

# EJERCICIOS

## TEMA 1



## INDICE

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS. ....	3
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web. ....	3
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados. ....	3
4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme) /URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS. ....	4
5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa. ....	4
6. Modelo de división funcional front-end / back-end para aplicaciones web. ....	5
7. Página web estática – página web dinámica – aplicación web – mashup. ....	5
8. Componentes de una aplicación web. ....	6
9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso. ....	7
10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual). ....	7
11. Características y posibilidades de desarrollo de una plataforma XAMPP. ....	7
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. ....	8
13. IDE más utilizados (características y grado de implantación actual). ....	8
14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual). ....	9
15. Apache HTTP vs Apache Tomcat. ....	10
16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual). ....	10
17. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, ... ....	11
18. Repositorios de software – sistemas de control de versiones: GIT, CVS, Subversion, ....	11
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. ....	12
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. ....	12
21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web: CMS – Sistema de gestión de contenidos ERP – Sistema de planificación de los recursos empresariales. ....	12
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) • Java EE vs Spring • Microsoft .NET • Angular 7 • Symfony • Laravel • CakePHP • CodeIgniter ....	13

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

- **IP:** El protocolo IP es uno de los pilares básicos de internet, permite el desarrollo y el transporte de paquetes de datos, aunque su recepción no está asegurada.
- **TCP:** El protocolo TCP es un protocolo encargado de informar del destino de los datos permitiendo la creación de conexiones seguras.
- **HTTP:** El protocolo HTTP es un protocolo de transferencia de hipertexto cliente-servidor que permite los intercambios de información entre los clientes Web y los servidores HTTP.
- **HTTPS:** El protocolo HTTPS es un protocolo de transferencia de hipertexto seguro que protege la integridad y confidencialidad de los datos de los usuarios.

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

El modelo de comunicaciones cliente-servidor es una arquitectura de software en el que las tareas se reparten entre los servidores y los clientes. Un cliente realiza una petición a otro programa y es un servidor el que le da respuesta.

En cuanto a la relación con las aplicaciones Web, el modelo de comunicaciones cliente servidor es el encargado de que funcione cualquier página web cuando un cliente solicita entrar en una.



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

- **HEAD:** Pide una respuesta.
- **GET:** Encargado de pedir una representación del recurso específico.
- **POST:** Encargado de someter los datos a que sean procesados para el recurso identificado.

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

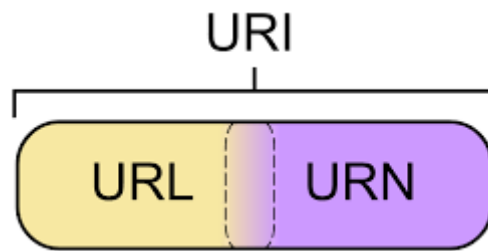
**URI:** Es una cadena de caracteres que se utilizan para identificar un recurso o un nombre en internet.

Un URI consta de un máximo de cinco partes, de las cuales solo dos son obligatorias:

- scheme (esquema): proporciona información sobre el protocolo utilizado.
- authority (autoridad): identifica el dominio.

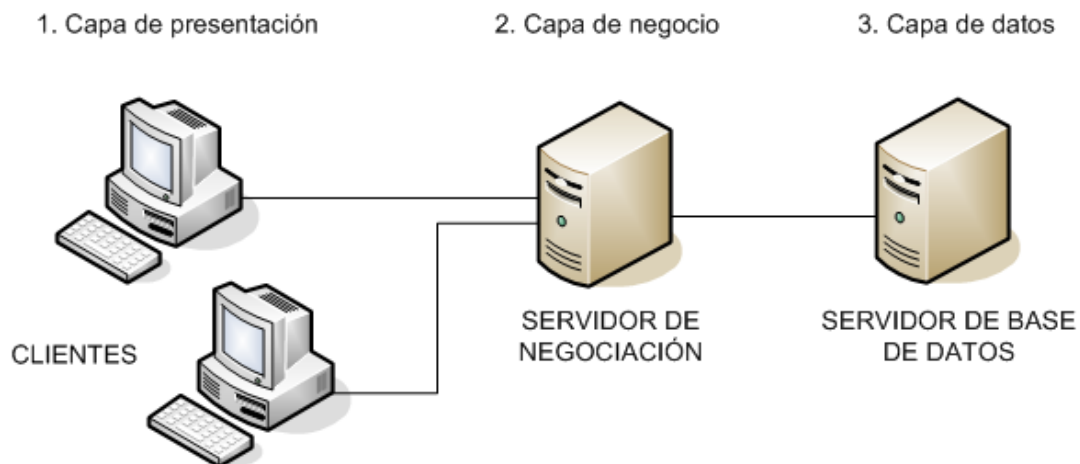
El **URL** se utiliza para indicar dónde se encuentra un recurso.

