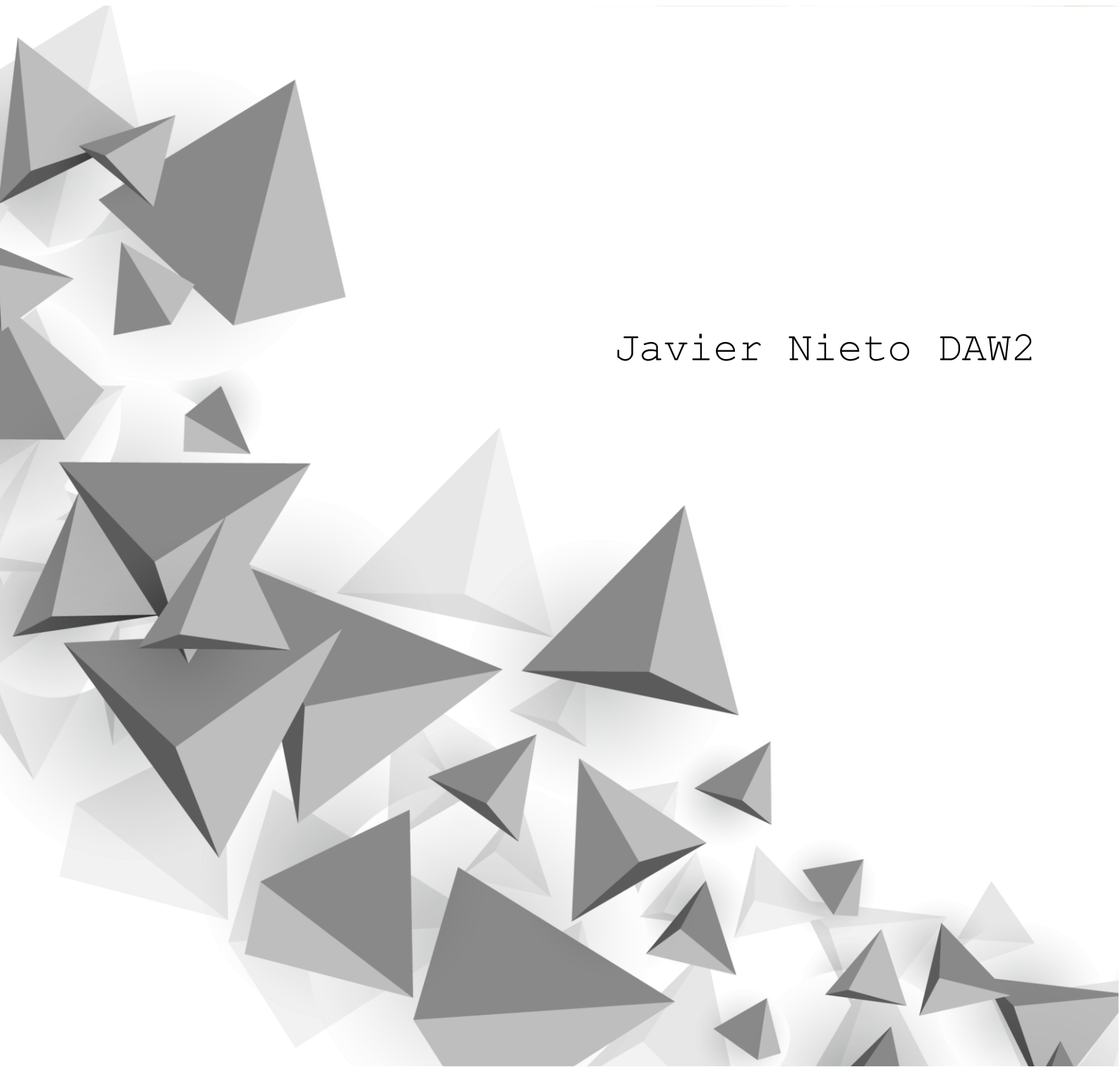


TEMA 1

EJERCICIOS

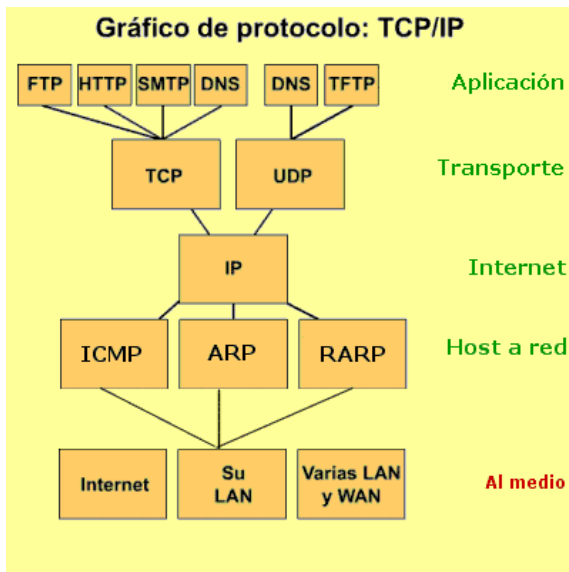
Javier Nieto DAW2



Índice

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.	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.	5
6. Página web estática – página web dinámica – aplicación web – mashup	6
7. Componentes de una aplicación web.	6
8. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor – lenguajes de programación utilizados en cada caso.	6
9. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).	7
10. Características y posibilidades de desarrollo de una plataforma XAMPP.	8
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.	8
12. IDE más utilizados (características y grado de implantación actual).	9
13. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).	10
14. Apache HTTP vs Apache Tomcat	10
15. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).	11
16. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen,	12
17. Repositorios de software – sistemas de control de versiones: GIT, CVS, Subversion,	12
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-W10ED.	13
