

# DAW2-DWES



## TEMA 1

### DESARROLLO WEB EN ENTORNO SERVIDOR

#### Alumnos:

Alberto Fernández 6, 7, 19 y 20

Aroa Granero 1,3,4,19 y 20

David del Prado 13, 14

Johanna Herrero 2,5,21

María Isabel Martínez 10,11,12

Oscar Llamas 15,16

Outmane Bouhou 17,18

Rodrigo Geras 8,9(A: 5,14)

2021-2022

## ÍNDICE

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.....	4
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.....	6
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.....	8
4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme) /URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.....	10
5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.....	11
6. Modelo de división funcional front-end / back-end para aplicaciones web.....	12
7. Página web estática – página web dinámica – aplicación web – mashup .....	14
8. Componentes de una aplicación web.....	16
9. Programas ejecutados en el lado del cliente y del servidor. Lenguajes de programación utilizados en cada caso.....	19
10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).....	21
11. Características y posibilidades de desarrollo de una plataforma XAMPP.....	24
12. En qué casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación.....	25
13. IDE más utilizados (características y grado de implantación actual).....	26
14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).....	27
15. Apache HTTP vs Apache Tomcat.....	29
16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).....	31
17. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, .....	33
18. Repositorios de software – sistemas de control de versiones.....	35
19. Propuesta de configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USED y xxx-WXED.....	37
20. Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USEE.....	39
21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web:.....	40

22. Elegir y realizar un estudio y una presentación para la exposición del trabajo sobre una de las siguientes arquitecturas de desarrollo de Aplicaciones Web:.....41

Realiza un estudio sobre los siguientes conceptos:

## 1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.

Los **protocolos de comunicaciones** son los encargados de determinar cómo deben circular los mensajes dentro de una red. Son conjuntos de normas que permiten la comunicación entre ordenadores, estableciendo la forma de identificación de estos en la red, la forma de transmisión de los datos y la forma en que la información debe procesarse.

La **dirección IP** (Internet Protocol) es un conjunto de números que identifica, de manera lógica y jerárquica, a una interfaz en la red (elemento de comunicación/conexión) de un dispositivo que utilice el protocolo o que corresponde al nivel de red del modelo TCP/IP.



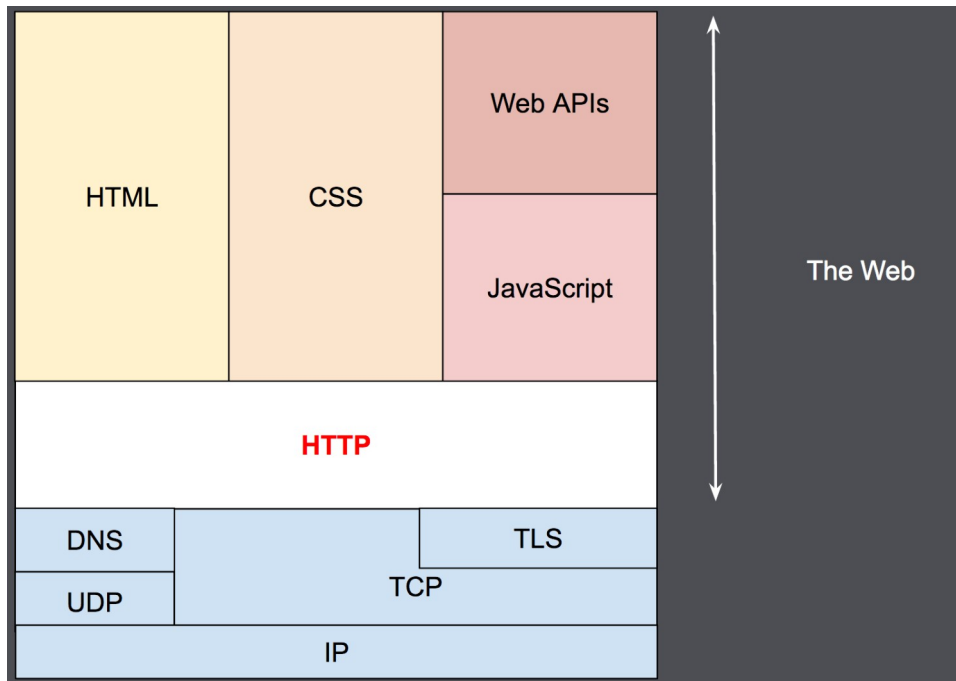
*Ilustración 1. Ejemplo de dirección IP.*

El **TCP** (Protocolo de Control de Transmisión, por sus siglas en inglés Transmission Control Protocol) es un protocolo de red importante que permite que dos anfitriones (hosts) se conecten e intercambien flujos de datos. El rol de TCP es garantizar que los paquetes se entreguen de forma confiable y sin errores, con un sistema de acuse de recibo que permite al cliente y servidor garantizar la recepción mutua de datos.

El **HTTP** (de sus siglas en inglés: "Hypertext Transfer Protocol") es un protocolo que permite realizar una petición de datos y recursos, como pueden ser documentos HTML.

Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web. Cada petición se envía a un servidor el cual gestiona y responde, página web (puerto 80).

Así, una página web completa resulta de la unión de distintos 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...



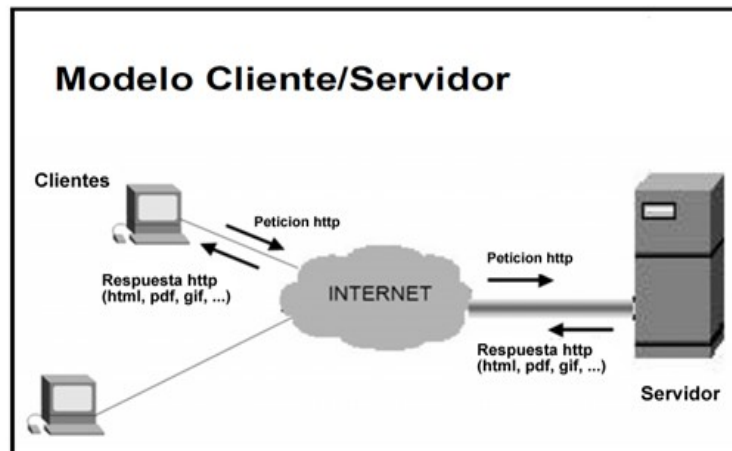
*Ilustración 2. Estructura de una página web y los protocolos que permiten acceder a ella.*

**HTTPS** es lo mismo que HTTP, pero esta se transmite de forma segura. Esto se consigue mediante un cifrado basado en SSL/TLS para crear un canal cifrado entre cliente y servidor (puerto 443).



*Ilustración 3. Diferencia entre HTTP y HTTPS.*

## 2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.



La arquitectura **cliente-servidor** es un modelo de diseño de software distribuido en dos partes, el cliente y el servidor, que se reparten las tareas. El cliente realiza peticiones al servidor y este le da las respuestas.

Se utiliza el protocolo HTTP/HTTPS tanto en la petición como en la respuesta.

### Tipos de arquitecturas cliente servidor:

Existen tres tipos dentro de la arquitectura cliente servidor:

- Arquitectura de dos capas

se utiliza para describir los sistemas **cliente servidor** en donde el cliente solicita recursos y el servidor responde directamente a la solicitud con sus propios recursos.

- Arquitectura de tres capas

Aquí existe un nivel intermediario en la arquitectura generalmente está compartida por un cliente que como hablamos más arriba es el que solicita los recursos equipados con una interfaz de usuario o mediante un navegador web.

La capa del medio es denominada software intermedio cuya tarea es proporcionar los recursos solicitados pero que requiere de **otro servidor** para hacerlo. La última capa es el servidor de datos que proporciona al servidor de aplicaciones los datos necesarios para poder procesar y generar el servicio que solicitó el cliente en un principio.

- Arquitectura N capas

La arquitectura anterior, en tres niveles, es potencialmente una arquitectura en N capas ya que puede estar compuesto por N servidores donde cada uno de ellos brindan su servicio específico

### Algunos ejemplos de la arquitectura cliente servidor pueden ser:

- **Navegar una web:** funciona basándonos en un cliente web (navegador) y un servidor web como Apache, Nginx o LiteSpeed.

- **Protocolo FTP:** funciona de idéntica forma, se utiliza un cliente de FTP (como FileZilla) para conectar a un servidor FTP (como Pure-FTPd, Proftpd, etc)
- **SSH:** es idéntico también, se utiliza un cliente SSH para conectar al servidor SSH que corre en una red remota.
- **Servidor de Correo:** donde clientes de correo consultan el correo al servidor de correo remoto, tanto desde móvil o una computadora de escritorio o laptop.

### ¿Cuál es su relación con las aplicaciones web?

Cualquier aplicación web que queramos desarrollar va a necesitar utilizar un servidor que haga uso del protocolo http o https, y un cliente que realice la petición, es decir, el modelo cliente-servidor.

El cliente (Navegador) hace una petición de una página al servidor web. Este servidor web tiene código programado en un lenguaje de programación (PHP). El servidor procesa este documento y el resultado es la devolución en formato HTML al cliente.

