
* PHP: Lenguaje interpretado PHP, aunque a veces se sustituye por Perl o Python.

Existe varias aplicaciones más, entre ellas la más conocida WAMP (Windows-Apache-MySQL-PHP).

## Servidor WEB

Programa que se ejecuta de forma continua en un ordenador, se mantiene a la espera de peticiones por parte de un cliente y contesta a estas peticiones de forma adecuada, sirviendo una página web que será mostrada en el navegador o mostrando el mensaje correspondiente si se detectó algún error.

Normalmente a la solicitud que hacen los clientes al servidor se le llama petición y a lo que el servidor devuelve a dicho cliente le llamamos respuesta. También hay que tener en cuenta que esta arquitectura cliente-servidor plantea la posibilidad de numerosos clientes atendidos por un mismo servidor.

## Protocolos de comunicación.

### OSI

El modelo OSI lo desarrolló allá por 1984 la organización ISO. Este estándar perseguía el ambicioso objetivo de conseguir interconectar sistemas de procedencia distinta para que estos pudieran intercambiar información sin ningún tipo de impedimentos debido a los protocolos con los que estos operaban de forma propia según su fabricante. Precisamente esta separación en niveles hace posible la intercomunicación de protocolos distintos al concentrar funciones específicas en cada nivel de operación.

Los protocolos de cada nivel se comunican con sus homólogos o peer, es decir su mismo protocolo situado en el otro extremo de la comunicación. Luego en la máquina de destino el flujo llegará a esta capa física y subirá hasta la capa más superficial que exista. Además, cada nivel trabaja de forma independiente de las demás, si necesidad se saber el funcionamiento del resto de niveles. El modelo OSI está conformado por 7 capas o niveles de abstracción.

* Capa 1: Física

Esta capa se encarga de los aspectos físicos de la conexión en una red, asegurando la transmisión de datos de manera fiable entre el transmisor y el receptor.

* Capa 2: Enlace de datos

Esta capa proporciona los medios para establecer la comunicación entre los dispositivos conectados físicamente. Se encarga del direccionamiento físico de los datos, la gestión del acceso al medio y, principalmente, de la detección de errores durante la transmisión.

* Capa 3: Red

Esta capa gestiona el enrutamiento de datos entre dos o más redes conectadas. Su función principal es garantizar que los datos viajen desde el transmisor hasta el receptor, realizando las conmutaciones y encaminamientos necesarios.

* Capa 4: Transporte

Esta capa gestiona el transporte fiable de los datos desde el origen hasta el destino, sin importar el tipo de red utilizada por las capas inferiores.

* Capa 5: Sesión

Esta capa gestiona el control y mantenimiento del enlace entre las máquinas que están intercambiando información, asegurando que la conexión permanezca activa durante toda la transmisión.

* Capa 6: Presentación

Esta capa se ocupa de la representación y formato de la información transmitida, garantizando que los datos sean comprensibles para los usuarios, independientemente de los diferentes protocolos utilizados por el transmisor y el receptor.

* Capa 7: Aplicación

Esta es la capa más alta del modelo y se encarga de permitir a los usuarios realizar acciones y comandos a través de sus aplicaciones, como enviar correos electrónicos o transferir archivos mediante FTP.

# TCP/IP

Es un conjunto de protocolos, toma su nombre de dos de sus protocolos más importantes, el protocolo TCP (Transmission Control Protocol) y el protocolo IP (Internet Protocol). Ç



# Servidores

Un servidor es un ordenador u otro tipo de equipo informático encargado de suministrar información a una serie de clientes, que pueden ser tanto personas como otros dispositivos conectados a él. Dicho de otra manera, un servidor es un equipo informático que forma parte de una red y provee servicios a otros equipos cliente.

La puesta a disposición de los servicios del servidor a través de una red informática se basa en el modelo cliente-servidor, concepto que hace posible distribuir las tareas entre los diferentes ordenadores y hacerlas accesibles para más de un usuario final de manera independiente.

## Tipos de servidores

* Servidor dedicado: atiende solicitudes de cientos.
* Servidor de Correo: Almacena las operaciones relacionadas con los emails de los clientes.
* Servidor Proxy: El que se encarga de “esconder” a los clientes de cara al servidor para que no se sepa que realmente son.
* Servidor de Impresión: se encarga de servir impresoras a los equipos cliente.
* Servidor FTP: su función es permitir el intercambio de ficheros entre equipos.
* Servidor de Base de Datos: Da servicios de almacenamiento y gestión de bases de datos a sus clientes.
* Servidores Clúster: Almacena grandes cantidades de información y evita la perdida de esta.
* Servidores de imágenes: permite alojar gran cantidad de imágenes sin consumir recursos.
* Servidor de Audio/Video: Permite la incorporación de servicios de streaming a los servidores.
* Servidor Web: Almacena principalmente documentos HTML, imágenes, videos, texto, presentaciones, y en general todo tipo de información.

