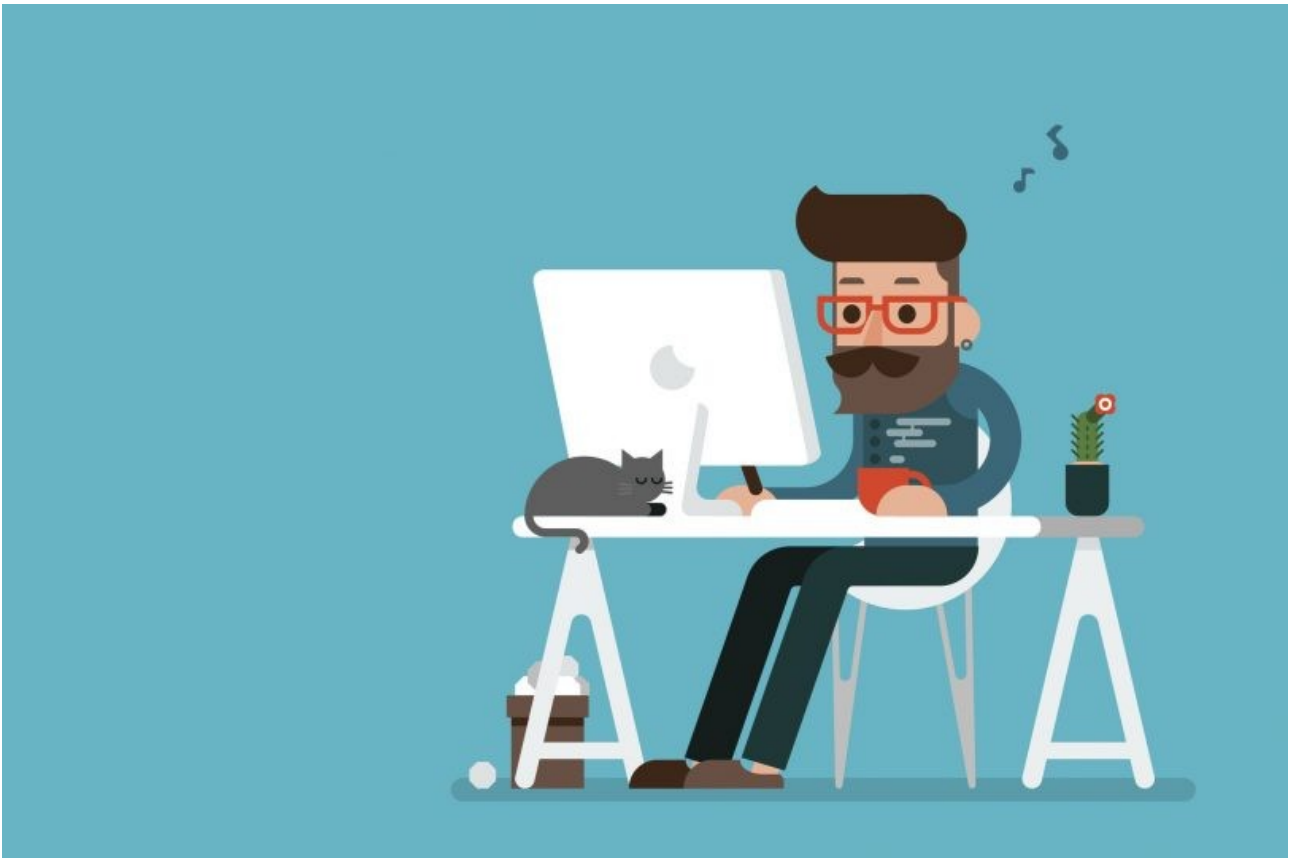


*desarrollo de aplicaciones web en el lado del servidor*



*Ismael heras salvador DAW2*

## **TEMA 1**

### **EJERCICIOS DEL TEMA 1**

# Sumario

|   |    |
|---|----|
| TEMA1.....  | 4  |
| Ejercicios Tema1.....   | 4  |
| 1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.....  | 4  |
| 2.Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.....   | 5  |
| 3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.....  | 6  |
| GET.....  | 6  |
| HEAD.....   | 6  |
| POST.....   | 6  |
| PUT.....  | 6  |
| DELETE.....   | 6  |
| CONNECT.....  | 7  |
| OPTIONS.....  | 7  |
| TRACE.....  | 7  |
| PATCH.....  | 7  |
| 4. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.....                                    | 7  |
| 5. Modelo de división funcional front-end / back-end para aplicaciones web.....   | 9  |
| BACK-END.....   | 9  |
| FRONT-END.....  | 11 |
| 6. Página web estática – página web dinámica – aplicación web – mashup .....  | 12 |
| Pagina web statica.....   | 12 |
| Pagina web dinamica.....  | 13 |
| aplicación web – mashup.....  | 14 |
| 7. Componentes de una aplicación web.....   | 14 |
| Servidor web.....   | 15 |
| Capa de aplicaciones.....   | 17 |
| SGBD - Sistema Gestor de Bases de Datos.....  | 17 |
| 8. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.....          | 18 |
| 9. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).....                         | 20 |
| ASP.NET.....  | 20 |
| PERL.....   | 21 |
| PHP.....  | 21 |
| 10. Características y posibilidades de desarrollo de una plataforma XAMPP.....  | 23 |
| 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 entrono de explotación..... | 25 |
| 12. IDE más utilizados (características y grado de implantación actual).....  | 27 |
| 13. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).....   | 28 |
| Servidor Apache.....  | 29 |
| Servidores nginx.....   | 29 |
| Servidor de Google.....   | 30 |
| Servidor IIS de Microsoft.....  | 30 |
| 14. Apache HTTP vs Apache Tomcat.....   | 30 |
| 15. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).....  | 31 |
| Google Chrome.....  | 32 |
| Mozilla firefox.....  | 32 |
| 16. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, ..... phpDocumentor.....   | 33 |

