



DAWS **TEMA 1**

Miguel Ángel Aranda García



Contenido

1º Protocolos de comunicación:	pag. 1
2º Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.	pag. 4
3º Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.	pag. 5
4º Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad década capa.	pag. 7
5º Modelo de división funcional front-end / back-end para aplicaciones web.	pag. 8
6º PÁGINA WEB ESTÁTICA – PÁGINA WEB DINÁMICA – APLICACIÓN WEB MASHUP.	pag. 9
7º Componentes de una aplicación web.	pag. 10
8º Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor – lenguajes de programación utilizados en cada caso.	pag. 11
9º Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).	pag. 14
10º Características y posibilidades de desarrollo de una plataforma XAMPP.	pag. 17
11º 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.	pag. 18
12º IDE más utilizados (características y grado de implantación actual).	pag. 19
13º Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).	pag. 23
14º Apache HTTP vs Apache Tomcat.	pag. 27



15º Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).
pag. 27

El Protocolo de Internet (en inglés **Internet protocol o IP**) es un protocolo de comunicación de datos digitales clasificado funcionalmente en la **capa de red** según el modelo internacional OSI.

Su función principal es el uso bidireccional en origen o destino de comunicación para transmitir paquetes conmutados a través de distintas redes físicas previamente enlazadas según la norma OSI de enlace de datos.

Una **dirección IP** es un número que identifica de manera lógica y jerárquicamente a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo de Internet (Internet Protocol) pudiendo ser esta IPv4 o IPv6.

TCP:

Protocolo de control de transmisión (en inglés **Transmission Control Protocol o TCP**) es uno de los protocolos fundamentales en Internet.

Muchos programas dentro de una red de datos compuesta por redes de computadoras, pueden usar TCP para crear "conexiones" entre sí a través de las cuales puede enviarse un flujo de datos. 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 (navegadores, intercambio de ficheros, clientes FTP, etc.) Y protocolos de aplicación HTTP, SMTP, SSH y FTP.

Funciones de TCP

En la pila de protocolos TCP/IP, TCP es la capa intermedia entre el protocolo de red (IP) y la aplicación. Muchas veces las aplicaciones necesitan que la comunicación a través de la red sea confiable.

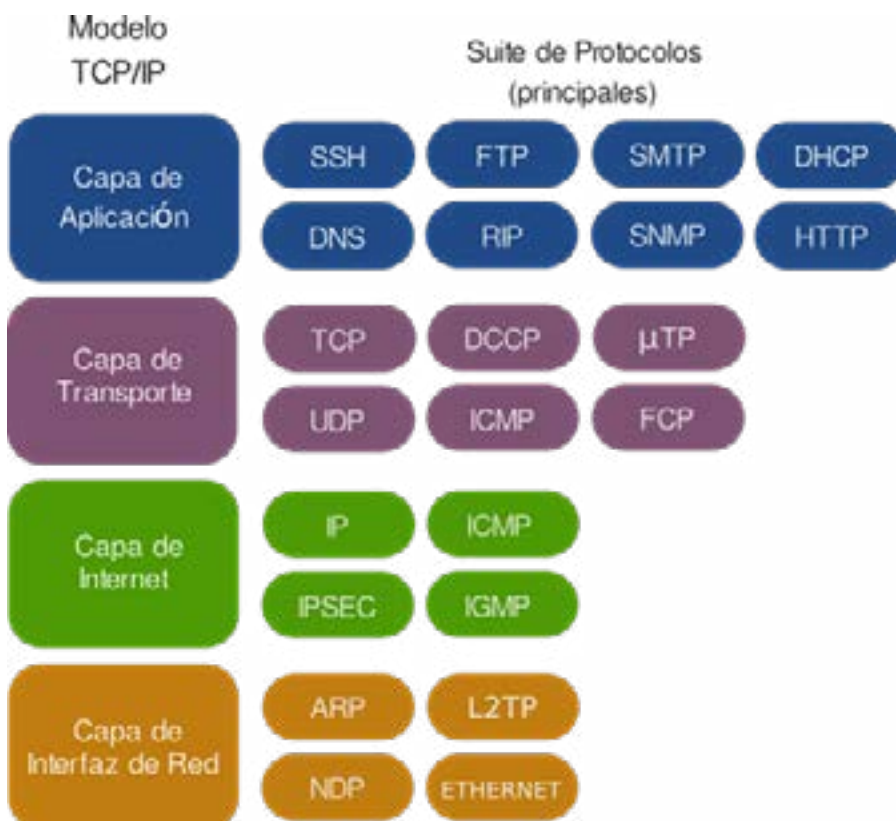
Características del TCP:

