

# **DESARROLLO DE APLICACIONES WEB EN ENTORNO SERVIDOR**

## **TEMA 1**

## EJERCICIOS TEMA 1

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.....	3
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.....	4
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.....	4
4. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.....	5
5. Modelo de división funcional front-end / back-end para aplicaciones web.....	6
6. Página web estática – página web dinámica – aplicación web – mashup.....	6
7. Componentes de una aplicación web.....	7
8. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.....	8
9. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).....	10
10. Características y posibilidades de desarrollo de una plataforma XAMPP.....	11
11. En que 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.....	12
12. IDE más utilizados (características y grado de implantación actual).....	12
13. Servidores HTTP / HTTPS más utilizados (características y grado de implantación actual).....	13
14. Apache HTTP vs Apache Tomcat.....	14
15. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual)..	14
16. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, .....	16
17. Repositorios de software – sistemas de control de versiones: GIT , CVS, Subversion.....	16
18. 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-W7ED.....	17
19. Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USEE.....	17
20. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web: CMS y ERP.....	17
21. Elegir y realizar un estudio y una presentación para la exposición del trabajo sobre una de las siguientes arquitecturas de desarrollo de Aplicaciones Web:.....	17

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

- **IP:** Internet Protocol (Protocolo de Internet).

Estándar que se utiliza enviar y recibir información a través de una red que reúne paquetes conmutados (compuesto por datos y por la ruta a seguir hasta llegar al destino). El IP no cuenta con la posibilidad de confirmar si un paquete de datos llegó a su destino o no.

Un equipo se identifica en una red por la IP.

Un ordenador se identifica en la capa de transporte por el puerto.

**Socket:** Es un método para la comunicación entre un programa del cliente y un programa del servidor en una red.

- **TCP:** Transmission Control Protocol (Protocolo de Control de Transmisión).

Protocolo que se encarga de crear "conexiones" entre sí para que se cree un flujo de datos. Este proceso garantiza que los datos sean entregados en destino sin errores y en el mismo orden en el que salieron. También se utiliza para distinguir diferentes aplicaciones en un mismo dispositivo.

- **HTTP:** Hypertext Transfer Protocol.

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.

- **HTTPS:** Hypertext Transfer Protocol Secure (protocolo de transferencia de hiper texto seguro).

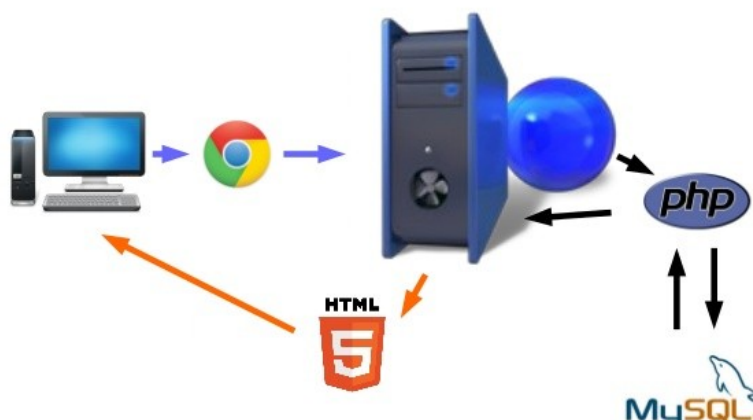
HTTPS es igual que HTTP pero añadiéndole "Seguro". Estos dos protocolos se usan para lo mismo, la transferencia de datos.

La diferencia básica entre ambos es la forma en la que viajan los datos. Si los datos son transferidos mediante HTTP, estos viajan en claro y son accesibles para cualquiera que intercepte la comunicación. En cambio, el protocolo HTTPS

usa una conexión segura mediante un cifrado SSL y por tanto los datos viajan de un modo seguro de un lugar a otro.

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

- La relación es que en una aplicación web utiliza el protocolo http o https, ambos son protocolos y son cliente-servidor, siempre funcionan de la misma manera, asimétrica, en un ordenador se carga trabajo y en otro ordenador se hace el trabajo.
- Todas las aplicaciones web utilizan 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.

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

- **HEAD:** Pide una respuesta idéntica a la de una petición *GET*, pero sin el cuerpo de la respuesta.
- **GET:** Solicita una representación de un recurso específico. Las peticiones que usan el método *GET* sólo deben recuperar datos.