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.	13
20. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web:	14

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

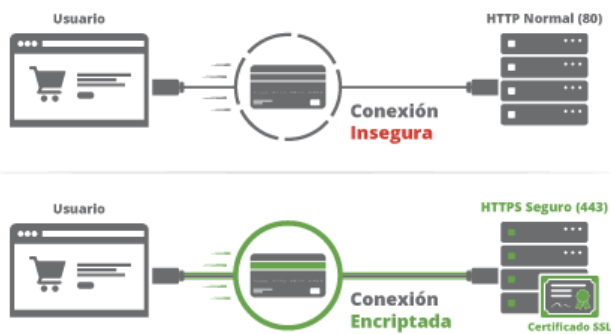


TCP/IP es un conjunto de protocolos que permiten la comunicación entre los ordenadores pertenecientes a una red, hace referencia a dos protocolos, IP y TCP.

IP (*Internet Protocol*) es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

TCP (Transmission Control Protocol) Protocolo de Control de Transmisión.: Este protocolo 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.

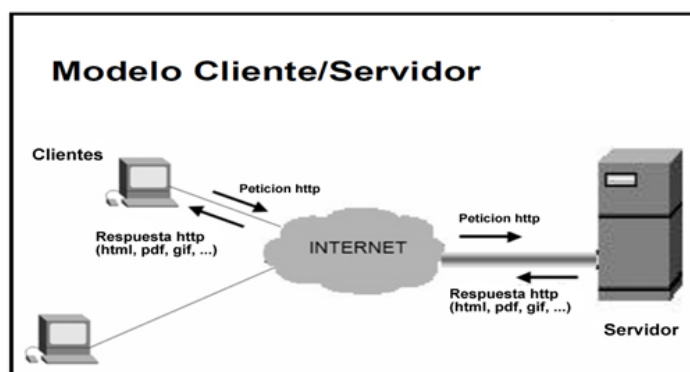
HTTP vs HTTPS



HTTP (Hypertext Transfer Protocol) es el protocolo de comunicación que permite las transferencias de datos de hipertexto.

HTTPS (Hypertext Transfer Protocol Secure) es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP. El sistema HTTPS utiliza un cifrado basado en la seguridad de textos SSL/TLS para crear un canal más apropiado para el tráfico de información sensible que el protocolo HTTP.

