



# TEMA 1-DESARROLLO WEB EN ENTORNO SERVIDOR

CRISTINA FALAGÁN CUESTA

## Contenido

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

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

**IP:** Su función principal es el uso bidireccional en origen o destino de comunicación para transmitir datos que transfiere paquetes conmutados a través de distintas redes físicas. Pertenece a la capa de red.

**TCP:** Las aplicaciones pueden comunicarse en forma segura independientemente de las capas inferiores. Esto significa que los routers (que funcionan en la capa de red) sólo tiene que enviar los datos. Pertenece a la capa de transporte.

**HTTP:** Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. El cliente realiza una petición enviando un mensaje, con cierto formato al servidor. El servidor le envía un mensaje de respuesta, utiliza el puerto 80. Pertenece a la capa de aplicación.

**HTTPS:** Basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, utiliza un cifrado basado en SSL/TLS para crear un canal cifrado. De este modo se consigue que la información sensible (usuario y claves de paso normalmente). El puerto estándar para este protocolo es el 443. Pertenece a la capa de aplicación.

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

Un cliente realiza peticiones a otro programa, el servidor, es quien le da respuesta. Las aplicaciones web se basan en el protocolo HTTP, el cual crea un vínculo entre el usuario y la aplicación ya que el navegador se comunica con el servidor web mediante este protocolo.

Las funcionalidades en los entornos cliente/servidor de la Web suelen estar agrupadas en diferentes capas, cada una centrada en la gestión de un aspecto determinado del sistema web, estas capas son:

- capa de presentación**
- Capa de negocio**
- Capa de presencia o datos.**

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

**HEAD:** pide una respuesta

**GET:** Pide una representación del recurso especificado

**POST:** Somete los datos que sea procesado para el recurso identificado.

**PUT:** Sube, cargo o realiza un upload de un recurso específico

**DELETE:** Borra el recurso específico.

**TRACE:** solicita al servidor que envíe de vuelta en un mensaje.

**OPTIONS:** Devuelve los métodos HTTP que el servidor soporta para un URL específico.

**CONNECT:** Se utiliza para saber si se tiene acceso a un host

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

**Capa de Presentación:** Será la que vea el usuario. Interfaz gráfica. Está situada en cliente.

**Capa de negocio:** Gestiona las funcionalidades de la aplicación. Utiliza datos para ejecutar un proceso y obtener un resultado.

**Capa de persistencia o Datos:** En ella residen los datos y es la encargada de acceder a estos. Formada por uno o más gestores de datos.

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

En diseño de software el *front-end* es la parte del software que interactúa con los usuarios y el *back-end* es la parte que procesa la entrada desde el *front-end*. La separación del sistema en *front-ends* y *back-ends* es un tipo de abstracción que ayuda a mantener las diferentes partes del sistema separadas. Partes que componen el back-end:

- Servidores web.
- Ficheros escritos en lenguajes de programación que generan el HTML que formara parte del front-end.
- Gestores de bases de datos para que consulte el servidor web.

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

**Aplicaciones web estáticas:** Página web cuya interacción no conlleva ningún tipo de acción, ni en la propia página, ni genera respuesta alguna por parte del servidor. este tipo de aplicaciones suele realizarse utilizando el lenguaje HTML.

**Aplicaciones web dinámicas:** Se refiere a aquellas aplicaciones en las que la interacción del cliente con el recurso recibido por parte del servidor produce algún tipo de cambio en la visualización del mismo (cambios de formato, ocultación de partes del documento, creación de elementos nuevos, etc.).

**Aplicaciones web interactivas:** Se basan en que dicha interacción hace que se genere un diálogo entre el cliente y el servidor. Desde el punto de vista del modelo de programación, la lógica asociada al inicio y gestión de dicho diálogo puede ser ejecutada tanto en el cliente como en el servidor (e incluso en ambos).

## 7. Componentes de una aplicación web.

Las aplicaciones Web son proporcionadas por un servidor Web y utilizada por usuarios que se conectan desde cualquier punto vía clientes Web (browsers o navegadores). La arquitectura de un Sitio Web tiene tres componentes principales:

- Un servidor Web con o sin Sistema Gestor de Base de Datos
- Lenguaje de programación
- Módulo de PHP.

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

**Ejecutado en el cliente:** Programas escritos en un lenguaje que interpreta el cliente web (navegador)(JavaScript). Pueden ser ficheros aparte o ficheros incrustados en HTML.