- **POST:** Somete los datos a que sean procesados para un recurso en específico.  
Ej: Al darle al botón de enviar en un formulario.
- **PUT:** Sube, carga o realiza un upload de un recurso en específico.  
Ej: Crear o subir ficheros completos al servidor.
- **DELETE:** Elimina un recurso específico.  
Ej: Para borrar ficheros o recursos.
- **TRACE:** Solicita al servidor que envía de vuelta un mensaje.
- **OPTIONS:** Devuelve los métodos HTTP que el servidor soporta para una URL específica.
- **CONNECT:** Se utiliza para saber si se tiene acceso a un host.
- **PATCH:** Se utiliza para aplicar modificaciones parciales a un recurso.

#### 4. Modelo de desarrollo de aplicaciones multicapa - comunicación entre capas - componentes - funcionalidad de cada capa.

- La programación por capas es un modelo de desarrollo software en el que el objetivo primordial es la separación de las partes que componen un sistema software o también una arquitectura cliente-servidor: lógica de negocios, capa de presentación y capa de datos.
- De esta forma es sencillo crear diferentes interfaces sobre un mismo sistema sin necesidad de realizar ningún cambio en la capa de datos o lógica.
- Una de las ventajas que tiene este modelo de desarrollo es que se puede llevar a cabo en varios niveles, y en el caso de que haya que realizar algún cambio, solo afectará al nivel correspondiente sin tener que revisar el código fuente de otros módulos.

#### CAPAS Y NIVELES:

- **Capa de presentación:** lo que ve el usuario. Presenta el sistema al usuario, le comunica la información y realiza un filtrado previo para comprobar que no hay errores de formato. También es conocida como interfaz gráfica, y debe de ser entendible y fácil de utilizar para el usuario.

Esta capa se comunica únicamente con la capa de negocio.

- **Capa de negocio:** es donde se almacenan los programas que se ejecutan. Reciben las peticiones del usuario y envía respuestas tras el proceso. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que se deben cumplir.

Esta capa se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados y con la capa de datos para solicitar al gestor de la base de datos almacenar o recuperar datos de él.

- **Capa de datos:** es donde se almacenan los datos y se encarga además 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.

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

- **Front-end:** Es la parte del software que interactúa con los usuarios. El conjunto de páginas que ven la gran mayoría de usuarios que las usan (usuarios externos).
- **Back-end:** es otro conjunto de páginas dinámicas que utilizan las personas que producen el contenido y las que administran la aplicación web (usuarios internos) para crear contenido, organizarlo, decidir la apariencia externa, etc.

## 6. Página web estática - página web dinámica - aplicación web - mashup.

- **Página web estática:** Una página web estática es aquella que se encuentra almacenada en su forma definitiva, tal y como se creó, y su contenido no varía. Para cualquier persona que visite una página web estática, a todo el mundo le va a salir lo mismo.
- **Página web dinámica:** A diferencia que las páginas web estáticas, estas otras páginas sí que varían. Pueden cambiar en función del navegador que estás utilizando, en usuario con el que te has identificado, o mismamente por acciones que se han efectuado con anterioridad.

- **Aplicación web:** Es una aplicación o herramienta informática que los usuarios pueden utilizar para acceder desde cualquier navegador, bien sea a través de internet, o bien a través de una red local.

#### **Ventajas:**

- No necesita ningún tipo de instalación.
  - Multiplataforma y multidispositivo.
  - La potencia no está en el dispositivo que accede.
  - Puede estar en la nube.
  - Visualmente intuitiva.
- **Mashup:** Es una aplicación web híbrida que integra herramientas o datos de una o más fuentes de la Red en un nuevo servicio. Los contenidos de los mashups se obtienen de sus fuentes a través de un API, que es una colección de funciones o servicios sobre los que se puede interactuar desde otra aplicación diferente de aquella para la que se diseñaron.

#### **Tipos:**

- **Mashup de consumidores:** es el tipo más utilizado y está orientado al público en general; por ejemplo, los mashups de Google maps.
- **Mashup de empresas:** orientados al ámbito empresarial. Combinan la integración de datos externos e internos.
- **Mashup de datos:** combina, en el mismo mashup, datos de varias fuentes, sin que haya datos propios.