- Permite colocar los segmentos nuevamente en orden cuando vienen del protocolo IP.
- Permite el monitoreo del flujo de los datos y así evitar la saturación de la red.
- Permite que los datos se formen en segmentos de longitud variada para "entregarlos" al protocolo IP.
- Permite multiplexar los datos, es decir, que la información que viene de diferentes fuentes (por ejemplo, aplicaciones) en la misma línea pueda circular simultáneamente.
- Por último, permite comenzar y finalizar la comunicación amablemente.

1º Protocolos de comunicación:

Protocolo **TCP/IP**:

Es el protocolo más usado del mundo es un estándar de protocolos de comunicaciones.



El protocolo wifi es 802.11.

Una red se identifica por la capa de transporte por el puerto, con la capa de Internet por IP.

Socket es la unión entre dos puntos de Internet siendo la unión de 2 protocolos TCP/IP.



HTTP:

El Protocolo de transferencia de hipertexto (en inglés, **Hypertext Transfer Protocol**, abreviado **HTTP**) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web.

El puerto estándar para este protocolo es el 80.

Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. El cliente (se le suele llamar "agente de usuario", en inglés user agent) realiza una petición enviando un mensaje, con cierto formato al servidor. El servidor (se le suele llamar un servidor web) le envía un mensaje de respuesta.

Los mensajes HTTP son en texto plano lo que lo hace más legible y fácil de depurar. Esto tiene el inconveniente de hacer los mensajes más largos.

HTTP define una serie predefinida de métodos de petición (algunas veces referido como "verbos") que pueden utilizarse. El protocolo tiene flexibilidad para ir añadiendo nuevos métodos y para así añadir nuevas funcionalidades. El número de métodos de petición se ha ido aumentando según se avanzaba en las versiones.



HTTPS:

El Protocolo seguro de transferencia de hipertexto (en inglés, **Hypertext Transfer Protocol Secure** o **HTTPS**) 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 cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente). El puerto estándar para este protocolo es el 443.

Las diferencias entre https y http son muy básicamente a nivel de seguridad dado que https está encriptado y http no lo está además usa el puerto 443 https y el puerto 80 el http. La forma de usar la url siendo https:// y http:// para cada una de ellas y antes de pasarle la información al protocolo tcp el https encripta la información.

Es lo mismo decir servidor https que servidor web que es lo mismo decir cliente web que cliente https.

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

La **arquitectura cliente-servidor** es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

Tiene partes claramente diferenciadas, por un lado la parte del servidor y por otro la parte de cliente o grupo de clientes donde lo habitual es que un servidor sea una máquina bastante potente con un hardware y software específico que actúa de depósito de datos y funcione como un sistema gestor de base de datos o aplicaciones.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

Una aplicación web es un cliente web que pide una páginas web con o sin código embebido a un servidor web.

HTML es un **lenguaje de marcas** que se visualiza en el cliente web y se almacena, procesa en el servidor web.

Dependiendo del lenguaje con el que está programado el programa el servidor enviara el código o lo interpretara antes de enviarlo.



PHP lado del servidor y **JavaScript** lado del Cliente.

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

Todos son solicitudes clientes/servidor por cada tipo petición hay una consecuencia, el cliente es el que realiza las peticiones y el servidor es el que da respuesta.

El método más usado al navegar es get, pero al usar webservices get es el método que menos se usa.

HTTP y HTTPS es el dialogo, el cliente es el que siempre inicia el dialogo.