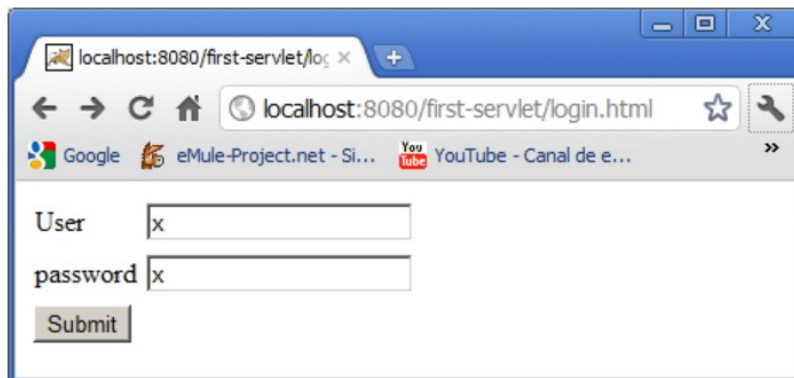
Enlace : <https://blog.infranetworking.com/modelo-cliente-servidor/>

### 3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.

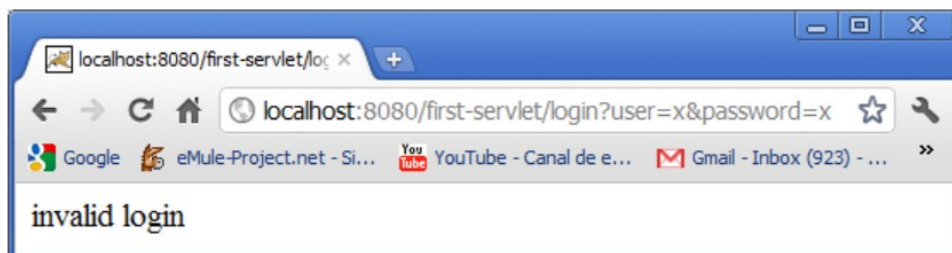
**GET y POST** son las técnicas más eficientes y utilizadas para enviar y obtener datos del servidor.

**GET:** los parámetros (campos) del formulario son enviados de manera visible a través de la URL de nuestro navegador. Está limitado a 2048 caracteres y estos deben ser ASCII.

Un ejemplo de un formulario enviado por GET:

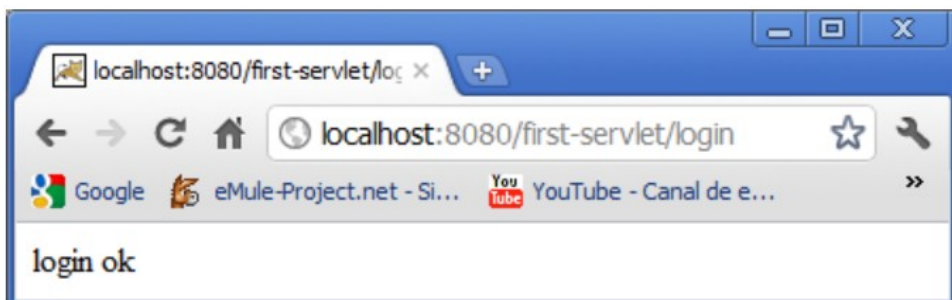


Como respuesta recibimos



El método GET se utiliza para recuperar los datos. Por el contrario, el método POST se utiliza para almacenar o actualizar los datos.

**POST:** los parámetros son invisibles para el usuario. Los datos de formularios no quedan guardados en el historial o caché como sí lo harán si utilizamos GET. Además, POST nos permite enviar al servidor más tipos de datos como textos, fotos, vídeos...





**Más Información:**

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/get-vs-post/>

[https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=527:get-y-post-html-method-formas-de-envio-de-datos-en-formulario-diferencias-y-ventajas-ejemplos-cu00721b&catid=69&Itemid=192](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=527:get-y-post-html-method-formas-de-envio-de-datos-en-formulario-diferencias-y-ventajas-ejemplos-cu00721b&catid=69&Itemid=192)

#### 4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme) /URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.

##### Poco

**URI** un sistema que ha sido diseñado para nombrar y localizar los recursos de la web de manera unívoca.

La estructura genérica de un URI se describe a continuación:

**[protocolo][nombre\_del\_equipo][ruta\_directorios][archivo]**

**Protocolo:** también llamado esquema URL, este componente especifica el protocolo que se requiere utilizar para acceder al documento o recurso.

**Nombre del equipo:** (habitualmente un nombre de dominio o una dirección IP) donde el contenido está alojado.

**Ruta de directorios:** una secuencia de directorios separados por barras que define la ruta a seguir en la computadora para llegar al archivo.

**Archivo:** El nombre del archivo que contiene al recurso.

El siguiente ejemplo muestra un URI con todas sus partes etiquetadas.

<small>Protocolo</small>	<small>Nombre de la computadora</small>	<small>Directorios</small>	<small>Archivo</small>
http://	www.htmlquick.com	/reference/tags/	a.html

Sin embargo, no es requerida la utilización de todas estas componentes para conformar un URI válido. Muchos URIs pueden tener sólo algunas de estas partes presentes (o ninguna) y aún ser válidas. La siguiente lista muestra algunos ejemplos de URIs válidos.

- |   |                            |
|---|----------------------------|
| <small>Nombre de la computadora</small> | <small>Directorios</small> |
| 195.156.123.1                           | /backup/                   |

 (sin protocolo; sin archivo)
- |                          |   |
|--------------------------|---|
| <small>Protocolo</small> | <small>Nombre de la computadora</small> |
| http://                  | yeahyeah.com                            |

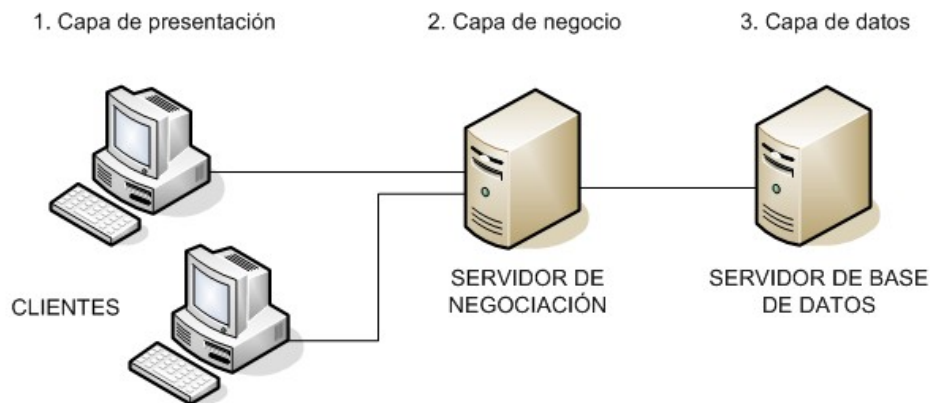
 / (sin directorios; sin archivo)
- |                            |                        |
|----------------------------|------------------------|
| <small>Directorios</small> | <small>Archivo</small> |
| /john/trabajo/             | reporte.odt            |

 (sin protocolo; sin nombre de la computadora)
- |                        |
|------------------------|
| <small>Archivo</small> |
| index.html             |

 (sin protocolo; sin nombre de computadora; sin directorios)
- / (sin protocolo; sin nombre de computadora; sin directorios; sin archivo)

## 5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.

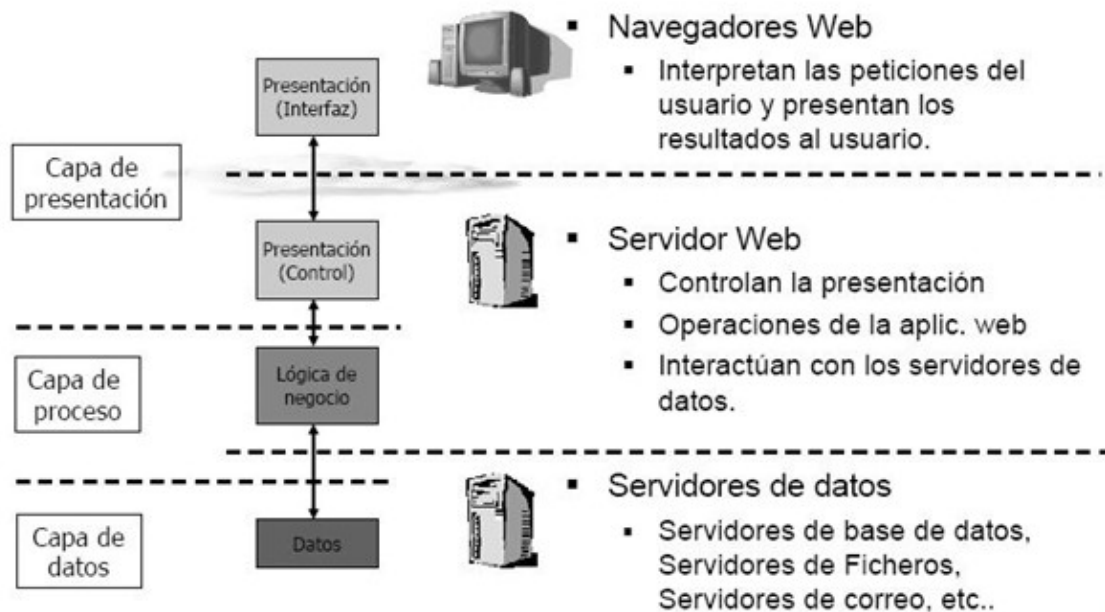
El **modelo de desarrollo de aplicaciones multicapa** es un modelo de desarrollo de software el cual nos permite separar las partes que componen una aplicación para que sea más sencillo el desarrollo de la misma. Se divide en la **capa de presentación**, la **capa de negocio** y la **capa de datos**.