El **URN** es independiente de la ubicación y designa un recurso de forma permanente.



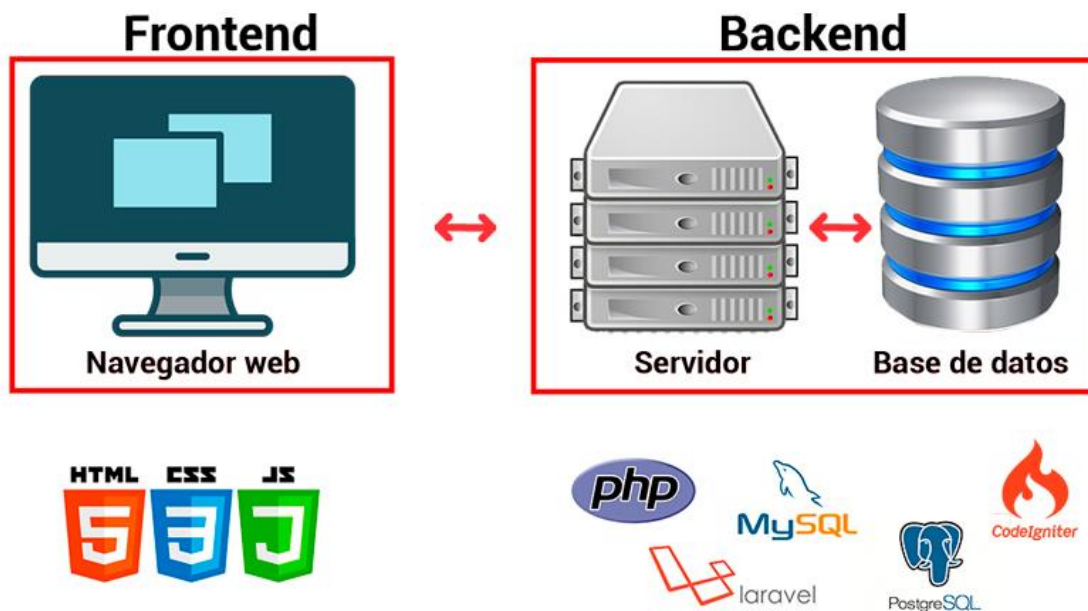
#### 5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.

- **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. Esta capa se comunica únicamente con la capa de negocio.
- **Capa de negocio:** es 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, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.
- **Capa de datos:** es 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.



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

- **front-end:** Es la parte de la aplicación web que interactúa con el usuario. Consta de la visualización del sitio web, la interfaz, el diseño web, etc.... El desarrollador de front-end se encarga de la experiencia del usuario, es decir, que para el usuario sea fácil de usar, atractiva y funcional.
- **back-end:** El back-end es la parte de la aplicación web del lado del servidor, la parte del back-end consta de un servidor, una aplicación y una base de datos. El desarrollador del back-end debe tener amplios conocimientos de los frameworks y los tipos de bases de datos, como también tener habilidad con algunos lenguajes, entenderlos y saber trabajar con ellos.



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

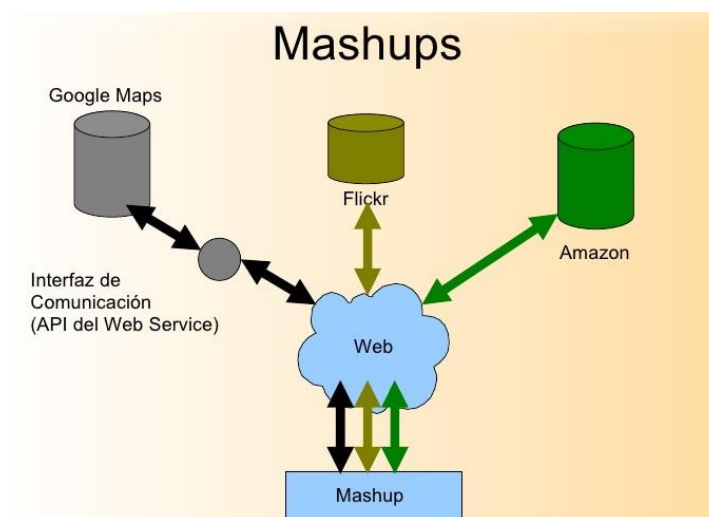
- **Página web estática:** Pagina encargada de mostrar una información permanentemente. El usuario se limita a obtener dicha información sin poder interactuar con la página web.
- **Página web dinámica:** Pagina web que permite crear aplicaciones dentro de la propia web, dando una mayor interactividad con el cliente. La ventaja de una web dinámica es su personalización de la información mostrada.