|   |    |
|---|----|
| PhpDoc.....   | 33 |
| 17. Repositorios de software – sistemas de control de versiones: GIT , CVS, Subversion, ..... | 34 |
| GIT.....  | 34 |
| CVS.....  | 34 |

## TEMA1

### Ejercicios Tema1

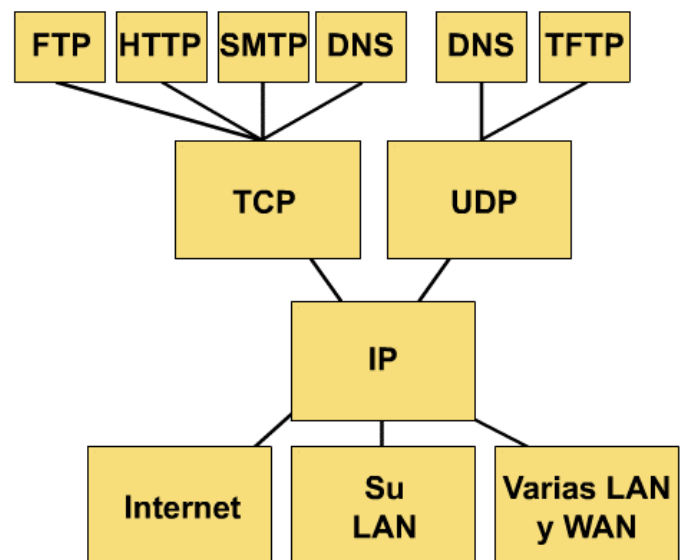
#### 1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS

**IP:** Protocolo de comunicación de **paquetes** a nivel de **red** no fiable que se basa en el **enrutamiento**.

**TCP:** Protocolo de comunicación a nivel de **red** que garantiza la **correcta comunicación** de paquetes entre las máquinas.

**HTTP:** Protocolo de comunicación **global** para internet. **Estandariza** el tránsito de información en la web a través de varios

#### Gráfico de protocolo: TCP/IP

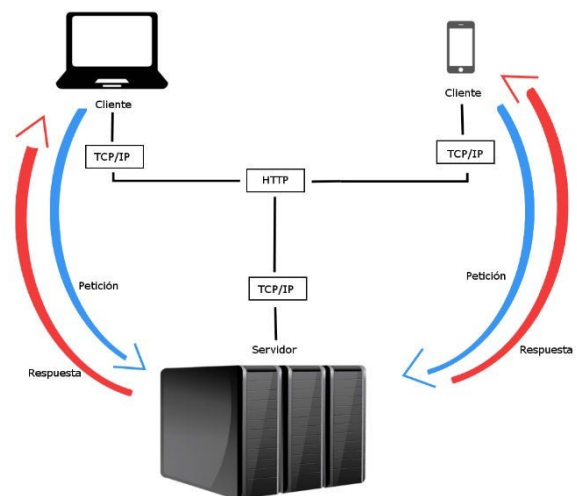


métodos. (Enlazar con el  
ejercicio de los métodos)

**HTTPS:** Protocolo de comunicación basado en **HTTP** caracterizado por evitar el robo de información al **encriptar** los paquetes intercambiados entre cliente y servidor.

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

El modelo de comunicación cliente servidor es aquél en el que un cliente hace una **petición** a un servidor, este la procesa y le envía una **respuesta**.



Esto es esencial en una **aplicación web** donde el cliente interacciona con una aplicación que se **ejecuta desde el servidor**.

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

*HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado*

#### **GET**

*El método GET solicita una representación de un recurso específico.  
Las peticiones que usan el método GET sólo deben recuperar datos.*

#### **HEAD**

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

#### **POST**

*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.*

#### **PUT**

*El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.*

#### **DELETE**

*El método DELETE borra un recurso en específico.*

**CONNECT**

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

**OPTIONS**

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

**TRACE**

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

**PATCH**

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

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

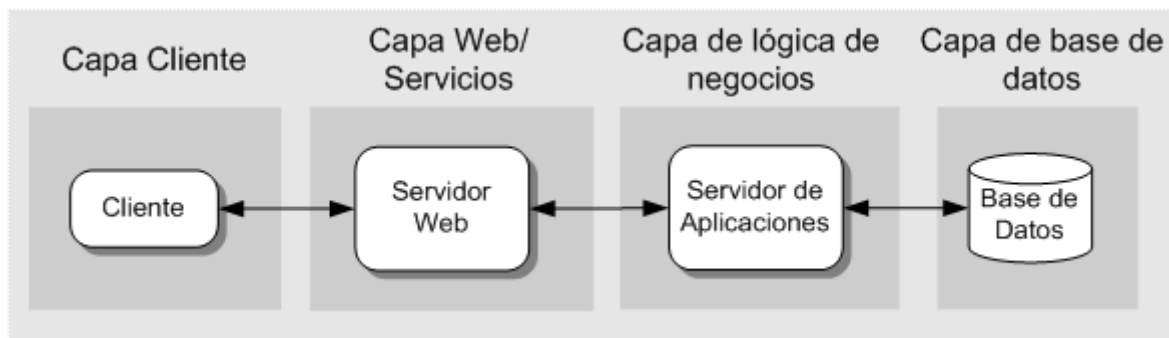
*La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada.*

