

EJERCICIOS

TEMA 1

DESARROLLO WEB EN ENTORNO SERVIDOR

Pablo Cidón Barrio

19 de septiembre de 2017

Sumario

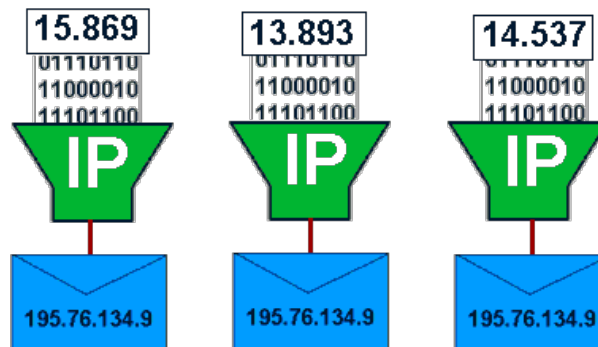
1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.....	4
Protocolo IP:.....	4
Protocolo TCP:.....	4
Protocolo HTTP:.....	4
Protocolo HTTPS:.....	1
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.....	1
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.....	2
Métodos HTTP:.....	2
4. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.....	3
5. Modelo de división funcional front-end / back-end para aplicaciones web.....	4
Front-end:.....	4
Back-end:.....	4
6. Página web estática – página web dinámica – aplicación web.....	5
Página web estática:.....	5
Página web dinámica:.....	5
Aplicación web:.....	5
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
En el lado del cliente:.....	6
En el lado del servidor:.....	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.....	7
11. 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.....	7
12. IDE más utilizados (características y grado de implantación actual).....	8
13. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).....	10
14. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).....	10

15. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen,	11
PhpDocumentor:.....	11
PhpDoc:.....	11
Doxygen:.....	11
ApiGen:.....	11
16. Repositorios de software – sistemas de control de versiones para aplicaciones web para aplicaciones sobre los Web: GIT, CVS, Subversion,	11
Git:.....	11
CVS:.....	11
Subversion:.....	11
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.....	12
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.....	12

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

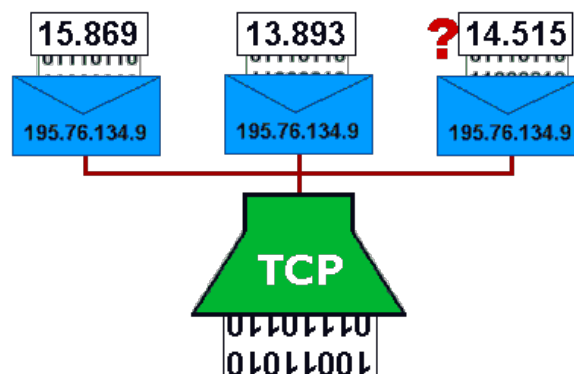
Protocolo IP:

Es un protocolo de comunicación de datos digitales clasificado en la capa de red según el modelo OSI. Su función principal es el uso bidireccional en origen o destino de comunicación para transmitir datos mediante un protocolo que transfiere paquetes conmutados a través de distintas redes físicas previamente enlazadas según la norma OSI de enlace de datos.



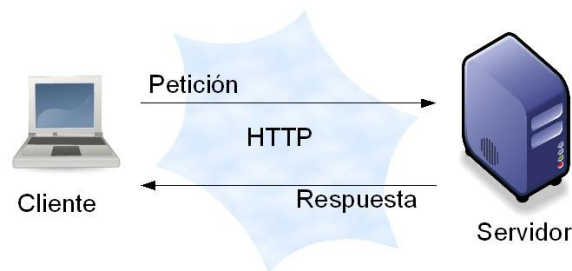
Protocolo TCP:

Protocolo de Control de Transmisión, que es el más utilizado en internet. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto. TCP da soporte a muchas de las aplicaciones más populares de Internet y protocolos de aplicación HTTP, SMTP, SSH y FTP.