# Tipos de servidores web

Un servidor HTTP, conocido como servidor web, es un software que se instala, corre y procesa información desde el lado del servidor. El servidor web es el encargado de despachar archivos por los protocolos HTTP y HTTPS hacia un cliente que solicita dicha información.

Los servidores web son, de manera muy simplificada, programas que se instalan en un ordenador para hacer que esté permanentemente conectado a Internet y pueda responder a las peticiones de los usuarios o de otras máquinas.



## Apache

El servidor HTTP Apache es el más antiguo y utilizado hasta la actualidad. La Apache Foundation lo desarrolló a mediados de los años 90 y se puede instalar en plataformas Linux, Windows y Mac OSX, entre otras.

Además, es uno de los más estables, lo que lo hace una opción muy rentable, y la mayor parte de los fallos de seguridad que se han descubierto en el pasado solo podían ser aprovechados por usuarios locales.

## NginX

La característica más destacada de nginx es que se basa en eventos, a diferencia de Apache —que usa un sistema de hilos— y puede gestionar miles de conexiones simultáneas —algo ideal para los servidores proxy durante situaciones de emergencia— empleando muchos menos recursos que otros modelos. Entre sus beneficios encontramos una configuración simple, pero al mismo tiempo poderosa, permitiendo configurarlo para integrarse nativamente con casi cualquier tecnología y lenguaje de programación moderno. Dropbox, Netflix y WordPress, entre otros muchos, hacen uso de servidores nginx.

## GWS

Es un servidor web personalizado utilizado por Google en su infraestructura. Escrito en C++, GWS gestiona aproximadamente el 2% de todas las páginas web activas en el mundo. Al analizar los encabezados de una página web de Google, se puede observar la indicación «GWS», lo que confirma que está en funcionamiento el servidor de Google.

## Microsoft IIS

Es un servidor web desarrollado por Microsoft para su sistema operativo Windows. Originado en el antiguo "Option Pack" para Windows NT, se integró en versiones posteriores como Windows Server 2003 y 2008 debido a su popularidad.

IIS permite el procesamiento y despacho de páginas desarrolladas en tecnologías como ASP y ASP.NET, y también puede interpretar páginas escritas en Perl o PHP. Sin embargo, su principal desventaja es que es un servidor propietario exclusivo de Windows, lo que limita su integración con diversas tecnologías y lenguajes en comparación con otros servidores web.

## Lite Speed

LiteSpeed es un servidor HTTP desarrollado por LiteSpeedTech, considerado una de las mejores alternativas a Apache para entornos Linux. Ofrece una versión de código abierto y una versión comercial, que requiere pago.