## **7. Componentes de una aplicación web.**

Todas las aplicaciones de base de datos basadas en la web tienen tres componentes primarios: un navegador web (o cliente), un servidor de aplicaciones web y un servidor de bases de datos.

Los componentes de una aplicación web del lado del servidor son:

- **Servidor web:** Para recibir las peticiones de los clientes web y enviarles la página que solicitan. El servidor web debe conocer el procedimiento a seguir



para generar la página web: qué módulo se encargará de la ejecución del código y cómo se debe comunicar con él.

- **Módulo encargado de ejecutar el código:** Es un programa que se encarga de generar la página web. Este módulo debe integrarse de alguna forma con el servidor web, y dependerá del lenguaje y tecnología que utilicemos para programar la aplicación web.
- **Sistema gestor de base de datos (si fuera necesario):** Normalmente también será un servidor. Este módulo no es estrictamente necesario, pero en la práctica se utiliza en todas las aplicaciones web que utilizan grandes cantidades de datos para almacenarlos.
- **Lenguaje de programación:** Utilizado para desarrollar las aplicaciones. Por ejemplo, PHP.

## 8. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.

Programas ejecutados en el lado del **cliente**:

- ◆ **HTML:** indica al navegador donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página.
- ◆ **JAVASCRIPT:** lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento.

En los últimos tiempos JavaScript se ha ido saliendo del lado del cliente y hay programadores que ya lo están programando en el lado del servidor.

- ◆ **APPLETS DE JAVA:** Es otra manera de incluir código a ejecutar en los clientes que visualizan una página web. Se trata de pequeños programas hechos en Java, que se transfieren con las páginas web y que el navegador ejecuta en el espacio de la página.
- ◆ **VISUAL BASIC SCRIPT:** Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su



utilización está desaconsejada a favor de Javascript. Está basado en Visual Basic, un popular lenguaje para crear aplicaciones Windows.

- ◆ **FLASH:** Flash es una tecnología, y un programa, para crear efectos especiales en páginas web. Con Flash también conseguimos hacer páginas dinámicas del lado del cliente. Flash en realidad no es un lenguaje; Sin embargo, si tuviéramos que catalogarlo en algún sitio quedaría dentro del ámbito de las páginas dinámicas de cliente.
- ◆ **CSS:** Cascading Style Sheets (Hojas de estilo en Cascada). Tecnología que nos permite crear páginas web de una manera más exacta. Gracias a las CSS somos mucho más dueños de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, etc.

Programas ejecutados en el lado del **servidor**:

- ◆ **CGI:** Es el sistema más antiguo que existe para la programación de las páginas dinámicas de servidor. Actualmente se encuentra un poco desfasado por diversas razones entre las que destaca la dificultad con la que se desarrollan los programas y la pesada carga que supone para el servidor que los ejecuta. Los CGI se escriben habitualmente en el lenguaje Perl, sin embargo, otros lenguajes como C, C++ o Visual Basic pueden ser también empleados para construirlos.
- ◆ **PERL:** es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como Javascript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito.
- ◆ **ASP:** (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft).
- ◆ **PHP:** Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

- ◆ **JSP:** Java Server Pages (Página de Servidor Java). Es una tecnología orientada a crear páginas web con programación en Java. Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.
- ◆ **PYTHON:** es un lenguaje de programación de código abierto, orientado a objetos, muy simple y fácil de entender. Tiene una sintaxis sencilla que cuenta con una amplia biblioteca de herramientas.

Programas del lado **cliente-servidor**:

- ◆ **DHTML:** no es precisamente un lenguaje de programación. Más bien se trata de una nueva capacidad de la que disponen los navegadores modernos, por la cual se puede tener un mayor control sobre la página que antes. Nos da más control sobre la página, gracias a que los navegadores modernos incluyen una nueva estructura para visualizar en páginas web denominada capa. Las capas se pueden ocultar, mostrar, desplazar, etc.
- ◆ **XML:** es una tecnología sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes.