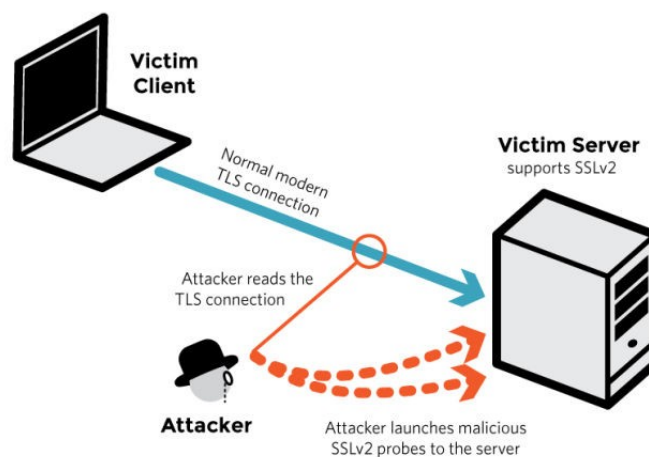
Protocolo HTTP:

El protocolo de comunicación que permite las transferencias de información en la World Wide Web. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto permite a las aplicaciones web rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.



Protocolo HTTPS:

Es un protocolo de aplicación basado en HTTP, la versión segura del mismo. El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.



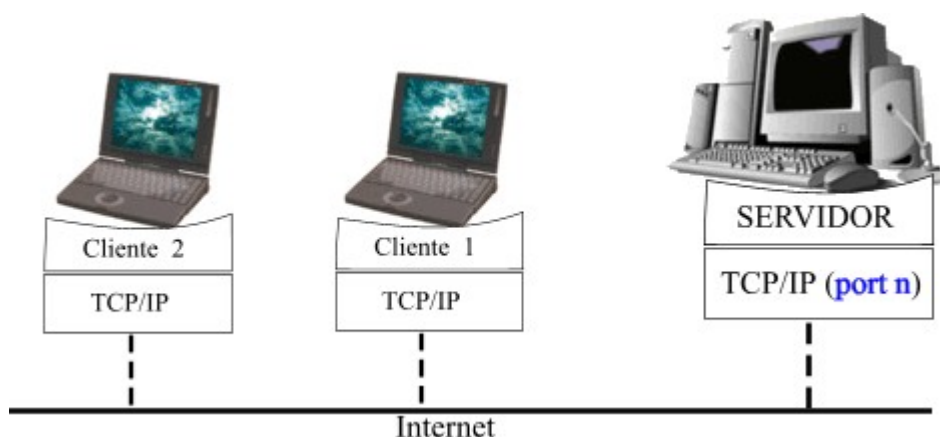
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.

Un servidor es una aplicación que ofrece un servicio a usuarios de Internet; un cliente es el que pide ese servicio. Una aplicación consta de una parte de servidor y una de cliente, que se pueden ejecutar en el mismo o en diferentes sistemas.

Los usuarios invocan la parte cliente de la aplicación, que construye una solicitud para ese servicio y se la envía al servidor de la aplicación que usa TCP/IP como transporte.

Algunos servidores esperan las solicitudes en puertos bien conocidos de modo que sus clientes saben a qué zócalo IP deben dirigir sus peticiones. El cliente emplea un puerto arbitrario para comunicarse. Los clientes que se quieren comunicar con un servidor que no usa un puerto bien conocido tienen otro mecanismo para

saber a qué puerto dirigirse. Este mecanismo podría usar un servicio de registro como Portmap, que utiliza un puerto bien conocido.



Fuente: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>

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

Métodos HTTP:

GET:

El método GET solicita una representación del recurso especificado. Las solicitudes con GET sólo deben recuperar datos y no deben tener ningún otro efecto.

HEAD:

El método HEAD pide una respuesta idéntica a la de una solicitud GET, pero sin el cuerpo de respuesta. Esto es útil para recuperar la meta-información escrita en los encabezados de respuesta, sin tener que transportar todo el contenido.

POST:

El método POST solicita que el servidor acepte la entidad incluida en la solicitud como un nuevo subordinado del recurso web identificado por el URI. Los datos POST podrían ser, por ejemplo, una anotación para los recursos existentes; un mensaje para un tablón de anuncios, un grupo de noticias, una lista de correo o un hilo de comentarios; un bloque de datos que es el resultado de la presentación de un formulario web a un proceso de tratamiento de datos; o un elemento para agregar a una base de datos.