*Por ejemplo, una aplicación web típica está compuesta por una capa de presentación (funcionalidad relacionada con la interfaz de usuario), una capa de negocios (procesamiento de reglas de negocios) y una capa de datos (funcionalidad relacionada con el acceso a datos).*

*El estilo de arquitectura basado en capas se identifica por las siguientes características:*

- Describe la descomposición de servicios de forma que la mayoría de la interacción ocurre solamente entre capas vecinas.*
- Las capas de una aplicación pueden residir en la misma máquina física (misma capa) o puede estar distribuido sobre diferentes computadores (n-capas).*
- Los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces muy bien definidas.*
- Este modelo ha sido descrito como una “pirámide invertida de re-uso” donde cada capa agrega responsabilidad y abstracción a la capa directamente sobre ella.*





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

*Cuando comenzamos en el campo del desarrollo web, pueden surgir algunos términos de los que no hemos oído hablar y que resultan fundamentales a la hora de comprender el procedimiento de construcción de un sitio en internet. Este es el caso del **Back-End**, **Front-End** y **Back Office**.*

#### ***BACK-END***

*Este término es utilizado para referirse al área lógica de toda página web. Nos referimos a la arquitectura interna del sitio que asegura que todos elementos desarrollen la función correcta. No está visible a ojos del usuario y no incluye ningún tipo de elemento gráfico.*

*Es la función de ingeniería a desarrollar por el programador, ya que se basa únicamente en el código interno de la página. Esta área es la*

encargada, además de la funcionalidad del sitio, de la seguridad y la optimización de los recursos.

En otras palabras, el Back-End es la parte o rama del desarrollo web encargada de que toda la lógica de una página funcione. Consiste en el conjunto de acciones que pasan dentro de una web, pero que no podemos ver. Un ejemplo de esto es la comunicación con el servidor.

Los encargados del desarrollo del Back End deben manejar eficientemente varios códigos de programación, en lenguajes como Java, PHP o .Net. Estos desarrolladores deben ser meticulosos, dado que un descuido en su trabajo se traduce en la caída de la página.

Los lenguajes más utilizados dentro del Back-End son:

- ASP.NET:** Es la plataforma de desarrollo web de Microsoft. Cuenta con las variantes de Web Forms y MVC.
- PHP:** Muy utilizado en algunos de los principales gestores de contenido como WordPress.
- Ruby:** También se utiliza su framework Ruby on rails.
- Python:** Se usa comúnmente con Django como framework.
- Node.js:** Más utilizado en los últimos tiempos, ya que utiliza un lenguaje similar al del lado del cliente (JavaScript).

*Las herramientas más utilizadas por los programadores en esta área del desarrollo web son los compiladores y editores de código. Es común que los mismos que desarrollan la parte del Back-End hagan lo propio con la del Front-End.*

## **FRONT-END**

*Ya mencionamos los principales lenguajes de programación que utilizan estos desarrolladores, los cuales forman parte de su perfil esencial como profesionales. Con ellos codifican el sitio web y los diseños de aplicaciones creados por los diseñadores web. Ahora, profundizaremos un poco más:*

### *HTML y CSS*

*El Hyper Text Markup Language (HTML) y el Cascading Style Sheets (CSS) son los bloques de construcción más básicos de la codificación web. Sin estos dos recursos, es imposible crear el diseño de un sitio web, y todo lo que obtendrás es texto sin formato en la pantalla. No es posible ni siquiera añadir imágenes a la página.*

*El conocimiento en codificación HTML y CSS te permitirá crear sitios básicos, por lo que es elemental tenerlo antes de emprender cualquier*

*carrera en desarrollo web. Algo positivo es que puedes obtener un sólido conocimiento práctico de cualquiera de estos lenguajes en unas pocas semanas.*

### *JavaScript*

*Un paso más adelante en el desarrollo tenemos a JavaScript, que permite incorporar más funcionalidad a los sitios web y crear muchas aplicaciones web básicas, utilizándolo en conjunto con HTML y CSS.*

*En su nivel más elemental, JavaScript es usado para crear y controlar cosas como mapas que se actualizan en tiempo real, películas interactivas y juegos online. Este es el lenguaje de programación más popular del mundo, por lo que sin importar los planes que tengas a nivel profesional, sería muy valioso aprenderlo.*

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

### ***Página web estática***

*Una página web estática es un sitio de Internet que muestra el mismo contenido para todos los usuarios, en vez de proporcionar contenido personalizado a la medida de cada usuario, y que no se actualiza a menudo. Las páginas web estáticas, que algunas veces se conocen como páginas planas, contrastan con las páginas web dinámicas que se*