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

Los lenguajes más utilizados en el lado del servidor son:

- ✓ **ASP.NET:** Lenguaje comercializado por Microsoft y muy utilizado por programadores de todo el mundo para la creación y el desarrollo de páginas web dinámicas. ASP.NET fue creado para resolver las limitaciones que presentaba ASP. Este lenguaje cuenta con una serie de clases .NET, las cuales se utilizan para la creación de aplicaciones web, tanto del lado del cliente como del propio servidor. Un dato a destacar es que los sitios web creados con este lenguaje de

programación del lado del servidor, se pueden ejecutar en todos los navegadores del mercado.

**Ventajas:** Permite el control de usuario personalizado y ofrece una división entre la capa de diseño y el código.

**Desventaja:** Consume una gran cantidad de recursos.

- ✓ **PERL:** Lenguaje de programación basado en una serie de herramientas UNIX. Es muy útil en el procesamiento tanto de textos como de archivos. Está disponible en una gran selección de plataformas y sistemas operativos. Es uno de los lenguajes del lado del servidor preferidos por parte de los programadores ya que el desarrollo de aplicaciones con PERL es muy rápido. Además, ofrece una gran libertad para crear y desarrollar el programa como quieras.

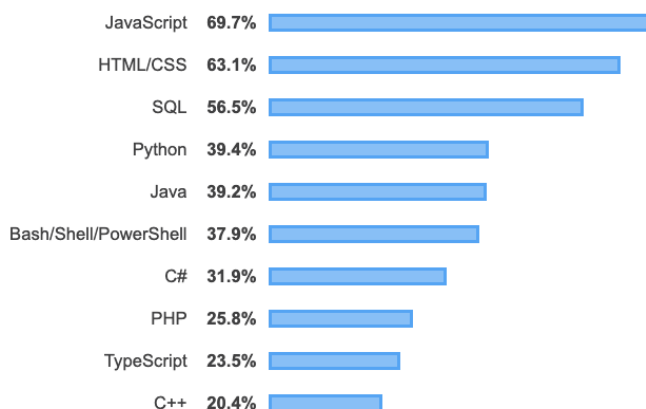
**Desventaja:** Resulta un poco lento para determinadas aplicaciones, como por ejemplo la programación a bajo nivel.

- ✓ **PHP:** Se trata de uno de los lenguajes de programación del lado del servidor más utilizados actualmente. Gratuito, independiente y rápido.

**Ventajas:** Fácil de aprender, multiplataforma. Incorpora una gran sección de funciones y ofrece una gran facilidad para establecer conexión con todo tipo de bases de datos, como Oracle o MySQL.

**Desventajas:** Todo el trabajo recae sobre el servidor, no delega ni una sola tarea en el lado del cliente.

## GRADO DE IMPLANTACIÓN:



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

- XAMPP es un instalador cuya función es la de un servidor independiente, software libre, que está compuesto principalmente por una base de datos MySQL, un servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.
- Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris).
- Permite a los usuarios desarrollar aplicaciones web gracias a todas las características que vienen dentro del instalador sin necesidad de crear un servidor web ni instalar un gestor de bases de datos.

## 11. En que 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.

- Es necesario utilizar el software JDK cuando estemos realizando una aplicación que contenga código de Java.
- Es necesario la instalación de la máquina virtual Java cuando se va a ejecutar el código precompilado.

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

- **NetBeans:** es un entorno de desarrollo. Una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

### **CARACTERÍSTICAS:**

- Permite subdividir una aplicación en partes más pequeñas.
  - Incluye servicios para el control de la interfaz con el usuario.
  - Permite el desarrollo en otros lenguajes, no solo en Java.
  - Acceso a base de datos.
- **Eclipse:** Se trata de un software compuesto por un conjunto de herramientas de programación de código abierto para desarrollar aplicaciones web.

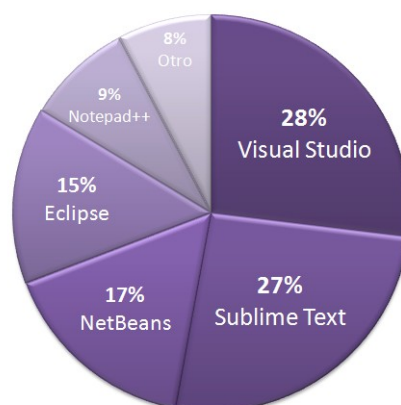
### CARACTERÍSTICAS:

- Es posible desarrollar plugins personalizados.
- Herramienta de código abierto.
- Permite el desarrollo de aplicaciones para el servidor y servicios web.
- Todas sus funcionalidades se componen de plugins descargables.
- Utilización de repositorios con un plugin.
- **Visual Studio Code:** Es un editor de código fuente desarrollado por Microsoft para varios Sistemas Operativos en el que se puede escribir y depurar código. Tiene un control integrado de Git. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privado.