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



La arquitectura **cliente-servidor** es un modelo de aplicación distribuida en el que las tareas se reparten entre los servidores y los clientes. Un cliente realiza peticiones al servidor, quien le da respuesta.

Tanto en la petición como en la respuesta se utiliza el protocolo HTTP/HTTPS.

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

El método **GET** solicita una representación de un recurso específico. GET devuelve la representación de un recurso en un formato concreto: HTML, XML, JSON o imágenes, JavaScript, CSS, etc. En caso de respuesta negativa devuelve 404 (not found) o 400 (bad request).

El método **HEAD** pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

El método **POST** se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor. Se da cuando el cliente envía información al servidor.

El modo **PUT** se utiliza para actualizar o crear ficheros completos al servidor.

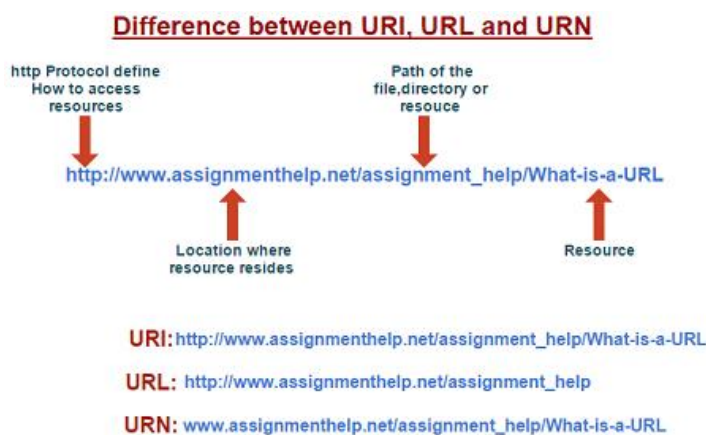
El método **DELETE** borra un recurso en específico.

El método **CONNECT** establece un túnel hacia el servidor identificado por el recurso.

El método **OPTIONS** es utilizado para describir las opciones de comunicación para el recurso de destino.

El método **TRACE** realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.

El método **PATCH** es utilizado para aplicar modificaciones parciales a un recurso.



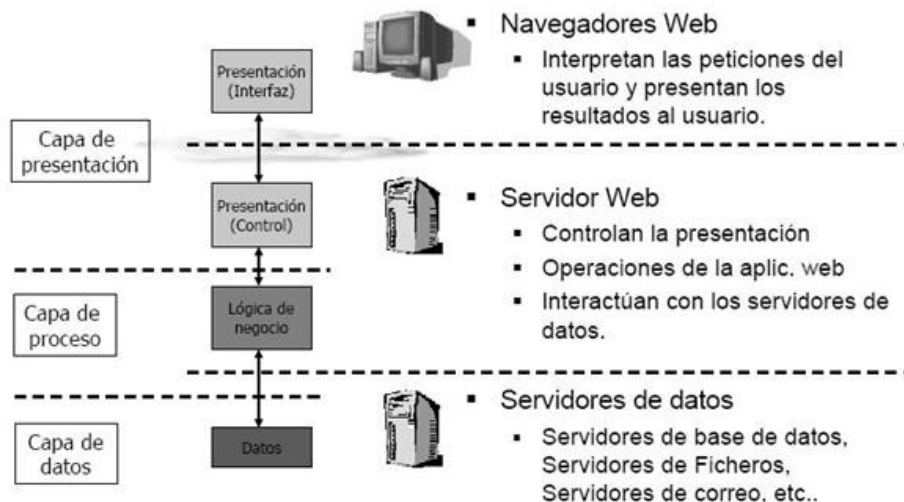
Un **URI** (Uniform Resource Identifier) es una cadena de caracteres que identifica los recursos de una red de forma unívoca.

Un **URL** (Uniform Resource Locator) es la dirección específica que se asigna a cada uno de los recursos disponibles en la red para que puedan ser localizados o identificados.