*actualizan frecuentemente y proporcionan información personalizada para usuarios individuales.*

### ***Pagina web dinamica***

*Los sitios web dinámicos son aquellos que permiten crear aplicaciones dentro de la propia web, otorgando una mayor interactividad con el navegante. Aplicaciones dinámicas como encuestas y votaciones, noticiarios, foros de soporte, libros de visita, envío de e-mails inteligentes, reserva de productos, pedidos on-line, atención al cliente personalizada ... donde el administrador dispone la opción de gestionar dichos apartados, agregando, editando o eliminando contenidos.*

*La gran ventaja de los sitio web dinámicos es la versatilidad, apunta a la personalización de la información mostrada. Permite la*

*toma de decisiones y muestra selectiva de información en función de criterios deseados.*

*Es útil para mostrar una imagen de empresa más sólida y profesional ya que permite guardar información en la computadora del navegante y mostrar en forma personalizada la página web de la empresa así como la información adecuada al tipo de navegante.*

### *aplicación web – mashup*

*En [desarrollo web](#), una **mashup** es una forma de integración y reutilización. Ocurre cuando una aplicación web es usada o llamada desde otra aplicación, con el fin de reutilizar su contenido o funcionalidad. El uso en otra(s) fuente(s), para crear nuevos servicios simples, visualizado en una única interfaz gráfica diferente. Por ejemplo, se pueden combinar las direcciones y fotografías de las ramas de una biblioteca con un mapa de [Google Maps](#) para crear un mashup de mapa.*

*El término implica integración fácil y rápida, a menudo usando varias [API](#) abiertas y fuentes de datos para producir resultados enriquecidos, que no fueron necesariamente el motivo original de producir la fuente primaria de datos.*

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

*Una aplicación web que reside en uno o varios servidores normalmente está estructurada en tres capas: servidor web, lógica de la aplicación y un sistema gestor de bases de datos (en adelante SGBD). El primero de ellos mantiene la comunicación con el navegador del usuario que es quién proporciona la interfaz de acceso al servicio web. La lógica de la aplicación está formada por los scripts o programas que se ejecutan en*

*el servidor para atender las solicitudes del usuario, y el SGBD es el repositorio en el que se guardan los datos con los que trabaja el servicio. Las tres capas pueden estar montadas sobre una o varias máquinas dependiendo del volumen de carga que tenga que soportar el servicio, y todo corriendo sobre algún sistema operativo como GNU/Linux.*

## ***Servidor web***

*El servidor web se encarga la comunicación a través de la red con el navegador del usuario. Normalmente escucha en el puerto TCP 80 cuando se trata de una conexión HTTP, y en el TCP 443 cuando se trata de una HTTPS, aunque este parámetro es configurable.*

*Cuando recibe una petición del usuario puede atenderla de manera estática cuando contesta con un fichero que está en el sistema de archivos (la descarga de una imagen, un fichero css o html) o de manera dinámica cuando se ejecuta algún programa y se envía al cliente el resultado de dicha ejecución.*

*El servidor web evidentemente es capaz de atender muchas peticiones de forma concurrente pudiendo realizar diferentes funciones habituales entre las que se encuentran:*

- Registro de actividad y errores*
- Control de acceso basado en la dirección del cliente, contenido o usuario/contraseña*
- Virtual Hosts, para mantener diferentes webs (por ejemplo <http://elpuig.xeill.net> y <http://blog.elpuig.xeill.net>)*

- *Proxy, para reenviar las peticiones a otro servidor*
- *Reescritura de URLs*
- *Alias o mapeados de rutas*

*Aunque existen diferentes alternativas Apache es el servidor web más utilizado en Internet. Se trata de una aplicación libre robusta y modular con una impresionante colección de funciones, pudiendo servir tanto para contenido estático como para prácticamente cualquier contenido dinámico.*

*Otras alternativas pueden ser:*

### Cherokee

*Un servidor web más moderno que Apache con la pretensión de ser eficiente y fácil de configurar (tiene una interfaz web de administración). Está bajo la GPL.*

### NGINX

*Orientado a sitios con una gran carga de trabajo en los que es necesario un gran rendimiento.*

### Tomcat

*Contenedor de servlets para aplicaciones web escritas en Java.*

### Zope

*Servidor de aplicaciones escrito en Python.*



## **Capa de aplicaciones**

*En cualquier página web dinámica se ejecutan programas que determinan qué se debe responder a las solicitudes del usuario. Estos programas pueden estar escritos en una gran variedad de lenguajes de programación, una posible clasificación es: lenguajes compilados y lenguajes interpretados o scripts.*

*Los primeros obligan a compilar el código fuente antes de obtener un ejecutable, en líneas generales priman el rendimiento en tiempo de ejecución y cuando una aplicación se debe utilizar en diferentes plataformas debe ser compilada para cada una de ellas. Ejemplos típicos son los lenguajes C, C++ y Java.*

*Los lenguajes interpretados, o scripts, no necesitan ser compilados pues un programa llamado intérprete realiza las operaciones indicadas por el código fuente. En líneas generales permiten un desarrollo más rápido aunque no alcanzan en velocidad de ejecución a los lenguajes compilados. Ejemplos típicos son Bash, Perl, PHP, Python y Ruby.*