- **Capa de presentación:** es la capa que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. También es conocida como interfaz gráfica y debe tener la característica de ser «amigable» (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio. Sus características principales son:
  - Recoge la información del usuario y la envía al servidor (cliente)
  - Manda información a la capa de proceso para su procesamiento
  - Recibe los resultados de la capa de proceso
  - Generan la presentación
  - Visualizan la presentación al usuario (cliente)
- **Capa de negocio:** donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados. Si la aplicación se comunica con otros sistemas que actúan en conjunto, lo hace mediante esta capa. También se comunica con la capa de datos para obtener información existente o ingresar nuevos datos. Sus principales características son:
  - Recibe la entrada de datos de la capa de presentación
  - Interactúa con la capa de datos para realizar operaciones
  - Manda los resultados procesados a la capa de presentación
- **Capa de datos:** donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento

de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. Para ello se utilizan APIs como pueden ser ADO o JDBC, estas son de bajo nivel para acceder a las diferentes fuentes de datos. Sus principales características son:

- Almacena los datos
- Recupera datos
- Mantiene los datos
- Asegura la integridad de los datos



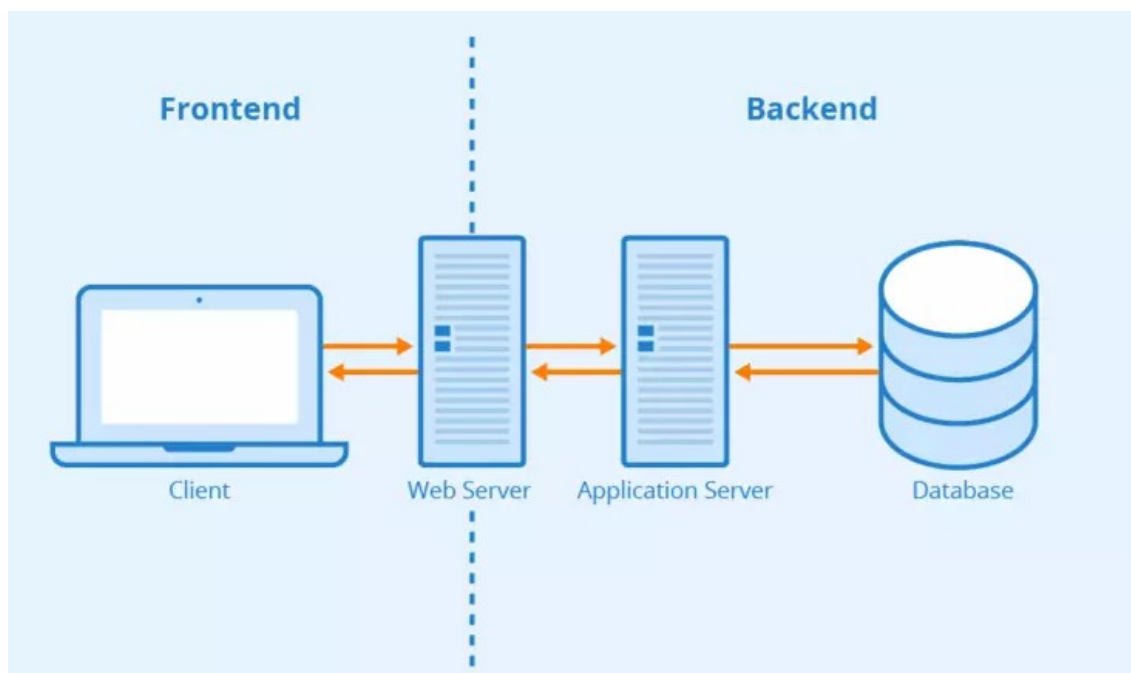
Enlaces de interés:

- [Modelo de desarrollo multicapa](#)
- [Arquitectura de las aplicaciones web](#)

## 6. Modelo de división funcional front-end / back-end para aplicaciones web.

El modelo de división funcional divide una aplicación web en dos partes. La primera es el **front-end**, que es todo lo que ve el usuario y con lo que puede interactuar. La segunda es el **back-end**, que se encarga de tomar estas interacciones del usuario y comunicarlás con la base de datos a través del servidor.

- **Front-end:** Es la parte de una aplicación web a la que todos los usuarios normales pueden acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.
- **Back-end:** Es la capa de acceso a datos de una aplicación web y también la del lado del servidor que utilizan los usuarios especiales como administradores, gestores, directores, etc.... Esta capa no es accesible por los usuarios, además contiene la lógica de la aplicación que maneja los datos. Está enfocado en trabajar con bases de datos y con el servidor. Todo lo que sucede antes de que se renderice una página así como toda la comunicación con la base de datos es trabajo del back-end.



Enlaces de interés:

- [Front-end y Back-end](#)
- [Que es Backend y FrontEnd](#)

## 7. Página web estática – página web dinámica – aplicación web – mashup .

- **Página web estática:** Es una página web en la cual el contenido no varía. El usuario se limita a obtener dicha información sin poder interactuar con la página web.

Una página web estática está compuesta por archivos HTML individuales por cada página que son pre-generados y presentados al usuario a través del navegador de la misma forma.

Como una página web estática básica está compuesta por elementos como títulos, cuadros de textos, etiquetas, imágenes y otros elementos multimedia, un usuario solo puede interactuar con una página web estática a través de lo que permiten los elementos HTML, por ejemplo, haciendo clic en enlaces, botones o rellenando formularios como el clásico formulario de suscripción.

No son tan complejos técnicamente como un sitio web dinámico, pero tampoco son tan versátiles y efectivos cuando se trata de entregar funcionalidad.

### Ventajas de una página web estática

- El coste inicial de una página web estática puede ser mucho menor que al de una dinámica.
- Son muy flexibles cuando se trata del diseño.
- Los tiempos de carga son muy rápidos.

### Desventajas de una página web estática

- Una página web estática puede ser más difícil de actualizar.
- Agregar contenido a la página web o realizar actualizaciones puede incurrir en costos adicionales.
- Agregar nuevas páginas o funcionalidades a una web estática puede ser más difícil que hacerlo para una web dinámica.

# Ejemplos de páginas web estáticas

Un ejemplo sencillo de cómo es una página web estática, es el siguiente:

```
<head>
<title>Ejemplo página web estática</title>
</head>
<body>
La fecha de hoy es Enero 1, 2020
</body>
</html>
```

- **Página web dinámica:** Pagina web que permite crear aplicaciones dentro de la propia web, dando una mayor interactividad con el cliente.

Una página web dinámica incluye aspectos que se caracterizan por la interactividad y la funcionalidad, por ejemplo, los usuarios pueden interactuar con la información que se presenta en la página gracias a las instrucciones creadas a través de los lenguajes de programación y la base de datos sobre la que está construida.

Los sitios web dinámicos basan su comportamiento y funcionalidad en dos tipos de programación, front-end (del lado del cliente) y back-end (del lado del servidor). Las instrucciones del lado del cliente es código JavaScript que se ejecuta en el navegador. Mientras que las instrucciones que se ejecutan del lado del servidor son instrucciones escritas en lenguajes de scripting o programación, como ASP.Net, PHP, Python, etc. y que son ejecutadas para crear lo que el usuario ha solicitado en su interacción con la página.

Una vez ejecutadas las instrucciones en el servidor, un nuevo HTTP response se envía al navegador del usuario para mostrarle lo que ha solicitado. El resultado final es el mismo que en un sitio web estático: una página HTML que el usuario ve desde el navegador.

## Ventajas de una página web dinámica

- Puede gestionar información a través de bases de datos.
- El contenido se puede gestionar a través de un CMS.
- El coste de mantenimiento es menor.

## Desventajas de una página web dinámica

- Pueden existir limitaciones en el diseño.
- Puede involucrar altos costos de construcción iniciales.



- **Aplicación web:** Es un conjunto de herramientas que los usuarios pueden utilizar accediendo a un servidor web mediante un navegador, dicha página almacenara la información del usuario en su servidor.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Se pueden dividir en tres capas:

- Capa del navegador o presentación
- Capa del servidor o negocio
- Capa de persistencia o de datos.



- **Mashup:** Es una aplicación web híbrida que combina servicios de diversas páginas web en una sola para ofrecer un nuevo servicio. Típicamente los mashups se basan en utilizar contenido de terceros a través de fuentes o interfaces públicas.

#### Ventajas de las Mashups

- Generación rápida de contenido.



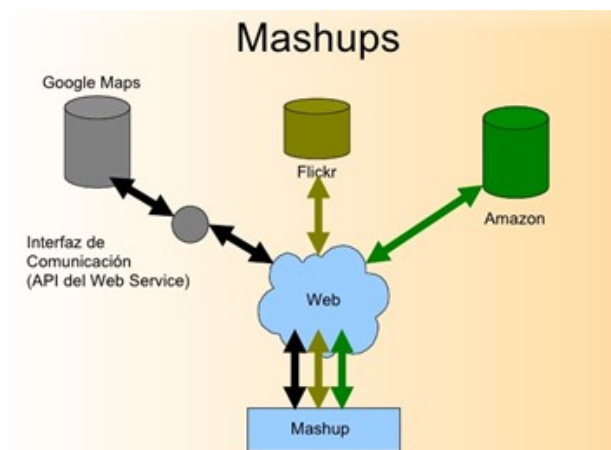
La integración con soluciones ya existentes permite acortar los tiempos de desarrollo. Lo que antiguamente podía llevar semanas de trabajo ahora se puede implementar en tan solo unos minutos.

- Facilidad de uso.

Los requisitos de diseño de las mismas son mínimos por lo que es más importante la creatividad que el perfil técnico.

- Sinergias.

Las marcas pueden crear experiencias innovadoras sin que los usuarios tengan que salir de sus páginas web. A su vez, las fuentes de mashup logran una mayor visibilidad que si sus servicios solo se ofrecieran a través de su página web.



Enlaces:

[Webs Dinamicas vs Webs Estaticas](#)

## 8. Componentes de una aplicación web.

Una aplicación web está compuesta por

**Un navegador web** (Cliente HTTP): es un programa o aplicación que interpreta distintos tipos de archivos, lenguajes de programación, texto y recursos multimedia incrustados a partir de código HTML y CSS, el cual es interpretado para que el cliente pueda navegar e interactuar por la web. Es el componente que se encarga de mandar las solicitudes al servidor para que las procese, el cual localiza mediante su URL.

Algunos ejemplos de navegadores web son [Google Chrome](#), [Firefox](#), [Edge](#)...

**Un servidor web**: sobre el que las aplicaciones web se van a ejecutar. Se comunica con el navegador del cliente proporcionándole datos que responden a sus solicitudes. Este las recibe mediante el protocolo HTTP a través del puerto TCP 80, o mediante el protocolo HTTPS a través del puerto TCP 443.