Un **URN** (Uniform Resource Name) indica el nombre de un recurso que debe ser único en todo internet.

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: capa de presentación, lógica de negocios y capa de datos. De esta forma es sencillo y mantenible crear diferentes interfaces sobre un mismo sistema sin requerirse cambio alguno en la capa de datos o lógica.



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

MODELO FRONT-END

Parte de la aplicación web que programamos y utiliza un usuario normal.

Se trabaja más lo estético.

MODELO BACK-END

Parte de la aplicación web que programamos para usuarios especiales (administradores).
Se trabajan más los contenidos.



6. Página web estática – página web dinámica – aplicación web – mashup .

Página Web Estática - Una página web estática es aquella en la cual el contenido de ésta no varía.

Página Web Dinámica - Una página web dinámica es aquella en la cual el contenido es variable en el tiempo al introducir la misma URL.

Aplicación Web – Es un conjunto de herramientas que los usuarios pueden utilizar accediendo a un servidor web mediante el navegador.

Mashup – Es aquella página web la cual utiliza web service, es decir se comunica con otras aplicaciones web y reutiliza su contenido o funcionalidad , tiene control de acceso y utiliza varias tecnologías y las mezcla.

7. Componentes de una aplicación web.

Servidor web - No puede existir una página web sin servidor web.

Módulo encargado de ejecutar el código - Sin ello no puede haber código embebido

Sistema gestor de base de datos (si es necesario) - Para la persistencia de datos

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

Programas de lado del cliente – Navegadores (Firefox, Chrome)

Programas del lado de servidor – Servidor web (Apache HTTP, Apache Tomcat), SGBD (MySQL)

Lenguaje del lado del cliente - JavaScript, HTML

Lenguaje del lado del servidor - PHP, Perl, Python

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



PHP (Hypertext Pre-Processor)

- Es un lenguaje de programación de **propósito general** que se ejecuta **en el lado del servidor**.
- Es un **lenguaje interpretado**.
- Tiene **múltiples formas de utilizarse**, ya que puede utilizarse con scripts, de **forma estructurada o programación en objetos**



JAVA

- Es **simple**, ofrece la funcionalidad de un lenguaje potente, derivado de C y C++, pero sin las características menos usadas.
- **Orientado a objetos**, permite diseñar el software de forma que los distintos tipos de datos que se usen estén unidos a sus operaciones.
- Es **distribuido**, proporciona una gran biblioteca estándar y herramientas para que los programas puedan ser distribuidos.
- **Independiente a la plataforma**, los programas escritos en el lenguaje Java pueden ejecutarse en cualquier tipo de hardware, lo que lo hace portable.
- Es **multihilo**, logra llevar a cabo varias tareas simultáneamente dentro del mismo programa. Esto permite mejorar el rendimiento y la velocidad de ejecución.



PYTHON

- Es un **lenguaje interpretado**, no compilado, usa tipado dinámico, fuertemente tipado.
- Es **multiplataforma**, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.
- Es un **lenguaje de programación multiparadigma**, el cual soporta varios paradigmas de programación como orientación a objetos, estructurada, programación imperativa y, en menor medida, programación funcional.

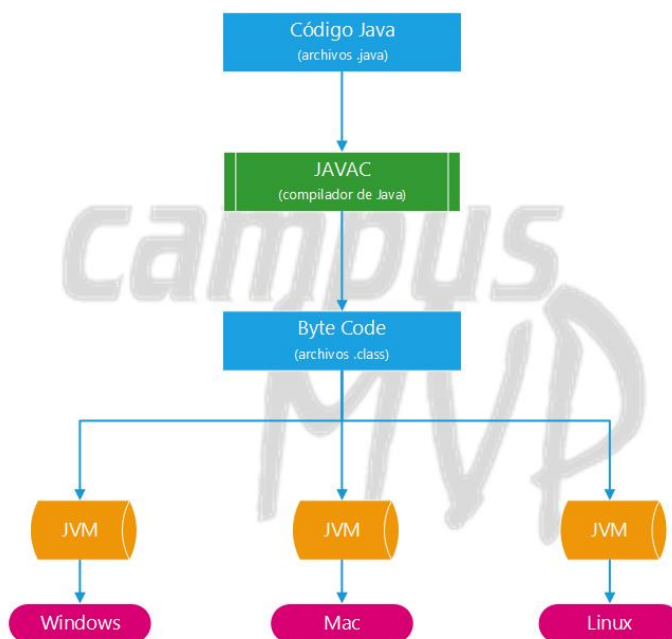
10. Características y posibilidades de desarrollo de una plataforma XAMPP.