- **Aplicación web:** Aplicación a la que un usuario puede utilizar accediendo a un servidor web mediante un navegador, dicha página almacenara la información del usuario en su servidor.



- **Mashup:** Es una aplicación web que combina servicios de varias páginas web en una sola para ofrecer un nuevo servicio.



## 8. Componentes de una aplicación web.

Los componentes de una aplicación web suelen ser:

- Un navegador web para el cliente.
- Una base de datos en el servidor.
- Un servidor web.
- Modulo con el código.

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

En el lado de cliente:

- Navegadores web como: Google Chrome, Firefox, Edge...
- Utiliza lenguajes de programación como pueden ser:
  - JavaScript
  - VBScript
  - HTML
  - CSS
  - AJAX
  - jQuery

En el lado de servidor:

- Programas como: MySQL, MariaDB...
- Lenguajes de programación como:
  - PHP
  - ASP.NET (C# O Visual Basic)
  - C++
  - Java y JSP
  - Python

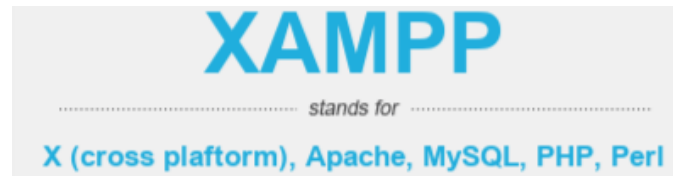
## 10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).

- **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.
- **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.
- **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.

## 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.

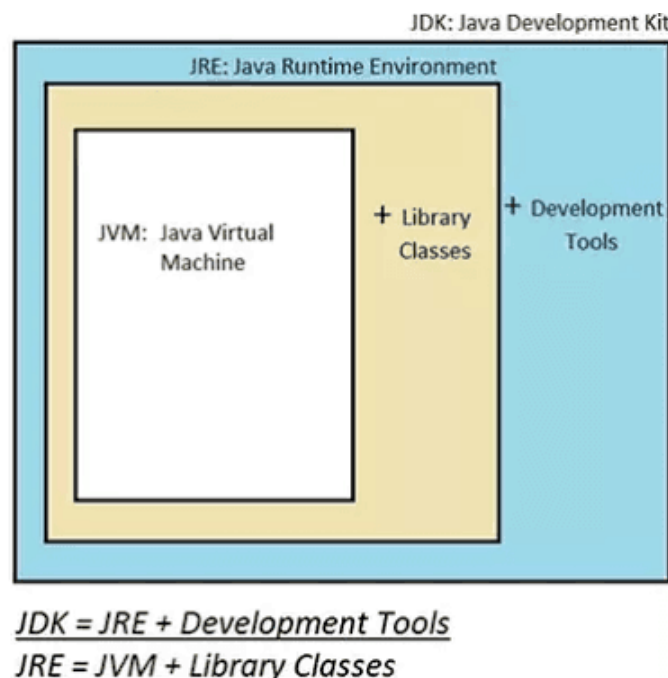




12. En qué casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación.

La **máquina virtual de Java** es el entorno en que se ejecutan los programas Java; es por eso por lo que un programa en Java puede ser ejecutado en cualquier sistema operativo, pero no si no tiene la JVM.

El **JDK** —Java Development Kit—, sin embargo, solamente es una herramienta para desarrollar programas. No es necesario que el usuario del programa lo tenga para poder ejecutarlo.



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

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

- 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.
- 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.

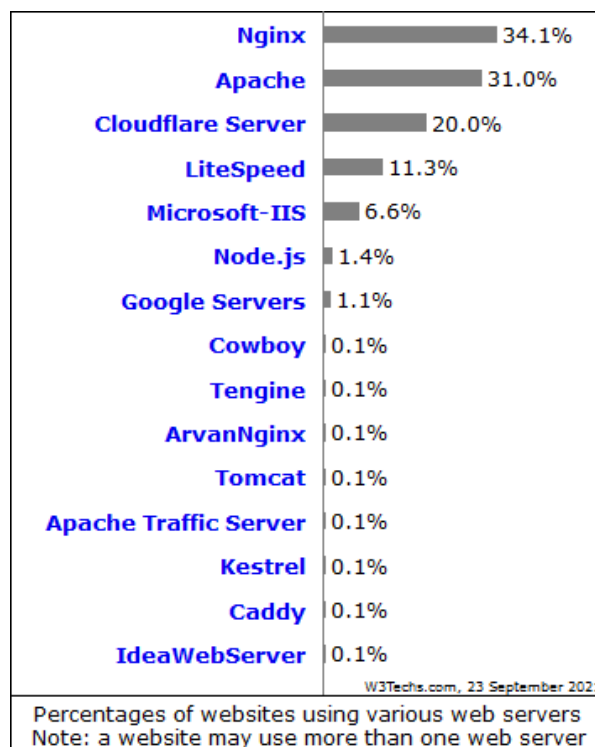


## 14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).

Los servidores HTTP más utilizados son Apache y Nginx.