GET:

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

HEAD:

Pide una respuesta idéntica a la que correspondería a una petición GET, pero en la respuesta no se devuelve el cuerpo. Esto es útil para poder recuperar los metadatos de los encabezados de respuesta, sin tener que transportar todo el contenido. Siendo usado para operaciones más básicas.

POST:

Envía datos para que sean procesados por el recurso identificado en la URI de la línea petición. Los datos se incluirán en el cuerpo de la petición. A nivel semántico está orientado a crear un nuevo recurso, cuya naturaleza vendrá especificada por la cabecera Content-Type.

Ejemplos:

Para datos formularios codificados como una URL.

Para bloques a subir, ej. Ficheros.

Además de los anteriores, no hay un estándar obligatorio y también podría ser otros como text/plain, application/json, application/octet-stream.

PUT:

Envía datos al servidor, pero a diferencia del método POST la URI de la línea de petición no hace referencia al recurso que los procesará, sino que identifica a los propios datos.

Otra diferencia con POST es semántica mientras que POST está orientado a la creación de nuevos contenidos, PUT está más orientado a la actualización de los mismos (aunque también podría crearlos).

DELETE:

Borra el recurso especificado.

TRACE:

Este método solicita al servidor que introduzca en la respuesta todos los datos que reciba en el mensaje de petición. Se utiliza con fines de depuración y diagnóstico ya que el cliente puede ver lo que llega al servidor y de esta forma ver todo lo que añaden al mensaje los servidores intermedios

OPTIONS:

Devuelve los métodos HTTP que el servidor soporta para un URL específico. Esto puede ser utilizado para comprobar la funcionalidad de un servidor web mediante petición en lugar de un recurso específico.

CONNECT:

Se utiliza para saber si se tiene acceso a un host, no necesariamente la petición llega al servidor, este método se utiliza principalmente para saber si un proxy nos da acceso a un host bajo condiciones especiales, como por ejemplo "corrientes" de datos bidireccionales encriptados (como lo requiere SSL).

PATCH:

Su función es la misma que PUT, el cual sobrescribe completamente un recurso. Se utiliza para actualizar, de manera parcial una o varias partes. Está orientado también para el uso con proxy. Hablar como se ven sobre la url como se construye el localizador de recursos universal.

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

El modelo de desarrollo de aplicaciones multicapa:

Es un modelo de desarrollo software en el que el objetivo es la separación de las partes que componen un sistema software o también una arquitectura cliente-servidor: lógica de negocios, capa de presentación y capa de datos. De esta forma, es sencillo crear diferentes interfaces sobre un mismo sistema sin necesidad de cambio alguno en la capa de datos o lógica.

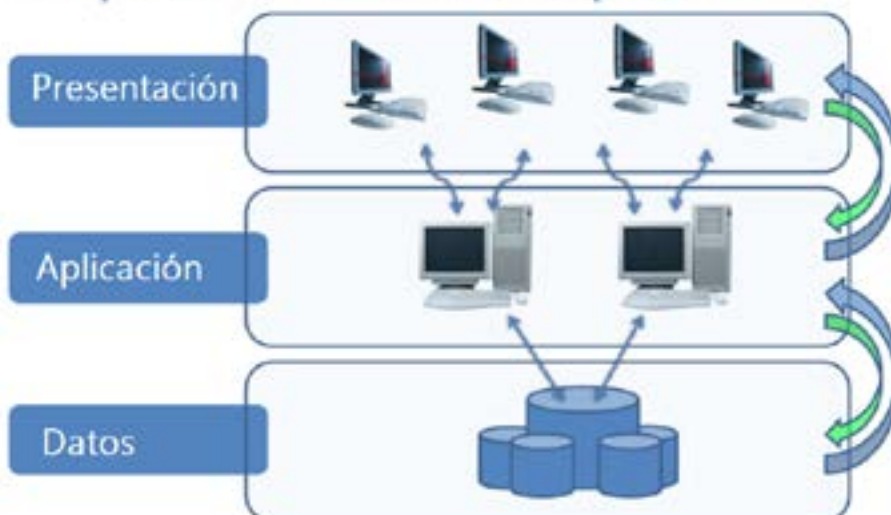
Presentación:

En esta capa se crea la interfaz del usuario. Su función es pasarle las acciones que realice el usuario a la capa de negocio y realiza un filtrado previo. Es conocida como interfaz gráfica.