## **SGBD - Sistema Gestor de Bases de Datos**

*Una aplicación sencilla puede trabajar únicamente con ficheros, pero en cuanto el servicio deja de ser trivial aparece la necesidad de utilizar un SGBD que organice la información con la que se trabaja.*

*La historia de las bases de datos es larga, pero en cuanto a lo que una introducción a aplicaciones web se refiere se puede resumir en diferentes modelos:*

### SGBD relacional:

*Las más extendidas y con una larga vida por delante. Guardan la información en registros de tablas y mediante [SQL](#) se realizan consultas u operaciones para manipular los datos. Dependiendo de la herramienta utilizada hay grandes diferencias en cuanto a sus capacidades y administración, pero en general, para las aplicaciones web prima la velocidad sobre las características.*

*Ejemplos de uso: MySQL y PostgreSQL*

### SGBD orientado a objetos:

*En lugar de almacenar registros directamente guardan y recuperan objetos. De manera que cuando se utiliza un lenguaje de programación orientado a objetos nos ahorramos el trabajo de adaptar las entidades con las que trabajamos al modelo relacional.*

*Ejemplos de uso: DB4O y Zope Object Database*

### SGBD NoSQL:

*Aquí se agrupan un conjunto de técnicas diferentes que tienen en común que no cumplen con los [requisitos ACID](#) y/o no estructuran la información en tablas como las bases de datos relacionales. La*

*ventaja es que consiguen romper algunas de las limitaciones de los sistemas relacionales en cuanto a escalabilidad y rendimiento permitiendo crear ingentes bases de datos distribuidas.*

**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 servidor se ejecutan programas como cualquier IDE de desarrollo tal como netbeans eclipse o visual studio.*

*en el lado del servidor se ejecuta con cualquier navegador web de los disponibles en el mercado chrome opera firefox etc.*

*los lenguajes que se utilizan en el lado del servidor son php python java ruby y un largo etc.*

*los lenguajes que se utilizan en el lado del cliente son practicamente estandars, javascript como principal y luego los lenguajes de marcas HTML y CSS, también se puede programar con .net de microsoft y la tecnología java.*

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

*Los lenguajes del lado del servidor más utilizados en la actualidad son: ASP, PERLyPHP.*

### *ASP.NET*

*Un lenguaje comercializado por Microsoft y ampliamente utilizado por programadores de todo el mundo para la creación y el desarrollo de sitios web dinámicos. 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.*

*Una de las principales ventajas que ofrece ASP.NET es que permite el control de usuario personalizado y, además, ofrece una división entre la capa de diseño y el código. En cuanto a las desventajas, la más destacada es que consume una gran cantidad de recursos.*

## PERL

*PERL es un lenguaje de programación basado en una serie de herramientas UNIX. Su principal punto fuerte es que resulta muy útil en el procesamiento tanto de textos como de archivos. Además, 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 “a medida”.*

*Claro que PERL también presenta alguna desventaja; resulta algo lento para determinadas aplicaciones, como por ejemplo la programación a bajo nivel.*

## PHP

*PHP es el acrónimo de **Hipertext Preprocesor**. Se trata de uno de los lenguajes de programación del lado del servidor más utilizados actualmente. Es gratuito e independiente y, además, muy rápido. Se creó a mediados de los 90; desde entonces, debido a que es un **lenguaje de código abierto**, ha recibido cientos de contribuciones por parte de desarrolladores de todo el mundo.*

*Una de las principales ventajas que ofrece es que es muy **fácil de***

*aprender; además, es multiplataforma. Incorpora una gran selección de funciones y ofrece una gran facilidad para establecer conexión con todo tipo de bases de datos, como Oracle o MySQL.*

*Respecto a las desventajas, en PHP todo el trabajo recae sobre el servidor; no delega ni una sola tarea sobre el lado del cliente.*



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

*XAMPP es el acrónimo de Cualquier Plataforma (X), Apache (A), MySQL (M), PHP (P) y Perl (P). Es una distribución de Apache sencilla y ligera que facilita enormemente a los desarrolladores crear un servidor web local para realizar pruebas. Todo lo que necesita para configurar un servidor web – la aplicación servidor (Apache), la base de datos (MySQL), y un lenguaje de script (PHP) – está incluido en un único fichero extraíble. XAMPP es también multi-plataforma, lo que significa que funciona bien tanto en Linux, como Mac o Windows. Dado que la mayoría de servidores web actuales usan los mismos componentes que XAMPP, la transición desde el servidor de prueba local al servidor de producción es extremadamente fácil también.*

*Incluye*

*XAMPP tiene cuatro componentes principales. Éstos son:*

*1·Apache: Apache es la aplicación de servidor web que procesa y entrega el contenido web a un ordenador. Apache es el servidor web más popular en internet, haciendo funcionar casi el 54% de todas las páginas web.*

*2·MySQL: Cada aplicación web, ya sea simple o compleja, requiere una base de datos para almacenar los datos que recoge. MySQL,*

*que es de código abierto, es el sistema gestor de base de datos más popular. Alimenta desde websites de aficionados hasta plataformas profesionales como WordPress.*