### CARACTERÍSTICAS:

- Se puede utilizar como lenguaje de programación.
- Soporta una gran variedad de lenguajes (HTML, CSS, JAVASCRIPT, etc).
- Posibilidad de configurar la interfaz a nuestro gusto.
- Cuenta con una amplia gama de temas personalizables.
- Cuenta con un buen soporte técnico.

### GRADO DE IMPLANTACIÓN:



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

Los servidores web más utilizados en la industria son:

#### → Apache.

##### **CARACTERÍSTICAS:**

- Código abierto y gratuito.
- Software estable.
- Parches de seguridad regulares y actualizados con frecuencia.
- Flexible debido a la estructura basada en módulos.
- Fácil de configurar.
- Multiplataforma.
- Listo para trabajar con CMS como por ejemplo WordPress.
- Buen soporte técnico.

#### → NGINX.

##### **CARACTERÍSTICAS:**

- Consume pocos recursos.
- Mejor escalabilidad.
- Recomendado para sitios web que coren en VPS.

#### → IIS (Internet Information Services).

##### **CARACTERÍSTICAS:**

- Ideal para principiantes por su sencilla interfaz.
- Buena integración con el monitor de rendimiento para estadísticas.

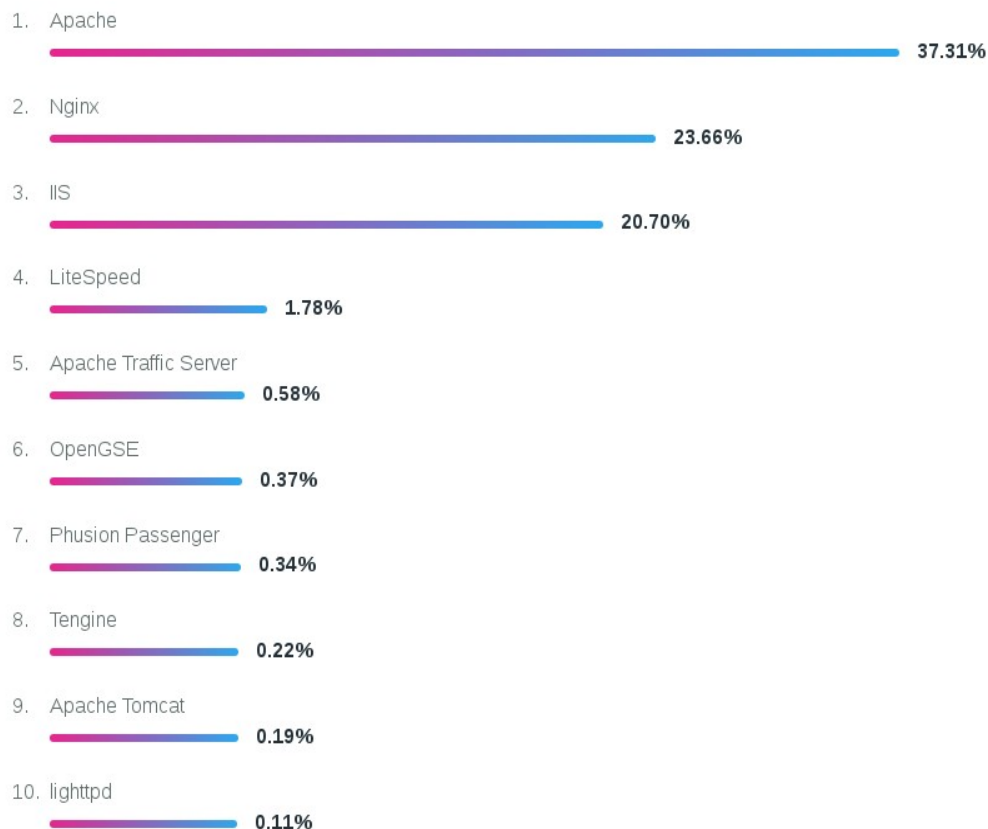
#### → Apache Tomcat.