Negocio:

En esta capa se gestiona la lógica de la aplicación. Es donde se dice que se hace con los datos. Estará conectada con la capa de persistencia para poder realizar sus funciones.

Arquitectura de 3 capas



Persistencia:

Esta capa se encarga de guardar los datos. Será donde se gestione todo lo relacionado a la base de datos y a la manipulación de la base de datos.

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

Front-end:

Es la parte de la aplicación web a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

HTML, CSS y JavaScript son los lenguajes principales del Front-end.

Back-end:

Es la capa de acceso a la aplicación web, contiene la administración de la web que maneja dichos datos.

Python, PHP, Ruby, C# y Java son los lenguajes principales del Back-end.



6º 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.

Estas páginas a menudo se usan para sitios orientados hacia la prestación de ofertas o información estable para los visitantes.

Página web dinámica:

Son aquellas web que permiten crear aplicaciones dentro de sí mismas, otorgando una mayor interactividad con el navegante. Aplicaciones dinámicas como encuestas y votaciones, noticieros, 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.

Persistencia de información.

Adaptación de contenido.

Control de acceso.

Webservices.

Utiliza muchas tecnologías.

Lenguas de programación gestor base de datos servidores etc.

Aplicación web:

Son herramientas que los usuarios pueden utilizar para acceder a un servidor web a través de internet o de una intranet mediante un navegador de ahí que reciban el nombre de App web. En otras palabras, es un programa que se codifica en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador.

Mashup:

Una **Mashup** o **aplicación web híbrida** es una aplicación que combina servicios de varias páginas web en una sola para ofrecer un nuevo servicio. Es habitual que la integración de datos se obtenga desde APIs abiertas. La libertad para integrar todas estas fuentes supone una oportunidad para desarrollar nuevos servicios en muy poco tiempo, al basarse en funcionalidades ya existentes.

Típicamente los mashups se basan en utilizar contenido de terceros a través de fuentes o interfaces públicas. Entre las fuentes más utilizadas para crear mashups destacan Google Maps, Amazon, Flickr, Youtube, Ebay y Yahoo. La facilidad con la que estas plataformas permiten integrar su contenido ha propiciado un gran éxito para las aplicaciones web híbridas.

7º Componentes de una aplicación web.

Servidor web:

El servidor web se encarga la comunicación a través de la red con el navegador del usuario.

Interpretes:

Son las aplicaciones traductor de código o interpretes (de Php, Asp, Perl, Python, Ruby).

Algunas de estas aplicaciones pueden ser Apache, Nginx, internet information services (ISS), Cherokee, Tomcat .

Sistema gestor de base de datos:

Una aplicación sencilla puede trabajar con ficheros que organice la información con la que se trabaja.

Lenguaje de programación:

Lenguajes de scripting:
PHP, JSP, Perl, Python.

Lenguajes compilados a código nativo:

Aplicaciones CGI(C, C++).

Lenguajes compilados a código intermedio:

Java EE (Servlets, JSP, EJB). ASP.net.

Lenguaje de marcas:

Lenguaje de marcas es una forma de codificar un documento, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

El lenguaje más extendido es el HTML (HyperText Markup Language, lenguaje de marcado de hipertexto)