Ambos son gratuitos y reciben actualizaciones y parches de seguridad constantemente.

Servidores	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



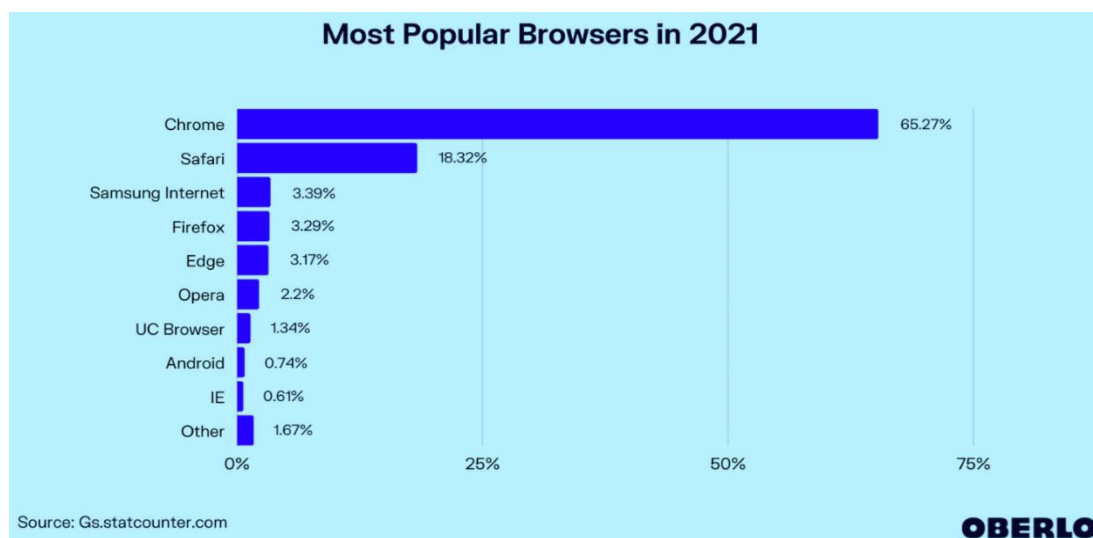
## 15. Apache HTTP vs Apache Tomcat.

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

- Apache HTTP proporciona cualquier contenido como texto, vídeo o imágenes en páginas estáticas HTML.
- Tomcat proporciona páginas web dinámicas basadas en aplicaciones Java, que cambian dependiendo del momento y del usuario.
- 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.

## 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.



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

- **PhpDocumentor** es una de las herramientas más utilizadas que permiten generar documentación de forma automática a partir del código fuente para PHP. JavaDoc, por ejemplo, es la herramienta estándar en Java para generar la documentación.
- En phpDocumentor la documentación se distribuye en bloques DocBlock. Estos bloques siempre se colocan justo antes del elemento al que documentan.
- Los elementos que pueden ser documentados son: define, function, class, class var, include/require/include\_once/require\_once y globales variables.
- Puede incluir documentación global a nivel de fichero y clase mediante la marca @package
- **ApiGen**: herramienta utilizada para la documentación de PHP mediante el estándar de PHPdoc.



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

Los programas de control de versiones son conjunto de aplicaciones, su idea es gestionar 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.
- **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.
- **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.
- **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.
- **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.

## 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, ...

- **Frameworks PHP:** Symfony / CakePHP
- **Cliente ssh:** NetBeans / Notepad++ / FileZilla client / Putty

## 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

## 21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web: CMS – Sistema de gestión de contenidos ERP – Sistema de planificación de los recursos empresariales.

- **CMS – Sistema de gestión de contenidos:**  
Es un programa informático que permite crear un entorno de trabajo para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás usuarios.
- **ERP – Sistema de planificación de los recursos empresariales:**

Es un software que permite a las empresas controlar todos los flujos de información que se generan en cada ámbito de la organización.

## 22. Elegir y realizar un estudio y una presentación para la exposición del trabajo sobre Laravel

**ULTIMA EDICION: 28/09/2021 21:58**