XAMPP es un servidor independiente de plataforma, de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X, Apache, MariaDB/MySQL, PHP, Perl.

Se utiliza en proyectos pequeños, para iniciar el aprendizaje, ya que tiene limitaciones.

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.



La **JVM** (Java Virtual Machine) es una máquina virtual que permite a un ordenador ejecutar programas Java. La JVM toma el código bytecode resultante de compilar una aplicación Java y lo compila a su vez a código nativo de la plataforma en la que se está ejecutando.

El **JDK** (Java Development Kit), es un conjunto de herramientas de desarrollo para Java, nos encontraremos con el compilador javac que es el encargado de convertir nuestro código fuente (.java) en bytecode (.class), el cual posteriormente será interpretado y ejecutado con la JVM.

La **JVM** la usaremos en cuando queramos interpretar el código que he compilado en una aplicación en lenguaje java.

El **JDK** lo usaremos cuando queramos compilar una aplicación en lenguaje java.

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

Worldwide, Sept 2020 compared to a year ago:

Rank	Change	IDE	Share	Trend
1	↑	Visual Studio	25.29 %	+3.5 %
2	↑	Eclipse	16.72 %	-1.6 %
3	↓↓↓	Android Studio	12.57 %	-10.8 %
4	↑	Visual Studio Code	8.29 %	+3.4 %
5	↑	pyCharm	7.39 %	+2.6 %
6	↑	IntelliJ	6.03 %	+1.4 %
7	↓↓↓	NetBeans	5.37 %	-0.2 %
8		Xcode	4.23 %	+0.4 %



Microsoft Visual Studio es un entorno de desarrollo integrado de Microsoft, compatible con Windows, GNU/Linux, macOS . Se utiliza para desarrollar programas informáticos , así como sitios web , aplicaciones web , servicios web y aplicaciones móviles. Es soportado por múltiples lenguajes de programación (C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP)

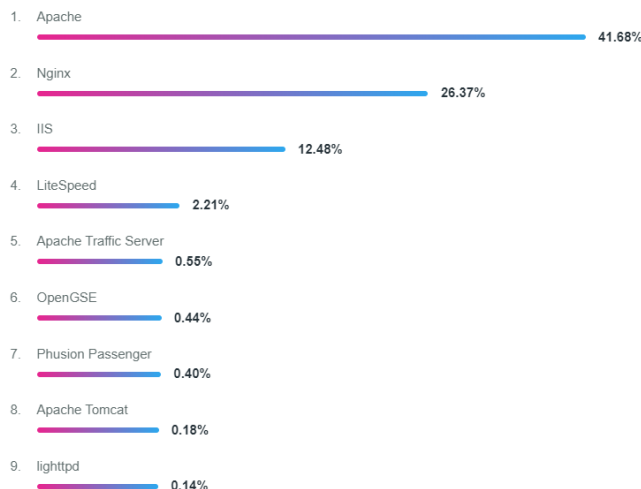


Eclipse es un entorno de desarrollo integrado (IDE) utilizado en programación de código abierto y multiplataforma desarrollado por Eclipse Foundation. Contiene un espacio de trabajo básico y un sistema de complementos extensible para personalizar el entorno. Eclipse está escrito principalmente en Java y su uso principal es para desarrollar aplicaciones Java, pero también puede usarse para desarrollar aplicaciones en otros lenguajes de programación a través de complementos.