**Ejecutados en servidor:** Programas que generan respuestas (Programas CGI, PHP, ASP, JSP, PERL...)

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

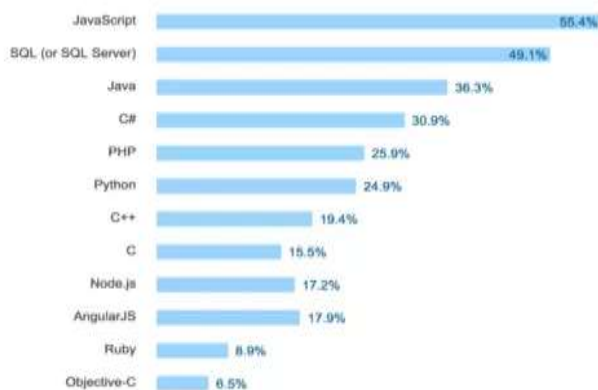
**PERL:** Es el lenguaje más utilizado para la creación de programas CGI en los servidores web. Es más rápido que los programas shell script de UNIX, puede leer y escribir ficheros binarios, y puede procesar ficheros grandes. La ventaja más importante de PERL sobre C es que PERL no necesita ser recompilado, es un lenguaje interpretado.

**ASP:** Tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente.

**JSP:** Tecnología para generar páginas web de forma dinámica en el servidor, basado en scripts que utilizan una variante del lenguaje Java. Permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos, así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento 'HTML' en el archivo jsp.

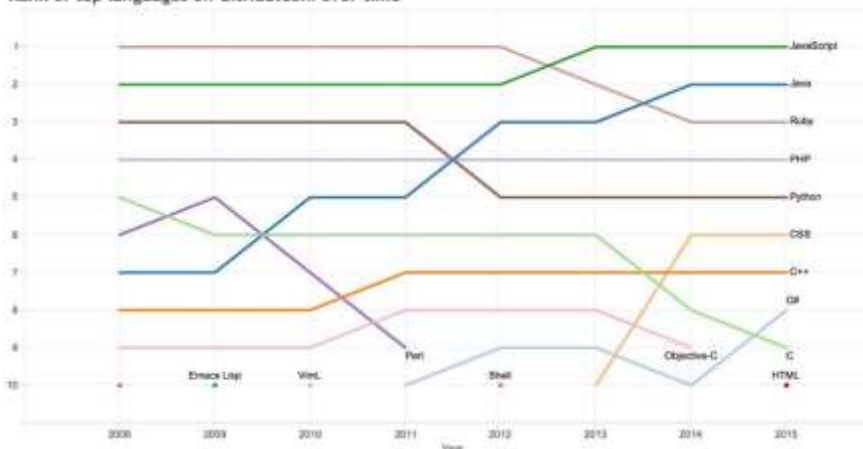
**PHP:** Lenguaje de programación interpretado. Destaca por su capacidad de mezclarse con el código HTML. Tiene un alto nivel de uso.

**Python:** Su nivel es normal/alto.



Tecnologías más populares de 2016

Rank of top languages on GitHub.com over time



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

### Características que lo compone:

- Servidor web(Apache)
- SGBD(MariaDB)
- Lenguajes de programación(PHP,Perl,Python)

### Posibilidades:

- En Windows existen dos versiones una con instalador y otra portable.
- Tiene una licencia publica, es decir orientada a proteger la libre distribución, modificación y uso del software.
- Es multiplataforma.
- Es un servidor independiente de plataforma.

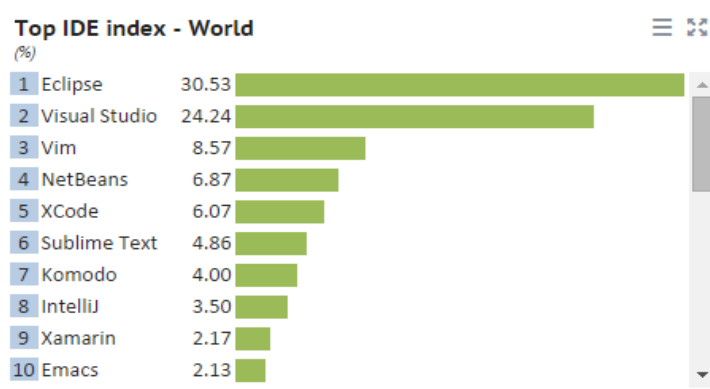
## 11. En que casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación.

En el entorno de desarrollo, cuando se quiere ejecutar un software basado en java o implementar aplicaciones escritas en este lenguaje.

En el entorno de explotación, cuando se quiere ejecutar JSP, es necesario instalar JRE, también para los servlets es necesario JDK.