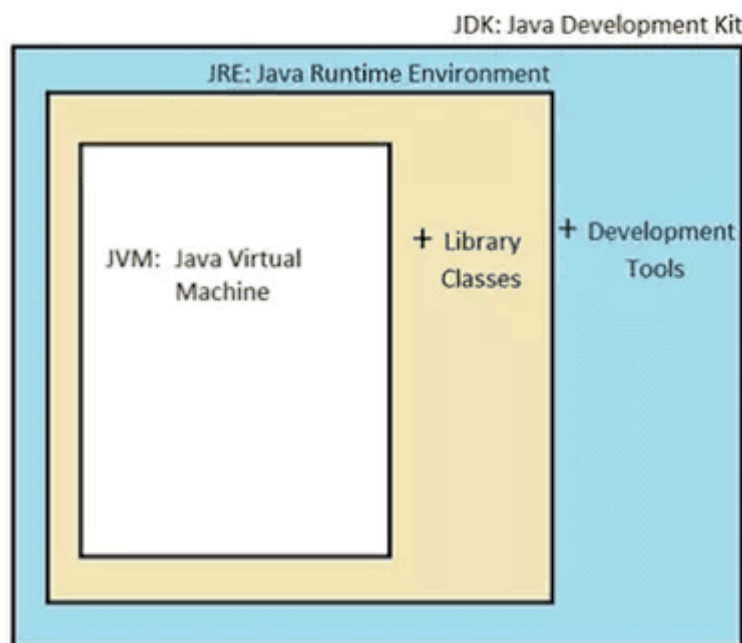
*3·PHP: son las siglas de Preprocesador de Hipertexto. Es un lenguaje de script del lado del servidor que hace funcionar algunas de las páginas web más famosas del mundo, incluyendo WordPress y Facebook. Es de código abierto, relativamente fácil de aprender, y funciona perfectamente con MySQL, lo que lo ha hecho una opción muy popular para los desarrolladores web.*

*4·Perl: Perl es un lenguaje de programación dinámico y de alto nivel que se usa ampliamente en la programación de redes, la administración de sistemas, etc. Aunque menos popular para el desarrollo web, Perl tiene un montón de aplicaciones específicas.*



*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.*

*Para entender la diferencia entre estos dos consideremos el siguiente diagrama:*



JDK = JRE + Development Tools

JRE = JVM + Library Classes

• **JDK - Java Development Kit** (en resumen JDK): es un kit que proporciona el entorno para desarrollar y ejecutar el programa Java. JDK es un kit (o paquete) que incluye dos cosas:

1. **Herramientas de desarrollo** (para proporcionar un entorno para desarrollar sus programas de Java)

2. **JRE** (para ejecutar su programa java).

**JRE- Java Runtime Environment** (JRE): es un paquete de instalación que proporciona un entorno para ejecutar (no desarrollar) el programa Java (o la aplicación) en su máquina. JRE solo lo utilizan aquellos que solo desean ejecutar los Programas Java, es decir, los usuarios finales de su sistema.

• **JVM- Java Virtual Machine** (JVM): es una parte muy importante de JDK y JRE porque está contenida o incorporada en ambos. Cualquier programa Java que ejecute utilizando JRE o JDK entra en la JVM y la JVM es responsable de ejecutar el programa Java línea por línea, por lo que también se lo conoce como intérprete.

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

| Desarrollador web  |       | Desarrollador móvil |       | SRE/DevOps         |       |
|--------------------|-------|---------------------|-------|--------------------|-------|
| Visual Studio Code | 55.6% | Android Studio      | 54.2% | Visual Studio Code | 55.2% |
| Visual Studio      | 32.5% | Visual Studio Code  | 53.8% | Vim                | 43.7% |
| Notepad++          | 30.4% | Xcode               | 34.5% | IntelliJ           | 29.4% |
| IntelliJ           | 27.3% | Visual Studio       | 31.3% | Visual Studio      | 28.7% |
| Vim                | 25.9% | IntelliJ            | 31.1% | Notepad++          | 28.2% |
| Sublime Text       | 24.0% | Notepad++           | 27.8% | Sublime Text       | 23.0% |
| Android Studio     | 15.1% | Sublime Text        | 26.5% | PyCharm            | 16.3% |
| Eclipse            | 14.2% | Vim                 | 18.6% | Atom               | 14.1% |
| Atom               | 12.7% | Atom                | 14.9% | Android Studio     | 13.1% |
| PyCharm            | 12.1% | Eclipse             | 14.4% | Eclipse            | 12.2% |
| PHPStorm           | 9.3%  | PyCharm             | 10.7% | PHPStorm           | 10.1% |
| Xcode              | 8.1%  | PHPStorm            | 8.8%  | IPython / Jupyter  | 9.9%  |
| IPython / Jupyter  | 7.2%  | NetBeans            | 8.1%  | Xcode              | 8.4%  |
| NetBeans           | 5.8%  | IPython / Jupyter   | 5.6%  | Emacs              | 6.5%  |
| Emacs              | 4.1%  | Emacs               | 2.8%  | NetBeans           | 4.5%  |
| RStudio            | 1.8%  | TextMate            | 1.8%  | RubyMine           | 2.6%  |
| RubyMine           | 1.5%  | RStudio             | 1.6%  | RStudio            | 2.1%  |
| TextMate           | 0.9%  | RubyMine            | 1.6%  | TextMate           | 1.5%  |
| Coda               | 0.5%  | Coda                | 1.1%  | Coda               | 1.0%  |
| Komodo             | 0.3%  | Zend                | 0.7%  | Komodo             | 0.8%  |
| Zend               | 0.3%  | Komodo              | 0.7%  | Zend               | 0.7%  |
| Light Table        | 0.2%  | Light Table         | 0.4%  | Light Table        | 0.6%  |