Los servidores de aplicaciones web contienen:

- Conectores web-servidor.
- Lenguajes de programación.
- Librerías de runtime.
- Conectores a la base de datos.
- El código necesario para desplegar, configurar, mantener y conectar los anteriores componentes en un host.

Estos se ejecutan detrás de servidores web como pueden ser *Apache HTTP* o *Microsoft Internet Information Services* y casi siempre están conectados a una base de datos.

Algunos ejemplos son los **Java Application Servers** o el **.NET Framework de Microsoft**.

**Un sistema gestor de bases de datos**: nos permite la organización de la información mediante el uso de tablas, índices y registros. Se encarga de la persistencia de datos de la aplicación para que, al cerrarla y volver a abrirla, la información no se pierda.

*Desde un punto de vista de hardware es un equipo informático especializado en proporcionar consultas a clientes remotos o locales que solicitan información o realizan modificaciones a los registros y tablas que existen dentro de dicha base de datos.*

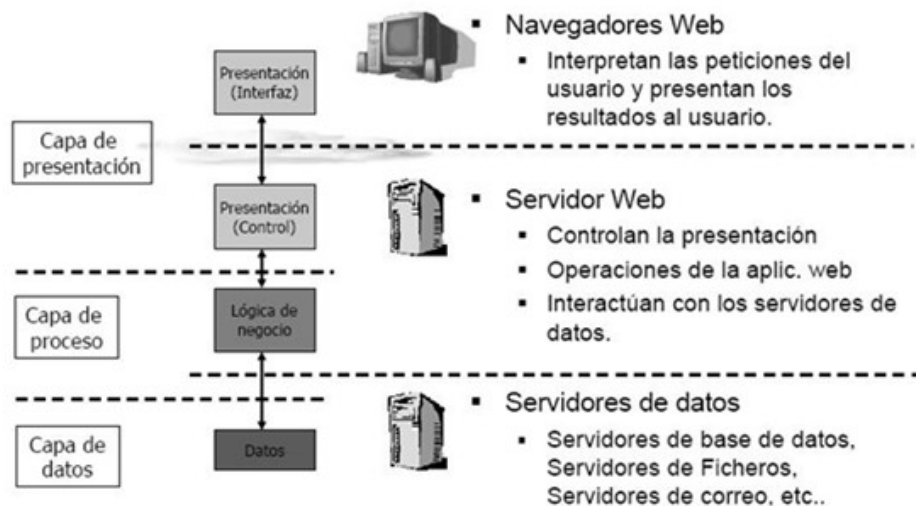
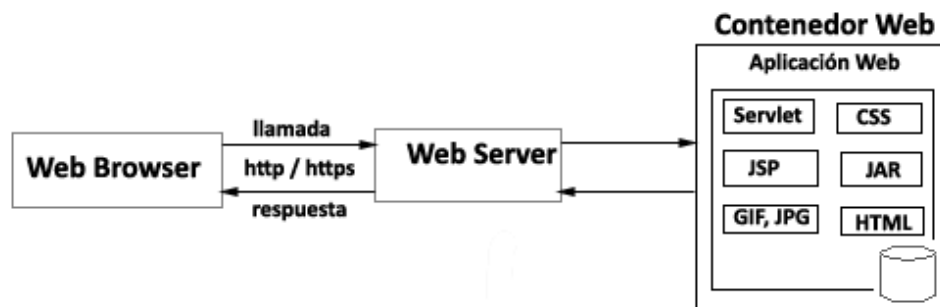
La mayoría de estos servidores responden a un lenguaje basado en Query, donde cada base de datos entiende su propio lenguaje y convierte cada query a una forma legible para el servidor, ejecutándolo y proporcionando una respuesta.

Algunos ejemplos de sistemas gestores de bases de datos son [MySQL Server](#) o [MongoDB Server](#) entre otros.

**Un módulo para ejecutar el código**: será el encargado interpretar el lenguaje utilizado y así generar la página web resultante.

## El código para desarrollar la aplicación:

- Lenguaje de marcas: HTML, XML, CSS..
- Lenguajes de scripting: PHP, Perl, Python..



<https://elpuig.xeill.net/Members/vcarceler/asix-m09/uf1/nf1/a1>

<https://www.webempresa.com/hosting/que-es-servidor-web.html>

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-un-navegador/>

<https://blog.infranetworking.com/servidor-base-de-datos/>

[#Que\\_es\\_un\\_Servidor\\_de\\_Base\\_de\\_Datos](#)

## 9. Programas ejecutados en el lado del cliente y del servidor. Lenguajes de programación utilizados en cada caso.

**En el lado del cliente** se utilizan navegadores —Google Chrome, Mozilla Firefox, Safari, etc.—, que se comunican con los servidores e interpretan lenguajes tipados como HTML y CSS, y JavaScript o VBScript —de Visual Basic— que son lenguajes interpretados, del que derivan otros como TypeScript. Los navegadores, a su vez, pueden contar con extensiones añadidas por el usuario, como traductores o buscadores de color.

Lenguajes y tecnologías del lado de cliente
<b>HTML:</b> Lenguaje de marcas que indica la estructura de los contenidos de una página.
<b>CSS:</b> Da estilo a las páginas añadiendo márgenes, colores o tipografías.
<b>JavaScript:</b> Lenguaje que crea pequeños programas que realizan acciones dentro de una página web. Se insertan en HTML mediante la etiqueta <script>
<b>Applets de Java:</b> Están programados en Java y también son código ejecutable en HTML. A diferencia de JavaScript, tienen más independencia del navegador y el sistema operativo
<b>VBScript (Visual Basic Script):</b> Lenguaje solo compatible con Internet Explorer, se desaconseja su utilización. Es una versión limitada de Visual Basic.
<b>Flash:</b> No es un lenguaje, pero tenía las mismas funcionalidades que JavaScript. Funcionaba con un plug-in en el navegador.

**En el lado del servidor** se encuentran los servidores y las bases de datos.

Los **servidores** almacenan las aplicaciones que, a petición del cliente, deberán, exactamente servir; por ejemplo, un cliente con un navegador que desee acceder a una página en concreto o a un programador que necesite recuperar código almacenado en un repositorio.

Lenguajes y tecnologías del lado del servidor
<b>CGI:</b> El sistema más antiguo para páginas dinámicas para seguidor. Algo desfasado en la actualidad. Se suelen escribir en Perl, aunque también en C, C++ o Visual Basic.
<b>Perl:</b> Lenguaje interpretado y extensible, se conecta muy bien con otros lenguajes.
<b>ASP:</b> Tecnología de Microsoft para páginas dinámicas de servidor. Se escribe en la propia página web con VBScript o JScript (Javascript de Microsoft)
<b>PHP (Hypertext Preprocessor):</b> Lenguaje rápido e independiente de la plataforma con una gran librería de funciones y una extensa documentación
<b>JSP (JavaServer Pages):</b> Tecnología que permite crear páginas de servidor en lenguaje Java. Es multiplataforma.
<b>Python:</b> Lenguaje interpretado que se utiliza cada vez más. Destaca por su sencillez y legibilidad.

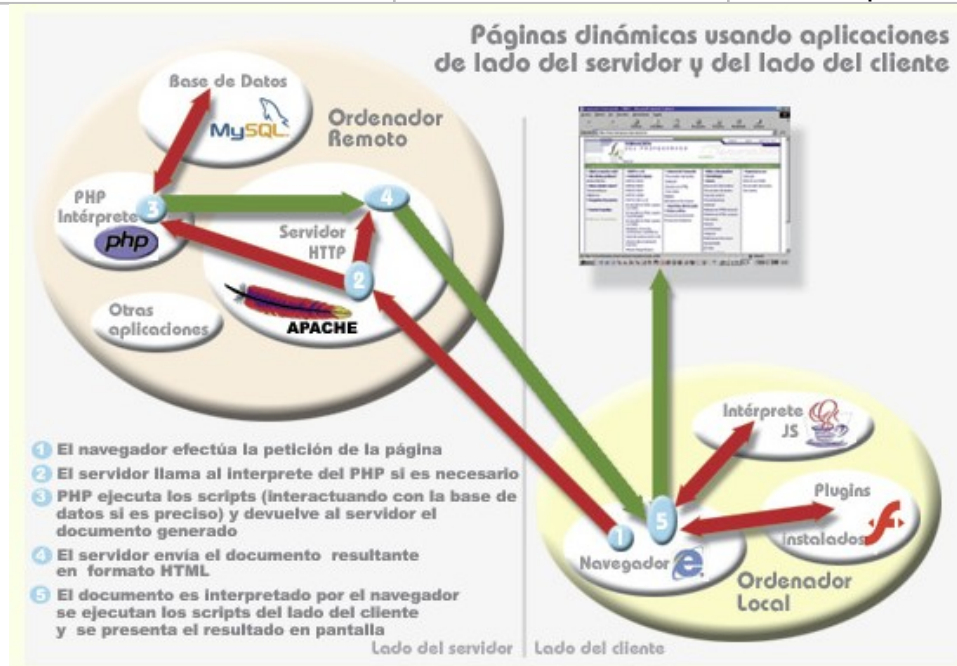
Las **bases de datos** almacenan información necesaria para las aplicaciones. Los sistemas

gestores de bases de datos las administran mediante queries. Ejemplos son MariaDB, MySQL o MongoDB, que se diferencia de las anteriores en que su almacenamiento de datos no es relacional.

Ejemplos de lenguajes de consultas pueden ser cualquier variante de SQL, XQuery u OQL.