PUT:

El método PUT solicita que la entidad incluida se almacene bajo URI suministrada. Si el URI se refiere a un recurso ya existente, se modifica; si el URI no apunta a un recurso existente, entonces el servidor puede crear el recurso con ese URI.

DELETE:

El método DELETE elimina el recurso especificado.

TRACE:

El método TRACE se hace eco de la solicitud recibida para que un cliente pueda ver qué (si es que) los cambios o adiciones han sido realizados por los servidores intermedios.

OPTIONS:

El método OPTIONS devuelve los métodos HTTP que el servidor admite para la URL especificada. Esto se puede utilizar para comprobar la funcionalidad de un servidor web solicitando '*' en lugar de un recurso específico.

CONNECT:

El método CONNECT convierte la conexión de la petición a un túnel TCP / IP transparente, usualmente para facilitar la comunicación cifrada en SSL (HTTPS) a través de un proxy HTTP no cifrado. Vea el túnel HTTP CONNECT .

PATCH:

El método PATCH aplica modificaciones parciales a un recurso.

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

Es un modelo de desarrollo del software cuyo objetivo principal es la separación entre las distintas que componen un software. Así es más sencillo crear diferentes interfaces sobre un mismo sistema sin requerir cambios en la capa de datos o en la lógica.

La principal ventaja es que el desarrollo se puede llevar a cabo en varios niveles, y en caso de algún cambio, que este solo afecte al nivel requerido sin tener que revisar entre el código de los otros niveles. También permite distribuir el trabajo, ya que cada grupo podrá ocuparse de un nivel.

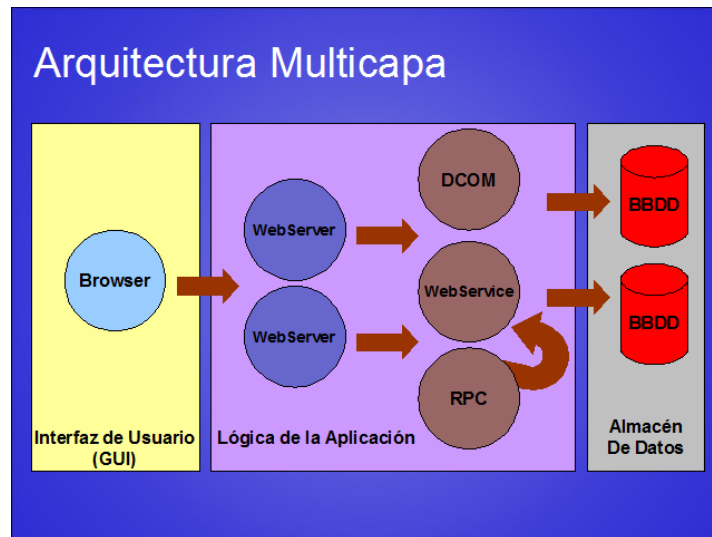
Actualmente el más utilizado se realiza en tres capas:

Capa de presentación: es la que ve el usuario, le comunica la información y captura la información del usuario. También se conoce como interfaz gráfica y debe ser fácil de usar para el usuario. Esta capa se comunica únicamente con la capa de negocio.

Capa de negocio: en ella se encuentran los programas que se ejecutan, reciben las peticiones del usuario y envía las respuestas de negocio. Se le llama así porque es donde se establecen las normas que deben seguirse. Se conecta con la capa de presentación para recibir solicitudes y enviar resultados y con la capa de datos para solicitar el almacenamiento o recuperación de datos.

Capa de datos: es donde se encuentran los datos y se encarga de acceder a los mismos. Está compuesta por uno o varios gestores de bases de datos que almacenan los datos o recuperan los mismos desde la capa de negocio.

Todas estas capas pueden encontrarse en un mismo ordenador, aunque la capa de presentación es normal que se encuentre en multitud de ordenadores.