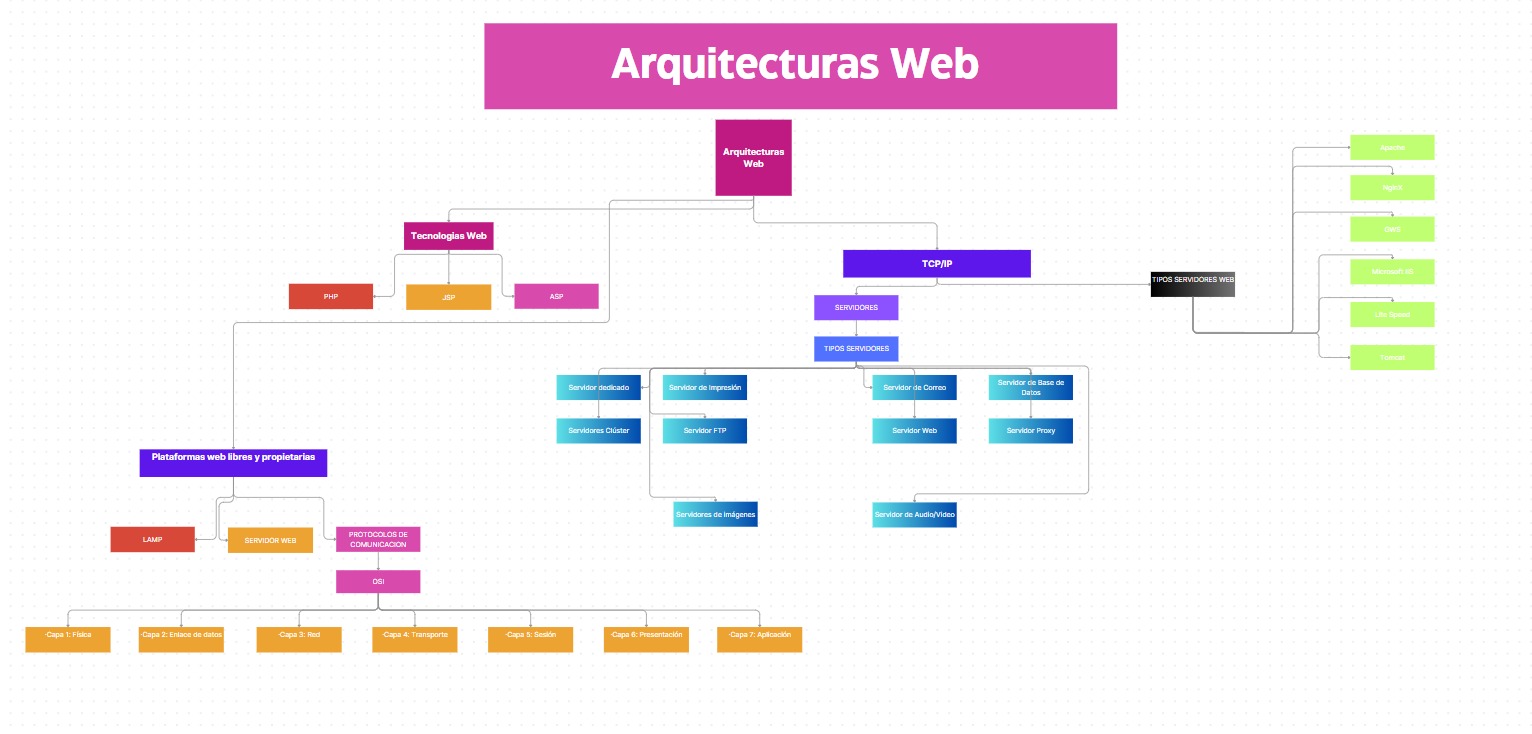
LiteSpeed es capaz de manejar una gran cantidad de conexiones simultáneas con un bajo consumo de recursos, incluso para aplicaciones exigentes que utilizan PHP. En cuanto al manejo de archivos estáticos, es comparable a Nginx. Además, se integra de manera nativa con paneles de control de hosting como Plesk y cPanel, lo que ha facilitado su adopción por proveedores de alojamiento web.

## Tomcat

Tomcat (también conocido como Jakarta Tomcat o Apache Tomcat) es un servidor de contenedores web desarrollado por la Apache Software Foundation y publicado bajo la licencia Apache 2.0. Se utiliza principalmente para el desarrollo de Java Servlets y JavaServer Pages (JSP).

A diferencia de un servidor HTTP/HTTPS convencional, Tomcat está diseñado específicamente para soportar aplicaciones basadas en Java y no debe confundirse con un servidor de aplicaciones Java como JBoss. Fue creado para cumplir con las especificaciones de Sun Microsystems y es ampliamente utilizado en entornos empresariales.

# Mapa Conceptual:



# Fundamentación teórica

* Investiga y resume las diferencias entre web 1.0, web 2.0 y web 3.0

Web 1.0

Características: Conocida como la «web estática», era principalmente unidireccional, donde la mayoría de los usuarios eran consumidores de contenido y había pocos creadores. Enfoque: Se centraba en el contenido controlado por empresas, donde la mayoría de la información era presentada en formato HTML.

Web 2.0

Características: Conocida como la "web social", permitió la creación de contenido generado por los usuarios, facilitando la interacción y la colaboración. Las aplicaciones web se volvieron más dinámicas y usables. Enfoque: Se orienta hacia el contenido comunitario, donde los usuarios pueden interactuar, compartir y colaborar.

Web 3.0

Características: También conocida como la "web semántica", busca transformar la web en una base de datos interconectada y comprensible para máquinas, utilizando tecnologías avanzadas como inteligencia artificial y blockchain. Enfoque: Se centra en el usuario individual y en la personalización de la experiencia online, facilitando la interacción inteligente y la agregación de contenido relevante.