##### **CARACTERÍSTICAS:**

- Liviano; se inicia de manera rápida y utiliza menos memoria.

- Se integra fácilmente con Apache HTTP.
- Despliegue sencillo en VPS, Cloud o Docker.
- Fácil de usar.

## GRADO DE IMPLANTACIÓN:



### 14. Apache HTTP vs Apache Tomcat.

- Apache HTTP es un servidor web desarrollado por Apache que se utiliza para almacenar páginas web estáticas y dinámicas, utilizando lenguajes de programación como PHP, Perl, Python y Ruby.
- Apache Tomcat también es un servidor web, desarrollado por Apache al igual que Apache HTTP. También soporta páginas web estáticas aunque su rendimiento es



menor. Su especialidad son las páginas web dinámicas desarrolladas con la tecnología Java, como Java Servlet o JavaServer Page (JSP).

- Ambas son software que se ejecutan en un servidor cuya finalidad es la de ayudar a desplegar un proyecto web.

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

#### → Google Chrome.

##### CARACTERÍSTICAS:

- Un cuadro para todo.
- Página Nueva pestaña.
- Acceso directo a diferentes aplicaciones.
- Pestañas dinámicas.
- Control de fallos.
- Modo incógnito.
- Navegación segura.
- Marcadores instantáneos.
- Importar configuración.
- Descargas más simples.

#### → Mozilla Firefox.

##### CARACTERÍSTICAS:

- Multiplataforma.
- Navegación con pestañas.
- Restauración de sesión.
- Corrector ortográfico.
- Sugerencias de búsquedas.

- Canales RSS.
- Bloqueador de ventanas emergentes.
- Protección antiphishing.
- Actualizaciones automáticas.
- Protección contra espías.
- Temas.
- Plugins.

→ **Internet Explorer y Microsoft Edge.**

**CARACTERÍSTICAS:**

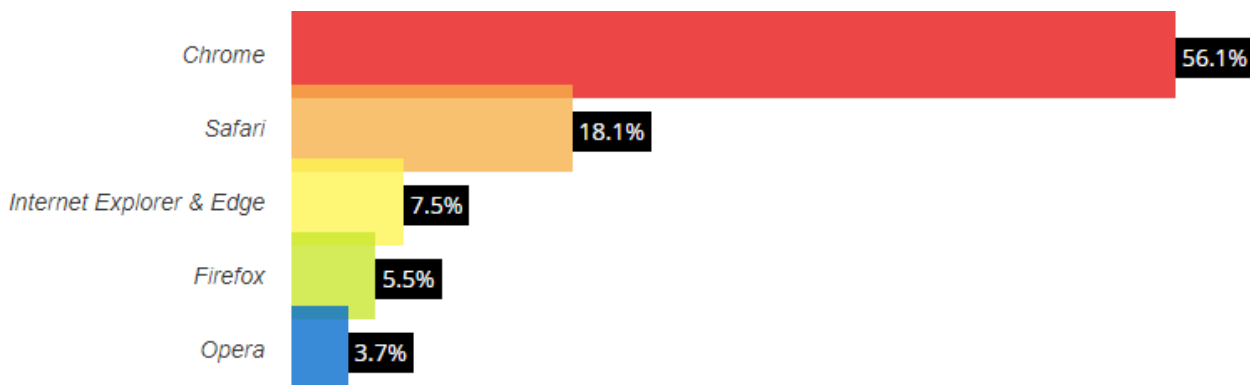
- Cortana.
- Escritura en las páginas web.
- Nueva pestaña.
- Extensiones.
- Otras funciones de seguridad.

→ **Safari.**

**CARACTERÍSTICAS:**

- Rápido.
- Soporta extensiones.
- Rendimiento óptimo en portátiles.
- Sincronía entre Mac, iPad o iPhone.
- Elegir el motor de búsqueda predeterminado.

## GRADO DE IMPLANTACIÓN:



### 16. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, ...

- Es un programa que es capaz de sacar de una determinada estructura los comentarios de un código en la página web.

### 17. Repositorios de software – sistemas de control de versiones: GIT , CVS, Subversion.

Un repositorio de software es un almacén, un lugar en internet donde se almacena, se descarga y se sube información de diversos tipos (documentación, proyectos, prácticas, código, etc).