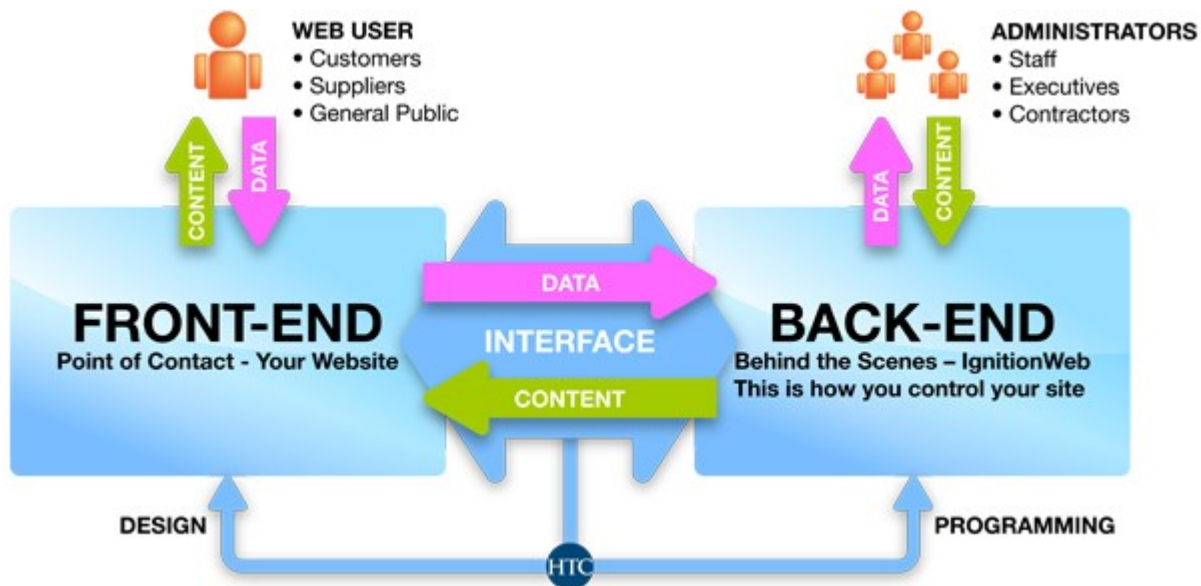
5. Modelo de división funcional front-end / back-end para aplicaciones web.

Front-end:

Está orientado a lenguaje de marcas y al lenguaje de programación web de ejecución en equipos clientes, sin necesidad de uso de servidores externos. Casi todo lo que ves en la pantalla cuando accedes a una web es desarrollo front-end, la estructuración de los apartados, tamaños, márgenes entre estructuras, tipos de letra, colores, adaptación para distintas pantallas, los efectos de ratón, teclado, movimientos, desplazamientos, efectos visuales...Esto sería la base origen en la que se centra la especialidad front-end, dar formato a contenidos, desarrollo del aspecto de la web y manipular resultados de datos obtenidos.

Back-end:

Está enfocado sólo a lenguajes de programación, orientado a funcionamientos, trabajar con datos internos, crear aplicaciones que controlen datos de la base de datos de la web, para poder ser consultados por el usuario. Creación de funciones que realicen una acción o acciones que controlen el buen funcionamiento de las aplicaciones, Crear estructuras de operaciones y cálculos para devolver unos resultados. Es una especialidad más orientada a una programación menos intuitiva y en ocasiones algo más compleja, ambas especialidades van a ir ligadas, aunque no nos meteremos más a fondo porque sería ya un tema que sale de nuestro dominio.



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

Página web estática:

Muestra información permanente, donde el usuario se limita a obtener información sin tener que interactuar con la página. Se construyen con enlaces entre las diferentes páginas que componen el sitio.

Una de las ventajas está en lo económico que resulta crearlas, pero la gran desventaja se encuentra en lo difícil que resulta su actualización.

Página web dinámica:

Las páginas web dinámicas son aquellas en las que la información presentada se genera a partir de una petición del usuario de la página. Tiene asociada una Base de Datos desde la que se permite visualizar la información contenida en ella.