* El protocolo TCP/IP se basa en una amplia pila de protocolos. Investiga al menos 8 y explica a grandes rasgos cómo funcionan, para qué sirven y en qué capa actúan. (Entre ellos tienen que estar los siguientes: HTTP, UDP, TCP e IP)

**HTTP** : Fundamental para la transferencia de páginas web a través de Internet, opera en la capa de aplicación. Permite a los clientes solicitar recursos a los servidores web. Al ingresar una URL, HTTP facilita la solicitud y entrega de la página correspondiente.  
**HTTPS** : Es una extensión de HTTP que incorpora medidas de seguridad, generalmente utilizando SSL/TLS para cifrar los datos transferidos entre el cliente y el servidor, asegurando la confidencialidad e integridad de la comunicación.  
**TCP** : Funciona en la capa de transporte y garantiza la entrega confiable, ordenada y verificada de datos entre aplicaciones. Segmenta los datos en paquetes, proporciona confirmaciones de recepción y retransmite paquetes perdidos, lo que lo hace adecuado para aplicaciones donde la precisión es crítica, como las transferencias de archivos y la navegación web.  
**UDP** : También en la capa de transporte, UDP ofrece un modelo de comunicación sin conexión, permitiendo que los datos se envíen sin necesidad de establecer primero una conexión. No garantiza la entrega ni el orden de los paquetes, lo que lo hace más rápido pero menos confiable que TCP. Se usa a menudo en aplicaciones en tiempo real como la transmisión de video y los juegos en línea.  
**IP** : Este protocolo opera en la capa de red y es responsable de la dirección y el enrutamiento de los paquetes de datos a través de las redes. Cada dispositivo conectado a una red recibe una dirección IP única, que se utiliza para identificarlo y dirigir el tráfico de manera adecuada.  
**FTP** : En la capa de aplicación, FTP se utiliza para transferir archivos entre computadoras en una red. Permite a los usuarios cargar, descargar y gestionar archivos en un servidor remoto, admitiendo tanto el acceso autenticado como el anónimo.  
**SMTP** : Este protocolo de la capa de aplicación se utiliza para enviar correos electrónicos. Facilita la transferencia de mensajes de correo electrónico desde un cliente a un servidor o entre servidores, asegurando el enrutamiento y la entrega adecuada de los mensajes.  
**DNS** : Operando principalmente en la capa de aplicación, DNS traduce nombres de dominio legibles por humanos en direcciones IP, lo que permite a los usuarios acceder a sitios web sin necesidad de recordar direcciones numéricas.

* Investiga y explica con tus propias palabras la diferencia entre un servidor basado en hilos como Apache y un servidor basado en eventos como NginX.

Los servidores web, como **Apache y NginX**, usan diferentes enfoques para gestionar las conexiones de los clientes. Apache utiliza un modelo basado en procesos, donde se crea un nuevo proceso para cada solicitud. Es sencillo de usar y permite una buena compatibilidad con varias aplicaciones, pero consume muchos recursos. A medida que el número de conexiones aumenta, Apache puede enfrentar limitaciones de rendimiento.

Por otro lado, NginX opera con un modelo basado en eventos, lo que le permite manejar múltiples conexiones en un solo hilo mediante un bucle de eventos. Este es más eficiente en el uso de recursos y puede gestionar miles de conexiones simultáneamente sin la sobrecarga asociada con la creación de múltiples hilos. Esto lo convierte en una opción para sitios web de alto tráfico que requieren servir contenido estático de manera rápida y efectiva.

# Reflexión personal

* ¿Alguna vez has utilizado o configurado alguna máquina para que ejerciese de servidor? Cuenta tu experiencia.

Hemos utilizado este año **Docker** y el año pasado XAMP, la verdad que no he entendido mucho del proceso, pero lo que he podio entender es, **XAMPP** es ideal para un entorno de desarrollo local, mientras que **Docker** ofrece una solución más avanzada y escalable para la creación y despliegue de aplicaciones en contenedores.

* En un día normal, ¿con cuántos servidores crees que entablas conexión? ¿Qué tipo de servidor son, hay alguno no web?

Podara llegar a usar entre 10-15 servidores dependido los contenidos que consulte, personalmente creo que son mas ya que en mi trabajo y en clase me veo obligado a consultar bastantes sitos para obtener información sobre el código que desarrollo, entre las páginas que consulto también entra las redes sociales a las cuales su servidor no es web.

* ¿Cómo de importante consideras tener conocimientos básicos sobre nomenclatura y funcionamiento de arquitecturas web para tu futura vida profesional? ¿Por qué?

Para mi punto de vista lo veo como algo fundamental tener nociones básicas sobre este tema, no solo por la categoría de mi trabajado sino para poder solventar cualquier problema relacionado, te ayuda en gran medida tener conocimientos de lo que esta sucediendo.