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

- <https://www.keycdn.com/blog/best-ide/>



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

**Apache:** Permite emplear diversos lenguajes en el lado del servidor (PHP, Perl, Python...) así como incorporar características como la compresión de datos, las conexiones seguras y la utilización de URLs amigables. Es multiplataforma.

**Nginx:** Servidor de web ligero que funciona para múltiples plataformas. Es usado por algunos sitios como wordpress.

**Microsoft IIS:** Tiene un gran número de módulos, pero solo funciona para Windows.

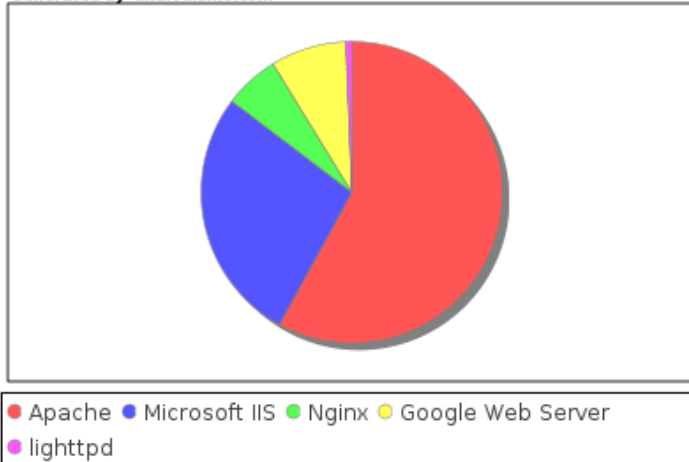
**Google Web Server(GWS):** Es el tercero más utilizado. servidor personalizado de desarrollo propio que se ejecuta en sistemas UNIX como GNU/Linux.

**Lighttpd:** Este servidor web ofrece soluciones rápidas, seguras y flexibles. Un servidor de software libre que está especialmente pensado para aquellos

servidores que tengan que soportar mucha carga y que, por lo tanto, necesiten de muchísima velocidad ya que Lighttpd es muy ligero y apenas consume recursos

### Servidores web más usados

Generated by ChartGizmo.com



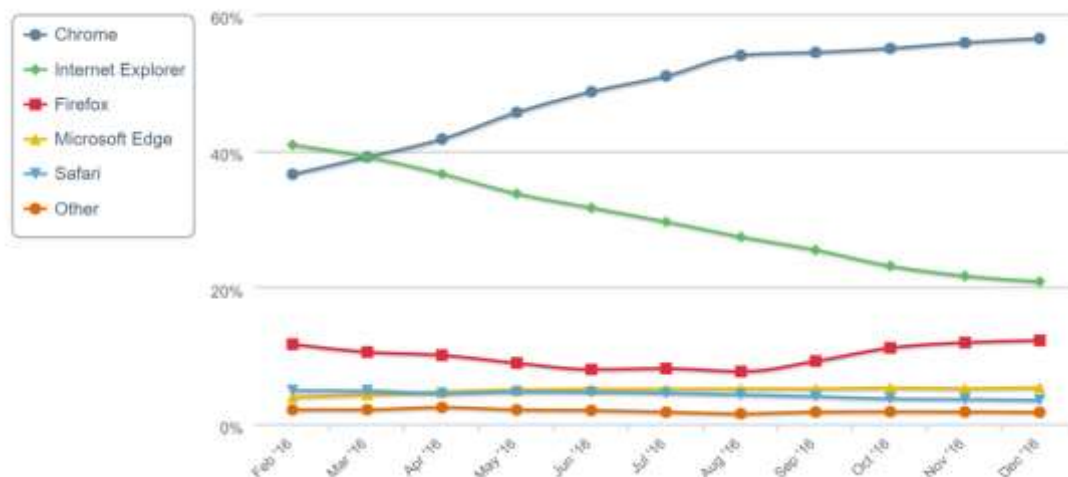
#### 14. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).

**Chrome:** Es uno de los más conocidos y más usados, básicamente porque es el que asegura una velocidad mayor. Se inicia rápidamente desde el escritorio, carga las páginas de forma instantánea y ejecuta aplicaciones web complejas a gran velocidad.

La ventana del navegador de Chrome es intuitiva y sencilla. Está diseñado para ofrecer una mayor seguridad en la web, al actualizarse automáticamente para que siempre tengamos las últimas mejoras en este campo.

**Firefox:** uno de los más veteranos (salió en el año 2003) es sólido, estable y presenta muy pocos errores. Firefox, el segundo navegador más utilizado en Internet, se caracteriza por ser un programa independiente, y para muchos es su favorito porque no tiene ánimo de lucro, además, es un navegador altamente personalizable, ya que cuenta con un amplio abanico de temas y complementos. Pero lo mejor de todo son las extensiones.