Una de las ventajas de este tipo de páginas es que permiten almacenar y hacer actualizaciones de la información contenida en la misma, así como también modificaciones dinámicas de la estructura y del diseño por parte de su propietario.

Para la creación de este tipo de páginas, deberán utilizarse etiquetas HTML y algún lenguaje de programación que se ejecute “del lado del servidor”, es decir, que esté alojado en el mismo servidor en el que está alojada la página web.

Los lenguajes utilizados para la generación de este tipo de páginas son principalmente: Perl CGI, PHP, JSP y ASP.

Aplicación web:

Es el conjunto de herramientas que los usuarios pueden usar accediendo a un servidor web a través de Internet, o de un navegador.

Es una aplicación que se codifica en un lenguaje soportado por los navegadores, que confían en la ejecución del navegador.

Son populares debido a lo práctico del navegador como cliente ligero, independiente del sistema operativo, y con facilidad para actualizar y mantener aplicaciones web sin distribuir ni instalar software. Algunos ejemplos son las tiendas en línea o Wikipedia.

7. Componentes de una aplicación web.

Toda aplicación web, debe componerse de un navegador web, un servidor de aplicaciones y un servidor de base de datos.

El servidor de base de datos es el programa que permite guardar, modificar o mostrar los datos que se almacenan en la aplicación.

El servidor de aplicaciones es el servidor que se encuentra en una red y que ejecuta ciertas aplicaciones. Se trata de un dispositivo que proporciona servicios de aplicación a las computadoras del cliente.

El navegador web es el programa que permite el acceso a la web, interpretando la información de los distintos archivos y sitios web para que estos puedan ser visualizados.

8. 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 del cliente:

Programas ejecutados	Lenguajes utilizados
Netbeans	HTML
Notepad ++	JavaScript
Editores de foto/video	Java
	Visual Basic
	Flash
	CSS

En el lado del servidor:

Programas ejecutados	Lenguajes utilizados
Apache	CGI
MySQL	Perl
	ASP
	PHP
	JSP

[Consultar fuente lenguajes](#)

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

CGI: Es el más antiguo que existe en la programación de páginas dinámicas de servidor. Actualmente se encuentra desfasado.

Perl: Es un lenguaje de programación interpretado. Esto supone que el código de los scripts no se compila, sino que se pone en marcha usando el que ya está escrito. Se puede ejecutar a través de otros lenguajes.

ASP: Es la tecnología desarrollada por Microsoft para crear páginas dinámicas en servidor. Se escribe en la misma página web usando Visual Basic.

PHP: Es un lenguaje de programación del lado del servidor gratuito, rápido, con una gran librería de funciones y gran cantidad de documentación.

JSP: Es una tecnología orientada a crear páginas web con programación de Java. Estas aplicaciones pueden ejecutarse en varios servidores, de múltiples plataformas. Podemos escribirlas con un editor de HTML o XML.

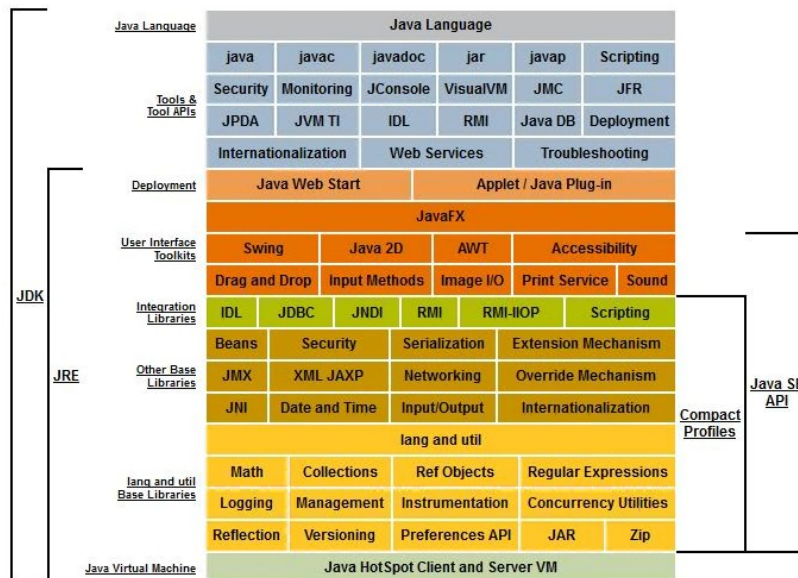
10. Características y posibilidades de desarrollo de una plataforma XAMPP.