NetBeans es un entorno de desarrollo integrado libre. Además del desarrollo de Java, tiene extensiones para otros lenguajes como PHP , C , C ++ , HTML5 , y JavaScript. Es un producto libre y gratuito sin restricciones de uso y compatible con Windows, macOS, Linux, Solaris.

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



Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, multi-plataforma, extensible y con posibilidad de instalar módulos para lenguajes como PHP, Perl, Python.



Nginx

NGINX es un servidor web de código abierto. En su versión inicial, funcionaba en servidores web HTTP. Hoy en día sirve como proxy inverso, balanceador de carga HTTP y proxy de correo electrónico para IMAP, POP3 y SMTP.

ISS (Internet Information Services)

ISS es un conjunto de servicios que transforman un sistema Microsoft Windows en un servidor capaz de ofrecer servicios Web, FTP y SMTP entre otros.

Nativamente el lenguaje dinámico soportado es ASP.NET sin embargo también es posible configurar para utilizar PHP, VBScript, Perl e incluso Java (Java Server Pages y Servlets) y prácticamente cualquier lenguaje como módulo adicional o a través de la interface CGI.

14. Apache HTTP vs Apache Tomcat

Ambos son servidores web que te permite trabajar con páginas web dinámicas y estáticas y ambos han sido desarrollados por "Apache Software Foundation".



Apache HTTP es un servidor web HTTP de código abierto y multiplataforma. Apache admite una variedad de características, muchas implementadas como módulos compilados que amplían la funcionalidad principal. Estos pueden variar desde esquemas de autenticación hasta admitir lenguajes de programación del lado del servidor como Perl, Python, Tcl y PHP, pero carece de la capacidad de administrar Java Servlets y JSP.

programación del lado del servidor como Perl, Python, Tcl y PHP, pero carece de la capacidad de administrar Java Servlets y JSP.

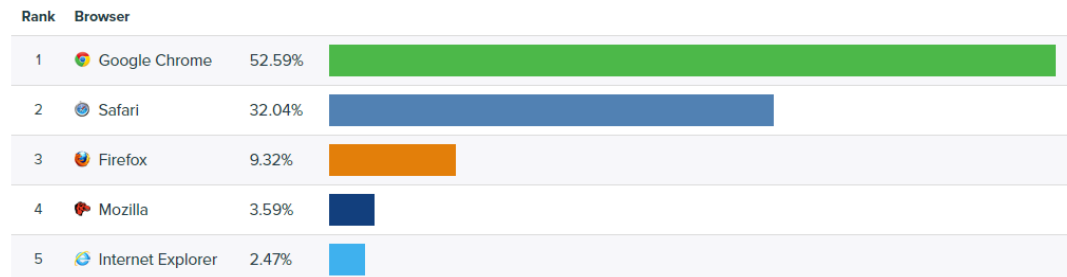


Apache Tomcat es una implementación de código abierto de las tecnologías Java Servlet, JavaServer Pages. Tomcat proporciona un entorno de servidor web HTTP "Java puro" en el que se puede ejecutar código Java ya que Tomcat se desarrolló originalmente como un medio para proporcionar la funcionalidad JSP de la que carecía Apache.

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

Browser Statistics September 2020

Here you see the market shares of the most used browsers for the current month.



	Desarrollador	Licencia de software	Motor de Renderizado actual	Otras características
Google Chrome	Google	Libre: BSD	Blink	<ul style="list-style-type: none"> Navegación segura Control de fallas Pestañas dinámicas Visor nativos de archivos PDFs Geolocalización
Safari	Fundación Mozilla	Software no libre, partes bajo GNU GPL	Gecko	<ul style="list-style-type: none"> Es quizá uno de los navegadores más veloces del mercado. Soporta extensiones. Proporciona un rendimiento óptimo de batería en portátiles. Sitios anclados. Sincroniza todo entre dispositivos tu teléfono, tableta y tu Mac.
Firefox	Safari	Libre: MPL, GNU GPL, GNU LGNU	Servo	<ul style="list-style-type: none"> Búsqueda progresiva Marcadores dinámicos Administrador de descargas Lector RSS Navegación privada

16. Generadores de documentación HTML

(PHPDoc): PHPDocumentor, ApiGen, ...