Las consultas que realizan las bases de datos se encuadran en cuatro categorías;

Categoría	Descripción	Instrucciones
DDL—Data Definition Language	Especifican el esquema de la base de datos.	-create -alter -drop -rename
DML —Data Manipulation Language	Modifican la información almacenada.	-insert -update -delete -select
DCL —Data Control Language	Controlan los permisos de acceso a los datos.	-grant -revoke
TCL —Transaction Control Language	Llevan a cabo grupos de tareas.	-commit -rollback -savepoint



[https://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_server\\_software](https://en.wikipedia.org/wiki/Comparison_of_web_server_software)

<https://www.indeed.com/career-advice/career-development/database-languages>

## 10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).

**PHP** (Hypertext Preprocessor), de los más utilizados actualmente, es de código abierto y muy rápido. Es fácil de aprender y multiplataforma. Permite establecer conexión con todo tipo de bases de datos, como Oracle o MySQL. El servidor, no obstante, no delega tareas al lado del cliente.

**ASP** evolucionó en ASP.NET. Tiene la ventaja que los sitios web pueden ejecutarse en todos los navegadores. Permite un control de usuario personalizado, y división entre capa de diseño y código. Sin embargo, consume muchos recursos.

**PERL** es muy útil en procesamiento de textos y archivos, y está disponible en muchas plataformas y sistemas operativos. El desarrollo de aplicaciones en él es muy rápido. Como desventaja, es algo lento en determinadas aplicaciones, como la programación a bajo nivel.

**Ruby**, al igual que PHP, es fácil de aprender, pero consume bastantes recursos. También resulta muy limitado, pues está muy basado en plantillas y estándares.

Otros lenguajes pueden ser Java, C, C++ o Python.

	ASP.NET	Java	Ruby
Desarrollado por	Microsoft	Sun Microsystems	Yukihiro Matsumoto et al.
Licencia	propietario	GNU GPL	BSD
Año de publicación	2002	1995	1995
Plataforma	Windows	independiente	independiente
Paradigmas de programación	Orientado a objetos	Orientado a objetos	multiparadigma

	Perl	PHP	Python
Desarrollado por	Larry Wall, et al.	Rasmus Lerdorf	Guido van Rossum, Python Software Foundation
Licencia	GNU GPL y licencia artística	Licencia PHP, entre otras	Python Software Foundation License
Año de publicación	1987	1995	1991
Plataforma	independiente	independiente	independiente
Paradigmas de programación	procedural, modular, en parte orientado a objetos	imperativo, funcional, orientado a objetos	multiparadigma

Ilustraciones 1 y 2. Tablas comparativas de lenguajes de programación del lado del servidor.  
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/lenguajes-del-lado-servidor-o-del-cliente-diferencias/>

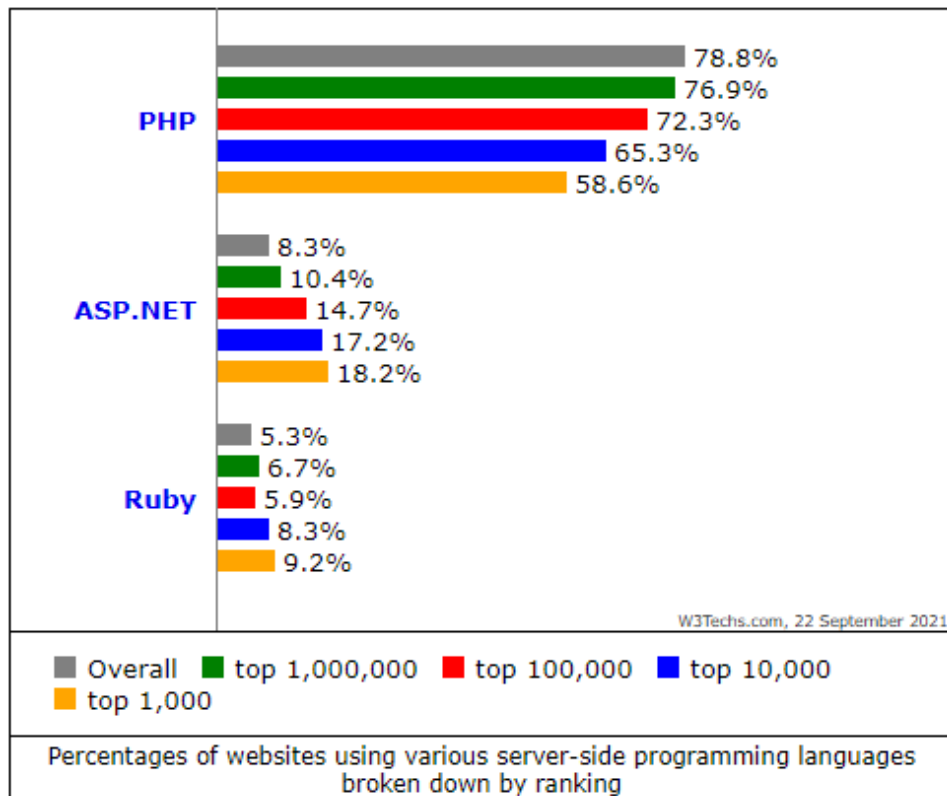
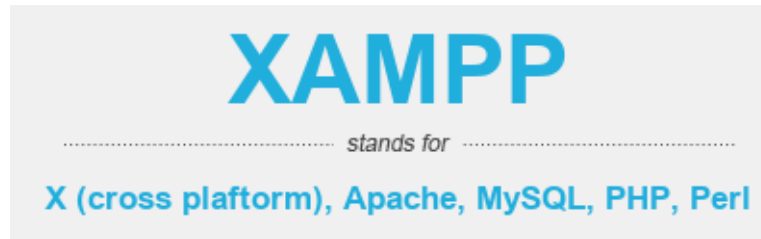


Ilustración 3. Porcentajes de uso de los principales lenguajes de programación del lado del servidor.  
[https://w3techs.com/technologies/cross/programming\\_language/ranking](https://w3techs.com/technologies/cross/programming_language/ranking)



## 11. Características y posibilidades de desarrollo de una plataforma XAMPP.

XAMPP permite la gestión de bases de datos MySQL y MariaDB, el servidor web Apache, y además tiene intérpretes de PHP y Perl. Se utiliza para probar aplicaciones web y como servidor para las mismas.



*Ilustración 4. Significado de las siglas de XAMPP, las aplicaciones utilizadas en la arquitectura AMP: servidor web, sistema gestor de base de datos, e intérpretes de PHP y Perl. <https://ythi.net/abbreviations/english/what-does-xampp-mean-what-is-the-full-form-of-xampp/>*

Y el cuatro componentes principales de XAMPP. Son:

- **Apache:** es el servidor web más popular en internet, Apache es la aplicación de servidor web que procesa y entrega el contenido web a tu ordenador.
- **MySQL:** es de código abierto, es el sistema gestor de base de datos más popular. Alimenta desde websites de aficionados hasta plataformas profesionales como WordPress. Cada aplicación web, que sea simple o compleja, requiere una base de datos para almacenar los datos que recoge.
- **PHP:** son las siglas de Preprocesador de Hipertexto. Es un lenguaje de script del lado del servidor que hace funcionar algunas de las páginas web más famosas del mundo, incluyendo WordPress y Facebook. Es de código abierto, relativamente fácil de aprender, y funciona perfectamente con MySQL, lo que lo ha hecho una opción muy popular para los desarrolladores web. Puede aprender a dominar PHP con este curso sobre los Fundamentos de Programación PHP.
- **Perl:** es un lenguaje de programación dinámico y de alto nivel que se usa ampliamente en la programación de redes, la administración de sistemas, etc. Aunque menos popular para el desarrollo web, Perl tiene un montón de aplicaciones específicas.

[Link de descarga de XAMPP](#)

[Tutorial de instalación y configuración](#)

[http://ventagchamilo.blogspot.com/p/blog-page\\_15.html](http://ventagchamilo.blogspot.com/p/blog-page_15.html)

12. En qué casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación.

La JVM – Java Virtual Machine - es el entorno en que se ejecutan los programas Java. Un programa en Java puede ser ejecutado en cualquier sistema operativo que tenga solamente la JVM.

El JDK — Java Development Kit — es la herramienta utilizada para desarrollar programas en Java. No es necesario que el usuario lo tenga para poder ejecutarlo.

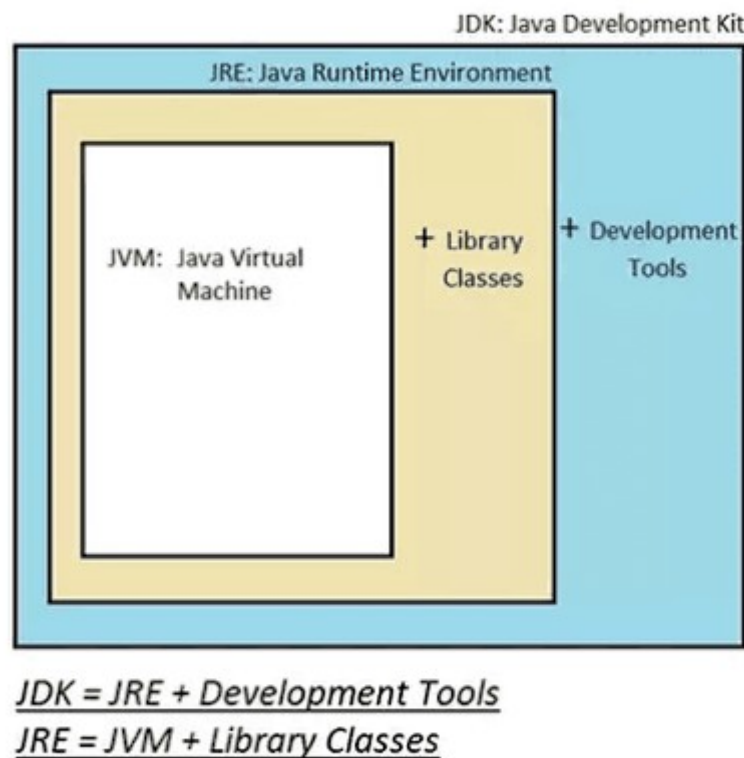


Ilustración 5. Diferencias entre JDK, JRE y JVM. <https://javadesdecero.es/fundamentos/diferencias-jdk-jre-jvm/>

Más Información: <https://javadesdecero.es/fundamentos/como-funciona-maquina-virtual/>

### 13. IDE más utilizados (características y grado de implantación actual).

Los IDE más utilizados son **Visual Studio**, **Eclipse**, **IntelliJ**, **Android Studio** y **NetBeans**.