1. Es de licencia pública.
2. Es multiplataforma.
3. Solamente requiere la descarga y ejecución.
4. Xampp es una herramienta muy práctica que nos permite instalar el entorno MySQL, Apache y PHP, suficiente para empezar proyectos web o revisar alguna aplicación localmente. Además, trae otros servicios como servidor de correos y servidor FTP.

11. 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.

Cuando se realiza la programación en Java, es necesario la instalación de la máquina virtual de Java, tanto en el entorno de desarrollo como en el de explotación.

En cambio, en el entorno de desarrollo hay que instalar el JDK, ya que es lo que provee al sistema de herramientas de desarrollo. Pero en el entorno de explotación, tendremos que instalar el JRE, que es la herramienta necesaria para la ejecución de programas en Java.



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

Visual Studio:

- Botón de barra de herramientas y ampliación de texto
- Opciones de tamaño de texto en los editores
- Personalización de colores en los editores
- Personalización de métodos abreviados de teclado
- Función Autocompletar para métodos y parámetros

Eclipse: dispone de un Editor de texto con un analizador sintáctico. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (*wizards*) para creación de proyectos, clases, tests, etc., y refactorización.

Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversion.⁴ e integración con Hibernate.⁵

Netbeans:

- Gestión de la interfaz de usuario (menús y barras de herramientas)
- Gestión de configuración de usuario
- Gestión de almacenamiento (guardar o cargar algún tipo de dato)
- Gestión de ventana
- Marco Asistente (soporta diálogos paso a paso)

- Librería visual de Netbeans
- Herramientas de desarrollo integrado

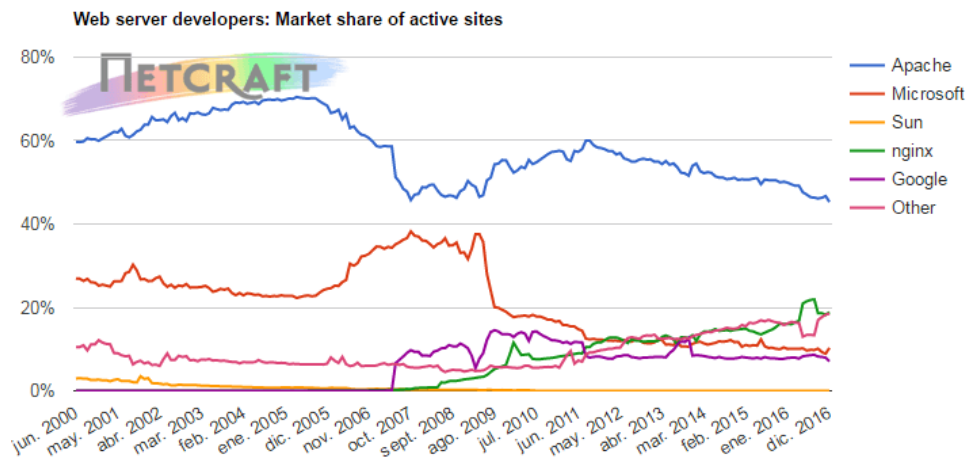
Pycharm:

- Asistencia y análisis de codificación, con finalización de código , sintaxis y resaltado de errores, integración de linter y arreglos rápidos
- Navegación de proyectos y códigos: vistas de proyectos especializados, vistas de estructura de archivos y saltos rápidos entre archivos, clases, métodos y usos
- Refactoring de Python: incluyendo renombrar, extraer método, introducir variable, introducir constante, tirar hacia arriba, empujar hacia abajo y otros
- Soporte para frameworks web: Django , web2py y Flask
- Depurador integrado de Python
- Prueba unitaria integrada, con cobertura de código línea por línea
- Desarrollo de Google App Engine Python
- Integración de control de versiones: interfaz de usuario unificada para Mercurial, Git, Subversion , Perforce y CVS con listas de cambios y combinación