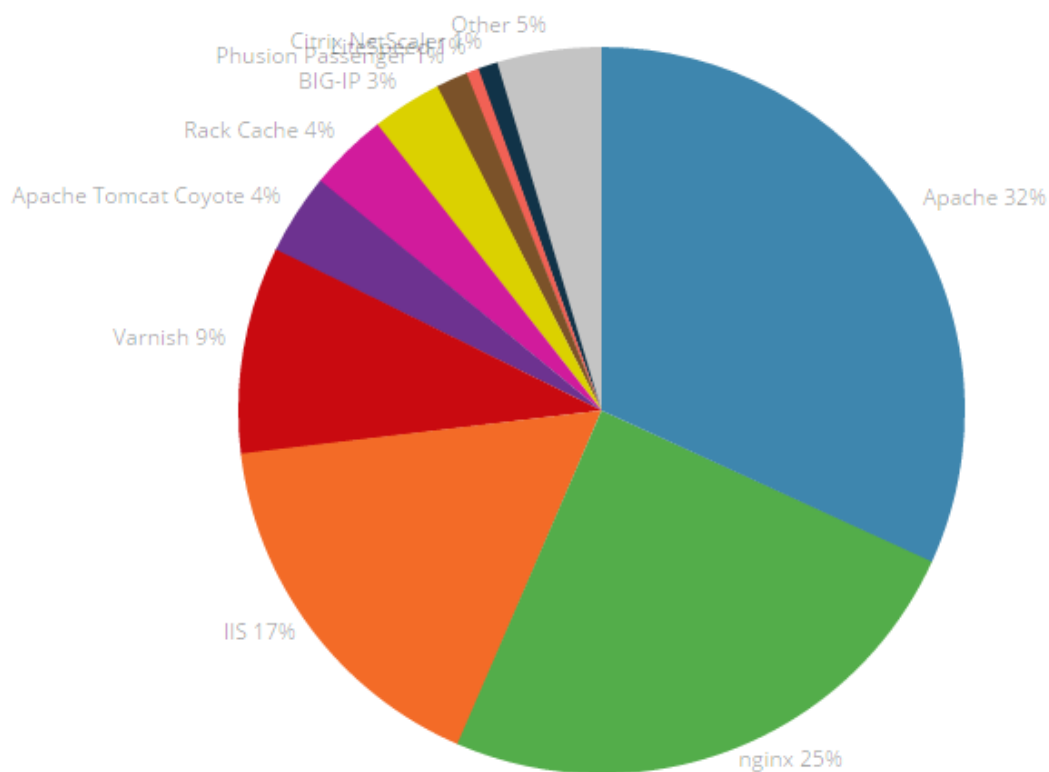
Entornos de desarrollo más populares

Stack Overflow

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

## Web Server Usage Statistics

Statistics for websites using Web Server technologies



## ***Servidor Apache***

*El servidor HTTP Apache es el más antiguo y utilizado hasta la actualidad. Si bien su cuota de mercado ha ido descendiendo en los últimos años, aún es usado por más del 40 % de las páginas web, el doble que nginx, el segundo servidor más popular.*

*La Apache Foundation lo desarrolló a mediados de los años 90 y se puede instalar en plataformas Linux, Windows y Mac OSX, entre otras.*

*La gran ventaja de Apache es que es de código abierto, por lo que se puede personalizar y ampliar por módulos sin afectar la base del servidor. Además, es uno de los más estables, lo que lo hace una opción muy rentable, y la mayor parte de los fallos de seguridad que se han descubierto en el pasado solo podían ser aprovechados por usuarios locales.*

*Entre las páginas web más conocidas que utilizan un servidor Apache están Adobe, PayPal y Apple.*

## ***Servidores nginx***

*La característica más destacada de nginx (leído como «Engine-X») es que se basa en eventos, a diferencia de Apache —que usa un sistema de hilos— y puede gestionar miles de conexiones simultáneas —algo ideal para los servidores proxy durante situaciones de emergencia— empleando muchos menos recursos que otros modelos. Además, como también es open-source, nos permite añadir módulos fácilmente.*

*Dropbox, Netflix y WordPress, entre otros muchos, hacen uso de servidores nginx.*

## ***Servidor de Google***

*También conocido como GWS, de Google Web Server, es un servidor a medida que Google usa en sus infraestructuras y que alberga aproximadamente un 10% de todas las páginas web activas del mundo.*

## ***Servidor IIS de Microsoft***

*El servidor IIS (de Internet Information Services) fue desarrollado por Microsoft y por tanto no es de código abierto, motivo por el cual añadir módulos no resulta nada fácil. Como contrapartida, IIS dispone de varios niveles de autenticación, por lo que es un servidor muy seguro. Incluye FTP, SMTP, NNTP y HTTP/HTTPS.*