Se puede diferenciar entre dos categorías de IDEs, tanto como para un Desarrollador web como para un Desarrollador móvil. En ambas categorías se usan prácticamente los mismos.

En general todos los IDEs cuentan con una interfaz gráfica de usuario para ayudar a los desarrolladores a crear aplicaciones de software, lo cual las hace mas sencillas a la hora de su uso.

Los cinco son de código abierto y son compatibles tanto con Windows, MacOS y Linux.

Todos ellos soportan múltiples lenguajes de programación como Java, PHP, C++, Python..., aunque algunos deben ser descargados vía plugin. También suelen incluir un intérprete o un compilador de código.

Android Studio es el más limitado en este sentido ya que permite Kotlin, Java y C++.

Los cinco tienen ayuda a la escritura, depuración de errores y soporte de control de versiones.

Mas información: [22 Entornos de Desarrollo Integrado más Populares 2021 \(diarlu.com\)](https://diarlu.com/2021/07/22/entornos-de-desarrollo-integrado-mas-populares-2021/)

Tutorial para instalar NetBeans:

<https://www.locurainformaticadigital.com/2019/07/21/descargar-e-instalar-netbeans-11-windows/>

<https://www.solvetic.com/tutoriales/article/8578-como-instalar-eclipse-ide-windows-10/>

## 14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).

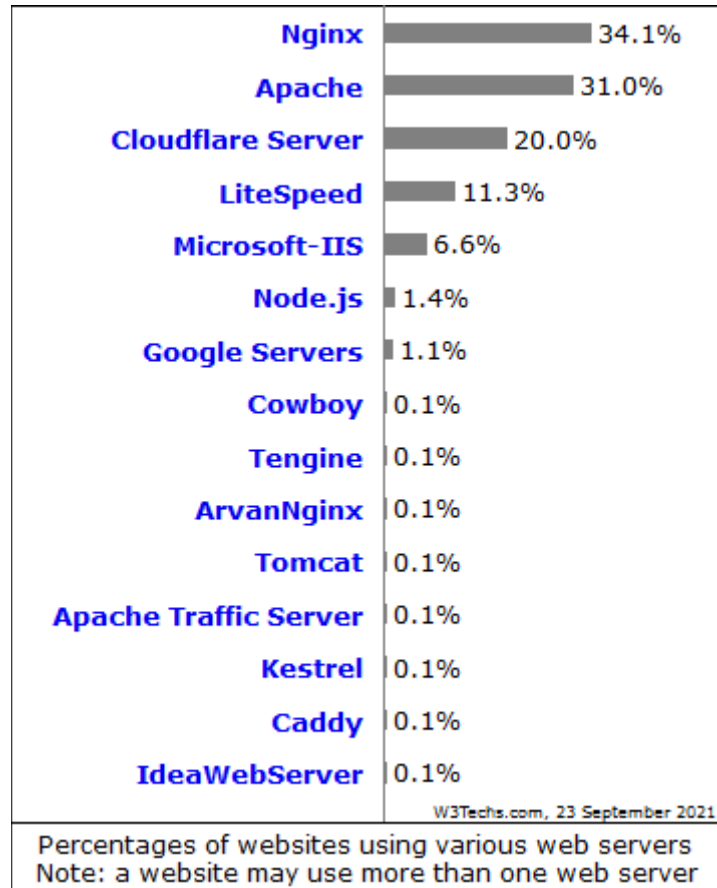
Los servidores HTTP/HTTPS más utilizados a día de hoy son **Nginx** en primer lugar gracias a que la configuración es sencilla y más personalizable, le siguen **Apache HTTP**, **Cloudflare Server**, **LiteSpeed** y **Microsoft-IIS**.

Los lenguajes de programación que se pueden utilizar en el lado del servidor en los siguientes servidores son:

- **PHP:** PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web del lado del servidor. PHP puede ser desplegado en la mayoría de los servidores web y en todos los sistemas operativos y plataformas sin costo alguno.
- **Perl:** Perl es un lenguaje de programación. Perl toma características del lenguaje C, del lenguaje interpretado bourne shell, AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.
- **Python:** Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y la programación funcional.
- **Ruby:** Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU. A parte, es un lenguaje de programación interpretado en una sola pasada.

	Lanzamiento	Lenguaje de programación	Desarrollador	Sistemas operativos	Ejemplos
<b>Nginx</b>	2004	C	Igor Sysoev	Varios, entre ellos Linux, macOS y Microsoft Windows	WordPress, Netflix, Hulu, GitHub
<b>Apache HTTP</b>	1995	C, XML	Apache Software Foundation	Sistemas tipo Unix, Microsoft Windows, OpenVMS	LinkedIn, The New York Times, Glovo, Trivago, IBM, Adobe, Facebook, eBay
<b>Cloudflare Server</b>					
<b>LiteSpeed</b>	2003	C, C++	LiteSpeed Technologies	Varios de Linux, FreeBSD	Varias empresas estadounidenses y de Reino Unido, como Facsimile Communications Inc. O

					Complete I.T. Ltd.
<b>Microsoft-IIS</b>	1995	C++	Microsoft	Windows NT	Aplicaciones de Microsoft



Enlaces de interés:

- [Nginx](#)
- [Apache HTTP Server](#)
- [Cloudflare Server](#)
- [LiteSpeed Web Server](#)
- [Microsoft-IIS](#)
- [PHP](#)
- [Perl](#)
- [Python](#)
- [Ruby](#)

## 15. Apache HTTP vs Apache Tomcat

**Apache HTTP y Apache Tomcat son ambos servidores para aplicaciones web. La principal diferencia que tienen es su orientación hacia los diferentes lenguajes de programación.**

**Apache Tomcat** proporciona páginas web dinámicas basadas en aplicaciones Java, que cambian dependiendo del momento y del usuario. Apache HTTP proporciona cualquier contenido como texto, vídeo o imágenes en páginas HTML. Sirve como intérprete de PHP, Perl y Python.

**Apache HTTP** se ubicaría en el lado del cliente y Tomcat se encontraría en el lado del servidor, más concretamente en la capa lógica, interactuando con la capa de datos.

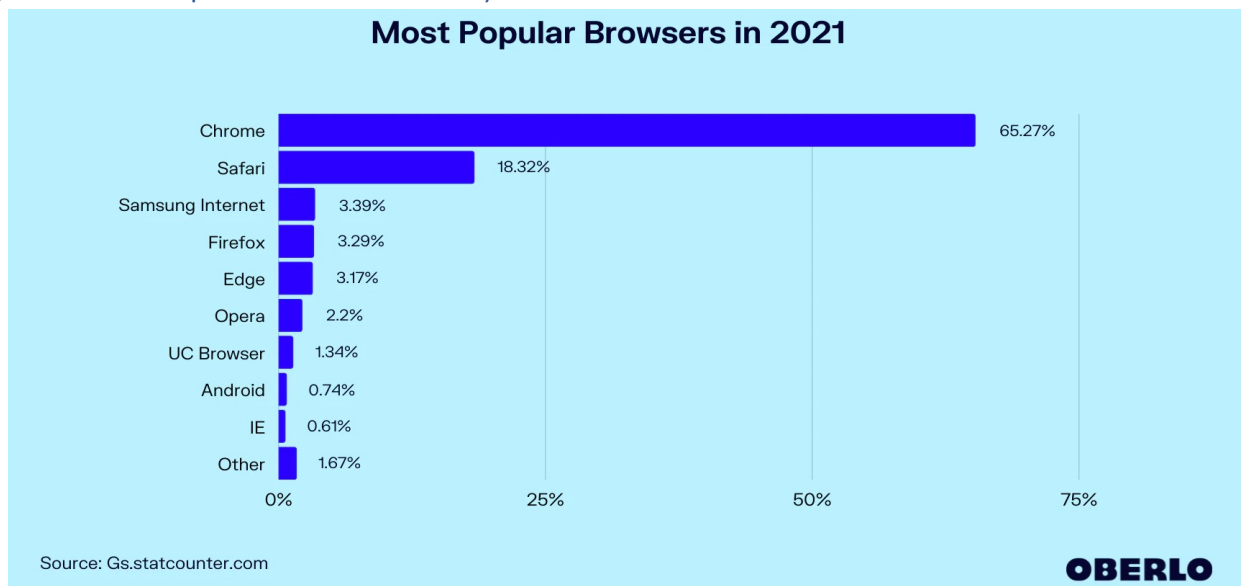
Algunas de las diferencias son:

- La diferencia fundamental es que Tomcat proporciona contenido dinámico en Java, mientras que el propósito principal del servidor web Apache es simplemente servir contenido estático como HTML, imágenes, audio y texto.
- Tomcat se creó específicamente para aplicaciones Java, mientras que Apache es un servidor HTTP de propósito general. Puedes utilizar Apache junto con diferentes lenguajes de programación (PHP, Python, Perl, etc.) con la ayuda del módulo de Apache apropiado al lenguaje. (mod\_php, mod\_python, mod\_perl, etc...)
- Tomcat también es menos configurable en comparación con otros servidores web. Por ejemplo, para ejecutar WordPress, la mejor opción es un servidor HTTP de propósito general como Apache o NGINX.
- El servidor web **Apache HTTP** tiene una carpeta especial llamada /htdocs. Los recursos colocados en esta carpeta se vuelven accesibles para las solicitudes entrantes. En cambio, en las aplicaciones implementadas en el servidor **Apache Tomcat** deben empaquetarse como archivos de aplicaciones web o archivos .war. El formato de estos archivos .war, junto con los descriptores de implementación que deben contener, están rígidamente definidos por la API de Servlet y JSP.
- El competidor más grande de Apache http es Nginx HTTP (Nginx casualmente también puede correr aplicaciones basadas en Java).  
Apache Tomcat compite contra otras Java EE y servidores Jakarta EE.
- En cuanto a velocidad, Apache HTTP es más rápido que Apache Tomcat en lo que a las páginas web estáticas se refiere.