Android Studio:

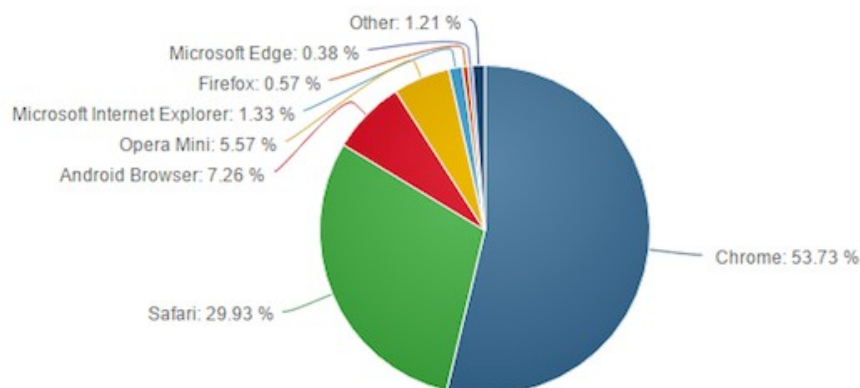
- Integración de ProGuard y funciones de firma de aplicaciones.
- Renderizado en tiempo real
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Soporte para construcción basada en Gradle.
- Refactorización específica de Android y arreglos rápidos.
- Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear.
- Soporte integrado para Google Cloud Platform, que permite la integración con Google Cloud Messaging y App Engine.
- Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones

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



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

- Chrome = Google Chrome.
- IE/Edge = Microsoft Internet Explorer/Microsoft Edge.
- Firefox = Mozilla Firefox (identified as Mozilla before 2005)
- Safari = Apple Safari (and Konqueror. Both identified as Mozilla before 2007)
- Opera = Opera (from 2011; Opera Mini is included here)



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

PhpDocumentor:

Es una herramienta que hace posible generar documentación directamente desde su código fuente PHP. Con esto usted puede proporcionar a sus consumidores con más información con respecto a la funcionalidad integrada en su fuente y no sólo lo que se puede utilizar para ellos desde su interfaz de usuario.

La documentación generada por phpDocumentor 2 no pretende ser un sustituto de la documentación convencional, sino más bien una documentación complementaria o de referencia.

PhpDoc:

Es una adopción de Javadoc para el mundo de PHP. PHPDoc está escrito en PHP. Le ofrece una manera de generar una documentación de API de código orientado a objetos y de procedimiento con cierto marcado en su fuente. PHPDoc es un proyecto de código abierto y se distribuye bajo la licencia de PHP. Eso significa que puedes usarlo en proyectos comerciales.

Doxygen:

Es la herramienta estándar de facto para generar documentación a partir de fuentes C++ anotadas, pero también soporta otros lenguajes de programación populares como C, Objective-C, C#, PHP, Java, Python, IDL (Corba, Microsoft y UNO / OpenOffice flavors), Fortran, VHDL, Tcl, y hasta cierto punto D.

ApiGen:

Es una distribución del repositorio en la nube de GitHub.

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

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. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end.

CVS:

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 (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren. CVS se ha hecho popular en el mundo del software libre. Sus desarrolladores difunden el sistema bajo la licencia GPL.

Subversion:

Apache Subversion (abreviado frecuentemente como **SVN**, por el comando *svn*) es una herramienta de control de versiones 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.

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.

US-ED	W7-ED
Apache2	Firefox 55.0.3 (32-bit)
PHP 7.0	Chrome 61.0.3163.91 (32-bit)
MySQL 5.7	Opera 41.0.2631.80 (32-bit)
	Safari 5.1.7 (32-bit)
	Filezilla
	Notepad++
	Netbeans 8.2
	Office 2007

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.

US-EE
Apache2
PHP 7.0
MySQL 5.7