Un **generador de documentación** es una herramienta de programación que genera documentación destinada a los programadores (documentación de API) o a usuarios finales, o a ambos, a partir de un conjunto de código fuente especialmente documentado. Utilizan una forma específica de poner los comentarios, para que se pueda acceder a ellos como parte de aplicación web.

PHPDoc - Es una adaptación de javadoc para php que define un estándar oficial para comentar código php. PHPDoc ofrece tres ventajas principales:

- Hace comentarios que pueda leerse en un método estándar para animar a los programadores a definir y comentar los aspectos del código que normalmente se ignoran.
- Permite que los generadores de documentos externos como phpDocumentor puedan crear la documentación API en buen formato y fácil de entender.
- Permite que algunos IDEs como NetBeans o PhpStorm interpreten los tipos de variables y otras ambigüedades en el lenguaje de programación.

phpDocumentor - Es un generador de documentación de código abierto escrito en PHP.

Automáticamente analiza el código fuente PHP y produce la API de lectura y documentación del código fuente en una variedad de formatos. phpDocumentor genera la documentación en base al estándar formal PHPDoc. Es compatible con la documentación del código orientado a objetos y programación procedimental, además es capaz de crear documentos HTML, PDF, CHM y formatos Docbook.

ApiGen - Es la herramienta para crear documentación API profesional a partir del código fuente PHP, similar a phpDocumentor / phpDoc descontinuado. ApiGen tiene soporte para espacios de nombres PHP 5.3, paquetes, vinculación entre documentación, referencias cruzadas a clases estándar PHP y documentación general, creación de código fuente resaltado y soporte experimental para rasgos PHP 5.4.

17. Repositorios de software – sistemas de control de versiones:

GIT, CVS, Subversion, ...

Repositorio de software - Como una colección de software ordenado, clasificado y disponible para su uso con herramientas compatibles que lo usen para descargar software y manipular dicho software.

Sistema de control de versiones - Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

GIT - Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

CVS (Concurrent Versions System) - Es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros que forman un proyecto y permite que distintos desarrolladores colaboren. CVS se ha hecho popular en el mundo del software libre. Sus desarrolladores difunden el sistema bajo la licencia GPL.

Apache Subversion - Es una herramienta de control de versiones open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Es software libre bajo una licencia de tipo Apache/BSD.

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-W10ED.

	xxx-USED	xxx-W10ED
Sistema operativo	Ubuntu Server 20.04	Windows 10
Servidor administración remota	SSH	SSH
Servidor de transferencia de ficheros	SFTP (SSH)	SFTP (SSH)
Repositorio		GIT Hub
Servidor Web	Apache HTTP 2.4.46	Apache HTTP 2.4.46
SGBD	MySQL 8.0	MySQL 8.0
Navegador		Mozilla 81.0
IDE	NetBeans 12.0	NetBeans 12.0
Ofimática, multimedia, generador html,...		Libre office 6.4.6, Gimp 2.10.20
Cliente ssh	Filezilla 3.50	Filezilla 3.50

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.

	xxx-USEE
Sistema operativo	Ubuntu Server 20.04
Servidor administración remota	SSH
Servidor de transferencia de ficheros	SFTP (SSH)
Repositorio	GIT Hub
Servidor Web	Apache HTTP 2.4.46
SGBD	MySQL 8.0
IDE	NetBeans 12.0
Cliente ssh	NetBeans 12.0

20. 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 – Es una herramienta software para crear, administrar y gestionar sitios aplicación web sin saber programar.

- WordPress
- Joomla

ERP – Software cuyo propósito es apoyar a los clientes de la empresa, **dar tiempos rápidos de respuesta a sus problemas**, así como un eficiente manejo de información que permita la toma de decisiones y **minimizar los costes**.

- SAP
- Microsoft Dynamics