Enlaces de interés:

- [Apache Tomcat vs Apache HTTP](#)
- [Apache HTTP](#)
- [Apache Tomcat](#)

## 16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).



**Google Chrome:** Rápido, sencillo e intuitivo. Tiene una fuerte seguridad, con actualizaciones automáticas. Como inconveniente consume demasiados recursos del PC.

**Safari:** Es el navegador que se suele utilizar en los Mac. Su sencillez le da estabilidad y eficiencia, aunque tampoco es el más rápido. La falta de actualizaciones hace que tenga menos seguridad.

**Samsung Internet:** Común en dispositivos portátiles y Smart TV. Rápido y sencillo. Se caracteriza por su bloqueo de contenido como los anuncios.

**Mozilla Firefox:** Es uno de los más veteranos. Tiene actualizaciones constantes y su funcionamiento es estable. Es fácil de personalizar con extensiones.

**Microsoft Edge:** Es la alternativa más modernizada de Microsoft a Internet Explorer. Está más actualizado y es más rápido. Consume pocos recursos del PC.

**Opera:** También es veterano. Similar a Google Chrome, con un apartado para guardar páginas web. Tan personalizable como Chrome o Firefox.

**UC Browser:** Utilizado en dispositivos móviles, se puede adaptar a sus distintos sistemas operativos. Su gestión de recursos permite aumentar la velocidad de carga.

**Navegador Android:** Se utilizaba en dispositivos con Android antes de que se usara Google Chrome. Sencillo y directo.

**Internet Explorer:** Está actualmente obsoleto y por lo tanto tiene muchos problemas de seguridad. Su velocidad tampoco destaca.

<http://tublogtecnologico.com/navegadores-web/>

<https://andro4all.com/2021/03/por-que-deberias-usar-el-navegador-de-samsung-en-vez-de-google-chrome-2021-03-11>

<https://android-browser.uptodown.com/android>

<https://hddrecover.ru/es/browsers/main-features-of-uc-browser-for-android/>



## 17. Generadores de documentación HTML (PHPDoc):

### PHPDocumentor, ApiGen, ...

**PHPDoc:** es un estándar que permite trasladar el javadoc al código PHP y generar comentarios en el lenguaje PHP para documentar un programa y exponer el funcionamiento de sus componentes.

Principalmente se documenta el papel que desempeñan las funciones, las clases (*estas se suelen pasar en un documento aparte, normalmente en HTML, para que lo lean quienes las van a utilizar.*), los algoritmos y los pasos dados para programarlos, y por último, la toma de decisiones en cuanto a rendimiento.

**PHPDocumentor** es una de las herramientas más utilizadas que permiten generar documentación de forma automática a partir del código fuente para PHP y otros más.

La documentación se distribuye en bloques DocBlock. Estos bloques siempre se colocan justo antes del elemento al que documentan, por ejemplo:

```
/**
 * Descripción breve (una línea)
 * Descripción extensa. Todas las líneas que
 * Todas las líneas comienzan con *
 * <- Esta línea es ignorada
 * Este DocBlock documenta la función suma()
 *
function suma()
{
    ...
}
```

Los elementos que pueden ser documentados son: **define**, **function**, **class**, **class var**, **include/require/include\_once/require\_once** y **global variables**.

Puede incluir documentación global a nivel de fichero y clase mediante la marca **@package**.

Otras etiquetas pueden ser:

<b>@access</b>	Usado como modificador de acceso en el que no se documentan los elementos privados.
<b>@author</b>	Para mostrar el autor del código.
<b>@copyright</b>	Información sobre derechos.
<b>@deprecated</b>	Para indicar que un elemento no debería utilizarse.
<b>@example</b>	Para especificar la ruta hasta un fichero con código PHP
<b>@ignore</b>	Para que algo no se documente.
<b>@internal</b>	Para mantener documentación privada.
<b>@link</b>	Para enlaces.
<b>@since</b>	Para datar elementos.

@version	Para señalar la versión del elemento
----------	--------------------------------------

**Doxygen** sirve para documentar otros lenguajes de programación, pero también se puede utilizar para PHP. Puede generar diferentes archivos de salidas además de HTML, como PDF o PostScript. Estos muestran las clases ordenadas en una lista por jerarquías, para mantener la fidelidad del código.

**PHPXref** es otro generador de documentación que se instala en el pc. Como característico principal permite vincular cada llamada a un elemento con el punto de origen en el que fue definido, lo que facilita la búsqueda en proyectos de gran tamaño.

**ApiGen** también genera documentación PHP, conocido como uno de los más sencillos.



<http://www.epsilon-eridani.com/cubic/ap/cubic.php/doc/phpDocumentor---documentacion-para-codigo-PHP-246.html>

<https://zonaphp.com/generadores-de-documentacion/>

## 18. Repositorios de software – sistemas de control de versiones.

Los programas para control de versiones son un conjunto de aplicaciones que gestionan ágilmente los cambios en el código fuente de los programas y poder revertirlos. Por ejemplo:

- **Git:** *una de las mejores herramientas de control de versiones disponible en el mercado actual. Es un modelo de repositorio distribuido compatible con sistemas y protocolos existentes como HTTP, FTP, SSH y es capaz de manejar eficientemente proyectos pequeños a grandes.*

<https://git-scm.com/>



- **CVS:** *es otro sistema de control de versiones muy popular. Es un modelo de repositorio cliente-servidor donde varios desarrolladores pueden trabajar en el mismo proyecto en paralelo. El cliente CVS mantendrá actualizada la copia de trabajo del archivo y requiere intervención manual sólo cuando ocurre un conflicto de edición.*

<https://www.nongnu.org/cvs/>



- **Apache Subversion:** *abreviado como SVN, apunta a ser el sucesor más adecuado. Es un modelo de repositorio cliente-servidor donde los directorios están versionados junto con las operaciones de copia, eliminación, movimiento y cambio de nombre.*

<https://subversion.apache.org/>



- **Mercurial:** *es una herramienta distribuida de control de versiones que está escrita en Python y destinada a desarrolladores de software. Los sistemas operativos que admite son similares a Unix, Windows y macOS. Tiene un alto rendimiento y escalabilidad con capacidades avanzadas de ramificación y fusión y un desarrollo colaborativo totalmente distribuido. Además, posee una interfaz web integrada.*

<https://www.mercurial-scm.org/>



- **Monotone:** *está escrito en C++ y es una herramienta para el control de versiones distribuido. El sistema operativo que admite incluye Unix, Linux, BSD, Mac OS X y Windows. Brinda un buen apoyo para la internacionalización y localización. Además, utiliza un protocolo personalizado muy eficiente y robusto llamado Netsync.*

<https://www.monotone.ca/>

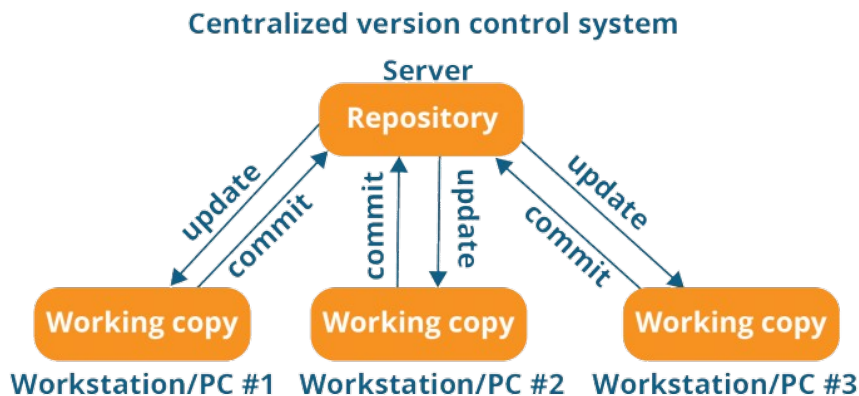


Ilustración 6. Sistema de control de versiones. <https://www.edureka.co/blog/what-is-git/>

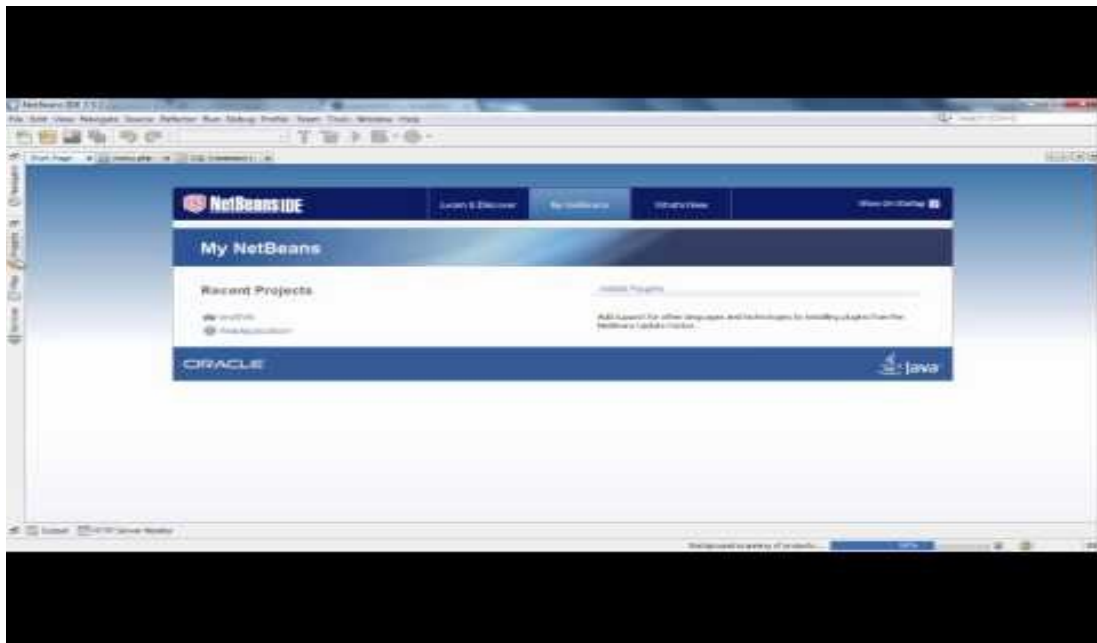
Fuente

- Tutoriales sobre el uso de repositorios:

[Tutorial - GitHub y Netbeans](#)



## subversion + visualsvn + netbeans



- **Tutorial de CVS con NetBeans**

## 19. Propuesta de configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USED y xxx-WXED.

Herramientas de desarrollo de aplicaciones web:

- **Sistema operativo:** Ubuntu Server 20.04.3 LTS/Windows 10(RAM, Disco, Particiones, Cuentas, Password, Red)
- **Servidor administración remota:** SSH
- **Servidor de transferencia de ficheros:** SFTP (SSH) Filezilla 3.55.1
- **Repositorio:** GIT Lab, **GIT Hub**
- **Servidor Web:** Apache HTTP 2.4.49 (mod\_php, mod\_ssl), XAMPP 7.4.23, Apache Tomcat 10.0.11 (JRE, JDK) (Actualizados) (para JSP, Servlets,EJB)
- **SGBD:** MariaDB 10.6.4/ MySQL 8.0, PhpMyAdmin 5.1.1, Workbench 8.0
- **Navegador:** Mozilla 92.0, Chrome 94.0.4
- **IDE:** NetBeans 12 LTS / NotePad++ 8.1/ Eclipse 2021-03 / Microsoft Visual Estudio 1.60
- **Ofimática, multimedia, generador HTML:** Libre office, Audacity, Gimp, Openshot, Dreamweaver, Microsoft FrontPage, ...
- **Framewoks PHP:** Symfony / CakePHP
- **Cliente ssh:** NetBeans / Notepad++ / FileZilla client / Putty Configurando la Máquina servidor y la Máquina cliente:

	xxx-USED	xxx-W10ED
<b>Sistema Operativo</b>	<b>Ubuntu Server 20.04.3 LTS</b>	<b>Windows 10</b>
<b>Servidor administración remota</b>	<b>SSH</b>	<b>SSH</b>
<b>Servidor de transferencia de ficheros (SSH)</b>	<b>SFTP (SSH)</b>	<b>SFTP (SSH)</b> <b>Filezilla 3.55.1</b>
<b>Repositorio</b>		<b>GIT Hub</b> GIT Lab
<b>Servidor Web</b>	<b>Apache HTTP 2.4.49</b> Con módulos _php, ssl...	<b>XAMPP 7.4.23</b>
<b>SGBD</b>	<b>MySQL 8.0</b> MariaDB 10.6.4 PhpMyAdmin 5.1.1	MySQL Workbench
<b>Navegador</b>		<b>Mozilla 92.0, Chrome 94.0.4</b>
<b>IDE</b>		<b>NetBeans 12 LTS</b> NotePad++ 8.1 Eclipse 2021-03 Microsoft Visual Studio
<b>SSH</b>	Filezilla Server 1.0.1	<b>FileZilla client 3.55.1</b> <b>Putty 0.76</b> <b>NetBeans / Notepad++</b>
<b>Ofimática, multimedia, generador HTML</b>		<b>Libre office 365</b> Audacity, <b>Gimp</b> , Openshot, Dreamweaver, Microsoft FrontPage
<b>Framewoks PHP:</b>	Laravel, Symfony , CakePHP	

Configurando la Máquina servidor y la Máquina cliente (aplicaciones java):

	<b>xxx-USED</b>	<b>xxx-W10ED</b>
<b>Sistema Operativo</b>	Ubuntu Server 20.04.3 LTS	Windows 10
<b>Servidor administración remota</b>	SSH	SSH
<b>Servidor de transferencia de ficheros (SSH)</b>	SFTP (SSH)	SFTP (SSH) Filezilla 3.55.1
<b>Repositorio</b>		GIT Hub GIT Lab
<b>Servidor Web</b>	Payara, WildFly, JBoss Con módulos JVM, JDK, Maven...	Apache Tomcat (con JVM, JDK, Maven...)
<b>SGBD</b>	MySQL 8.0 MariaDB 10.6.4	MySQL Workbench
<b>Navegador</b>		Mozilla 92.0, Chrome 94.0.4
<b>IDE</b>		NetBeans 12 LTS NotePad++ 8.1 Eclipse 2021-03 Microsoft Visual Studio
<b>SSH</b>	Filezilla Server 1.0.1	FileZilla client 3.55.1 Putty 0.76 NetBeans / Notepad++
<b>Ofimática, multimedia, generador HTML</b>		Libre office 365 Audacity, Gimp, Openshot, Dreamweaver, Microsoft FrontPage
<b>Framework Java:</b>	Spring, JSF, Struts	

## 20. Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USEE.

Herramientas de explotación de aplicaciones web:

- **Sistema operativo:** Ubuntu Server 20.04.3/Windows 10(RAM, Disco, Particiones, Cuentas, Password, Red)
- **Servidor administración remota:** SSH
- **Servidor de transferencia de ficheros:** SFTP (SSH)
- **Repositorio:** GIT Lab, GIT Hub
- **Servidor Web:** Apache HTTP 2.4.49 (mod\_php, mod\_ssl), XAMPP 7.4.23, Apache Tomcat 10.0.11 (JRE, JDK) (actualizados) (para JSP, Servlets,EJB)
- **SGBD:** MariaDB 10.6.4/ MySQL8.0, PhpMyAdmin 5.1.1, Workbench 8.0
- **Cliente ssh:** NetBeans 12 LTS/ Notepad++ 8.1 / FileZilla client 3.55.1 / Putty 0.76
- **Navegador:** Mozilla 92.0, Chrome 94.0.4

Configurando la Máquina servidor y la Máquina cliente:

	xxx-USEE	xxx-W10EE
<b>Sistema Operativo</b>	Ubuntu Server 20.04.3 LTS	Windows 10
<b>Servidor administración remota</b>	SSH	SSH
<b>Servidor de transferencia de ficheros</b>	SFTP (SSH)	SFTP (SSH)
<b>Repositorio (opcional)</b>		GIT Hub GIT Lab
<b>Servidor Web</b>	Apache HTTP 2.4.49 Con modulos_php, ssl...	
<b>SGBD</b>	MySQL 8.0 MariaDB 10.6.4 PhpMyAdmin 5.1.1	MySQL Workbench 8.0
<b>Cliente ssh</b>	Filezilla Server 1.0.1	FileZilla client 3.55.1 Putty 0.76 NetBeans / Notepad++
<b>Navegador</b>		Mozilla 92.0 Chrome 94.0.4



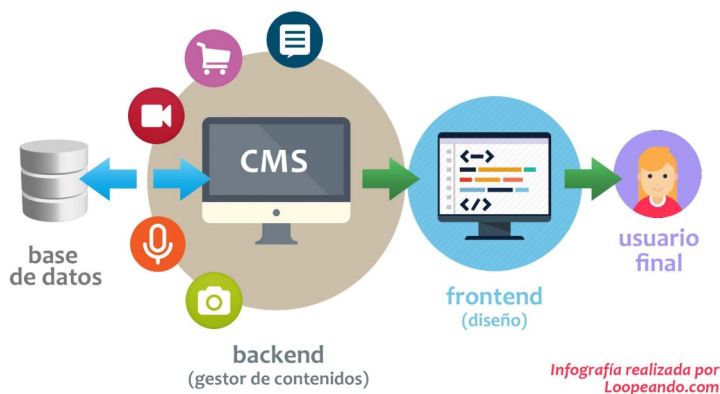
## 21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web:

**Muy poco**

### **CMS – Sistema de gestión de contenidos:**

Programa utilizado para poner en marcha una página web, ya sea un blog o una tienda, sin necesidad de programar. Permite actualizaciones constantes, creación de varios usuarios que pueden añadir, eliminar o modificar el contenido. Algunos ejemplos son: Wordpress, Joomla, WooCommerce o Prestashop.

### *¿Cómo funciona un CMS?*



### **ERP – Sistema de planificación de los recursos empresariales:**

Conjunto de aplicaciones que centraliza la información y datos procedentes de las distintas áreas de una empresa. Permite automatizar algunas tareas como informes, seguimientos, comunicaciones...



Mas información:

-CMS:

<https://rockcontent.com/es/blog/cms/>

<https://www.departamentodeinternet.com/que-es-un-cms-y-que-ventajas-tiene/>

-ERP:

<https://www.clavei.es/blog/erp-que-es/>

22. Elegir y realizar un estudio y una presentación para la exposición del trabajo sobre una de las siguientes arquitecturas de desarrollo de Aplicaciones Web:

- MEAN (con MongoDB y con MySQL) : Óscar Llamas Parra
- Java EE vs Spring --- Isabel Mtnez.
- Microsoft .NET ---Outmane bouhou
- Angular 7 ----- Rodrigo Geras
- Symfony ----Aroa Granero
- Laravel: Alberto Fernández Ramírez
- CakePHP : Johanna Herrero Pozuelo
- CodeIgniter: David del Prado Losada
- Eclipse: Sonia Antón Llanes