## ***14. Apache HTTP vs Apache Tomcat***

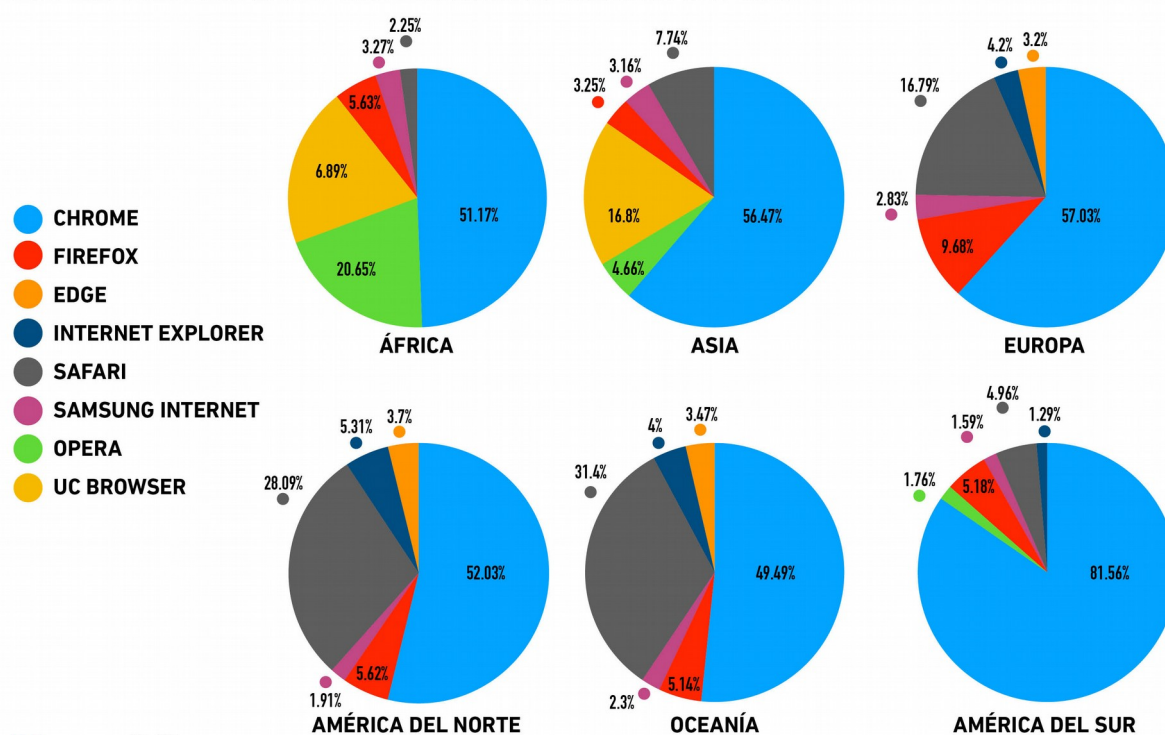
*En palabras simples, Apache es un servidor web destinado a servir páginas web estáticas. El ejemplo incluye páginas HTML simples (puede crear una página HTML simple y Apache lo servirá por usted. Intente usar XAMPP o el servidor WAMP para verlo usted mismo en máquinas Windows). Los sitios de Facebook, Google o Wordpress son algunos de los ejemplos.*

*Apache Tomcat, por otro lado, es un servidor de aplicaciones destinado a servir aplicaciones Java (Servlets, JSP, etc.). También puede servir*

*páginas web a través de Tomcat, pero es menos eficiente en comparación con Apache. IRCTC es uno de esos sitios web.*

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

### USO DE LOS NAVEGADORES EN EL MERCADO POR ESPACIO GEOGRÁFICO



**Merca2.0**  
mercado tecnología publicidad medios

Fuente: Departamento de investigación Merca2.0/ Statcounter

## Google Chrome

*Cuenta con más de 750 millones de usuarios, y dependiendo de la fuente de medición global, puede ser considerado el navegador más usado de la Web variando hasta el segundo puesto, algunas veces logrando la popularidad mundial en la primera posición.*

*Aquí tenemos un enlace al la [wikipedia sobre chrome](#)*

## Mozilla firefox

*Mozilla Firefox (llamado simplemente Firefox) es un navegador libre y seguramente el segundo mas utilizado del mundo. Aquí tenemos un enlace a la [wikipedia sobre firefox](#)*



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

### *phpDocumentor*

*Existen algunas herramientas que permiten generar documentación de forma automática a partir del código fuente. Javadoc es la herramienta estándar en Java. Para PHP una de las herramientas más utilizadas es [phpDocumentor](#) y en este enlace podremos ver toda su*

### *PhpDoc*

*Es una adaptación de javadoc para php que define un estándar oficial para comentar código php, el enlace siguiente a la [wikipedia de PhpDoc](#) nos mostrara todo lo que hace esta herramienta.*

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

### ***GIT***

*Git, es un software de control de versiones diseñado por Linus Torvalds. La pregunta es ¿qué es control de versiones? Pues bien, se define como control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo es decir a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración, y para los que aún no les queda claro del todo, control de versiones es lo que se hace al momento de estar desarrollando un software o una página web. Exactamente es eso que haces cuando subes y actualizas tu código en la nube, o le añades alguna parte o simplemente le editas cosas que no funcionan como deberían o al menos no como tú esperarías.*

### ***CVS***

*Concurrent Versions System o simplemente 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, en un único archivo para cada fichero correspondiente) 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**.*

*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 con comandos GNU.*