Existen diferentes repositorios, pero uno de los que más se están utilizando en estos momentos es el GitHub.

**GitHub:** Es un portal creado para el alojamiento de código de las aplicaciones de cualquier desarrollador. Fue comprada por Microsoft en junio de 2018. Fue creada para que los desarrolladores pudiera subir código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino que también puedas entrar en su perfil para leer sobre ella o incluso colaborar con su desarrollo.

**GitLab:** Es un servicio web de control de versiones y desarrollo de software colaborativo basado en Git. Además de gestor de repositorios, el servicio ofrece también alojamiento de wikis y un sistema de seguimiento de errores, todo ello publicado bajo una Licencia de código abierto.

**18. 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-W7ED.**

- **Sistema Operativo:** Windows 10.
- **Servidor administración remota:** SSH.
- **Servidor de transferencia de ficheros:** SFTP (SSH) Puerto 22.
- **Repositorio:** GitHub.
- **Servidor web:** Apache 2.4 HTTP
- **Gestor de Bases de Datos:** MySQL 8.0.
- **Navegador:** Mozilla Firefox 81.0.
- **IDE:** NetBeans 12.0.
- **Ofimática, multimedia, generador html:** LibreOffice 6.4.
- **Frameworks PHP:** Pendiente de actualizar.
- **Cliente SSH:** Filezilla 3.50.0.

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

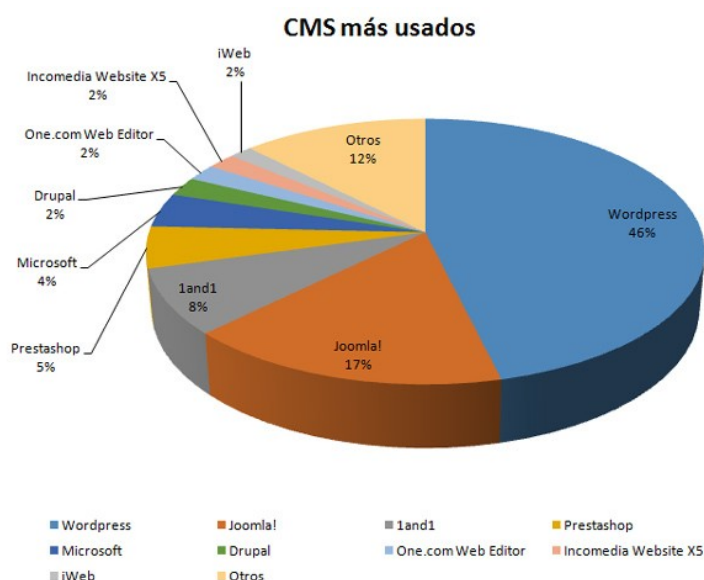
- **Sistema Operativo:** Ubuntu Server 20.04.
- **Servidor administración remota:** SSH.
- **Servidor de transferencia de ficheros:** SFTP (SSH) Puerto 22.
- **Repositorio:** GitHub.
- **Servidor Web:** Apache 2.4 HTTP
- **Gestor de Bases de Datos:** MySQL 8.0.
- **Cliente SSH:** NetBeans 12.0 / Filezilla client 3.50.0.
- **Navegador:** Mozilla Firefox 81.0.

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

- **CMS – Sistema de gestión de contenidos:** Es una herramienta software que permite crear, administrar y gestionar un sitio web.
  - Permite crear un sitio web
  - Permite la gestión y mantenimiento del sitio web.
  - Permite la administración del sitio web y del propio CMS.

Ejemplos: WordPress, Joomla, PrestaShop.

- Cualquier persona que no sepa nada de programación podría ser capaz de crear una página web utilizando un CMS.



- **ERP – Sistema de planificación de los recursos empresariales:** Se trata de un software de gestión empresarial compuesto por varios módulos. Estos programas se hacen cargo de distintas operaciones internas de una empresa, desde producción a distribución o incluso recursos humanos. Supone una gran inversión para las empresas. Ahorra tiempo y coste.

21. 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).
- Java EE vs Spring.
- Microsoft .NET.
- Angular 7.
- Symfony.
- **Laravel**.
- CakePHP.
- CodeIgniter.