**Opera:** Es uno de los más veteranos y, durante muchos años, ha sido de los más utilizados en los teléfonos móviles, hasta la popularización de los *smartphones*. No tiene ánimo de lucro. Usa el mismo motor que Google Chrome, por lo que se pueden utilizar en él las mismas extensiones disponibles para el navegador de Google. Además, incorpora lo que ellos llaman el “Estante“, una reinención de los marcadores o favoritos que permite ir guardando páginas que interesan para leerlas posteriormente. Opera es también altamente personalizable y su velocidad es buena.



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

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.

### 16. Repositorios de software – sistemas de control de versiones para aplicaciones web: GIT, CVS, Subversion, ...

Un sistema de control de versiones registra todos los cambios hechos en uno o más proyectos, guardando así versiones del producto en todas sus fases del desarrollo. Existen sistemas de control de versiones centralizados, distribuidos o locales.

**Apache Subversion:** 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 licencia.

Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Entre dos revisiones sólo guarda el conjunto de modificaciones (delta), optimizando así al máximo el uso de espacio en disco. SVN permite al usuario crear, copiar y borrar carpetas. Permite acceder al repositorio a través de redes.

**GIT:** La principal diferencia entre GIT y otros controles de versión es que cada vez que confirmas un cambio, o guardas el estado de tu proyecto en Git, él básicamente hace una foto del aspecto de todos tus archivos en ese momento, y guarda una referencia a esa instantánea. Para ser eficiente, si los archivos no se han modificado, Git no almacena el archivo de nuevo, sólo un enlace al archivo anterior idéntico que ya tiene almacenado.



**CVS:** Utiliza una arquitectura cliente-servidor: un servidor guarda la(s) versión(es) actual(es) del proyecto y su historial. Los clientes se conectan al servidor para sacar una copia completa del proyecto. Esto se hace para que eventualmente puedan trabajar con esa copia y más tarde ingresar sus cambios.

17. 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.

<b>Maquina cliente: CFCW7ED</b>	
<b>Sistema Operativo</b>	<b>Windows 7 Enterprise x64</b>
<b>Memoria RAM</b>	<b>1024MB</b>
<b>Usuario y contraseña</b>	<b>miadmin-paso</b>
<b>Configuración de la red</b>	
<b>Ip</b>	<b>192.168.3.19</b>
<b>Puerta de enlace</b>	<b>192.168.3.1</b>
<b>DNS</b>	<b>8.8.8.8</b>
<b>Software</b>	
<b>Navegador</b>	<b>Firefox, Chrome, Opera, Explorer...</b>
<b>IDE</b>	<b>Netbeans</b>
<b>Editor</b>	<b>Komodo edit</b>
<b>Otros</b>	<b>Office, Gimp, FileZilla</b>

<b>Máquina ED:CFCUSED</b>	
<b>Sistema Operativo</b>	<b>Ubuntu 16.04 x64</b>
<b>Memoria RAM</b>	<b>512MB</b>
<b>Usuario y contraseña</b>	<b>miadmin-paso/operadorweb-paso</b>
<b>Configuración de la red</b>	
<b>Ip</b>	<b>192.168.3.109</b>
<b>Puerta de enlace</b>	<b>192.168.3.1</b>
<b>DNS</b>	<b>8.8.8.8</b>
<b>Servicios</b>	
<b>Servidor web</b>	<b>Apache</b>
<b>Servicio SSH</b>	<b>OpenSSH Server</b>
<b>Servicio FTP</b>	<b>SFTP   usuario: operadorweb</b>
<b>SGBD</b>	<b>MySQL</b>

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

<b>Máquina ED:CFCUSEE</b>	
<b>Sistema Operativo</b>	<b>Ubuntu 16.04 x64</b>
<b>Memoria RAM</b>	<b>512MB</b>
<b>Usuario y contraseña</b>	<b>miadmin-paso/operadorweb-paso</b>
<b>Configuración de la red</b>	
<b>Ip</b>	<b>192.168.3.209</b>
<b>Puerta de enlace</b>	<b>192.168.3.1</b>
<b>DNS</b>	<b>8.8.8.8</b>
<b>Servicios</b>	
<b>Servidor web</b>	<b>Apache</b>
<b>Servicio SSH</b>	<b>OpenSSH Server</b>
<b>Servicio FTP</b>	<b>SFTP   usuario: operadorweb</b>
<b>SGBD</b>	<b>MySQL</b>