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

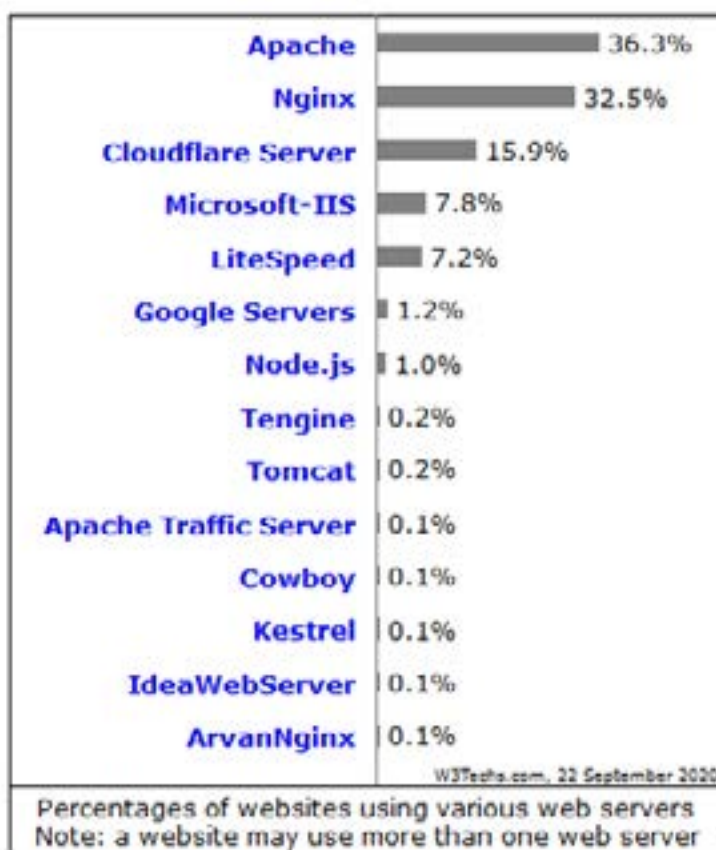
Los **programas** ejecutados en el **lado del cliente** son los navegadores (Firefox, Chrome, Safari, Opera, I.E...).

Los navegadores mas usados en la ultima década a nivel mundial son:



Los **programas** ejecutados en el **lado del servidor** son normalmente los servidores web (Apache, Nginx, internet information services(IIS), Cherokee, Tomcat).

Apache is used by 36.3% of all the websites whose web server we know.



Lenguajes de programación utilizados en el **lado del cliente**: HTML, JAVASCRIPT , CSS.

Lenguajes de programación utilizados en el **lado del servidor**: CGI, PERL, ASP, PHP, JSP.

Siendo estos los más usados a nivel mundial en ambos lados:

Rank	Change	Language	Share	Trend
1		Python	31.66 %	+2.9 %
2		Java	16.4 %	-3.1 %
3		Javascript	8.38 %	+0.3 %
4		C#	6.5 %	-0.8 %
5		PHP	5.65 %	-0.5 %
6		C/C++	5.8 %	+0.0 %
7		R	4.08 %	+0.3 %
8		Objective-C	2.79 %	+0.2 %
9		Swift	2.35 %	-0.1 %
10		TypeScript	1.92 %	+0.1 %
11		Matlab	1.65 %	-0.1 %
12		Kotlin	1.61 %	+0.1 %
13	↑↑	Go	1.44 %	+0.3 %
14	↓	Ruby	1.22 %	-0.2 %
15	↓	VBA	1.16 %	-0.2 %
16	↑↑	Rust	1.01 %	+0.3 %
17	↓	Scala	0.93 %	-0.2 %
18	↓	Visual Basic	0.79 %	-0.2 %
19	↑↑↑↑	Lua	0.68 %	+0.2 %
20	↑↑↑↑	Dart	0.57 %	+0.2 %
21	↑	Ada	0.52 %	+0.2 %
22	↓↓↓	Perl	0.51 %	-0.0 %
23	↓↓↓	Groovy	0.46 %	-0.0 %
24	↑↑↑	Julia	0.46 %	+0.2 %
25		Cobol	0.43 %	+0.1 %
26	↓↓↓↓↓	Abap	0.41 %	-0.1 %
27	↑	Delphi	0.31 %	+0.0 %
28	↓↓	Haskell	0.29 %	

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

ASP.NET:

Es un entorno para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, es la tecnología sucesora de la tecnología Active Server Pages (ASP).

Características:

ASP.NET no sólo funciona sobre el servidor de Microsoft IIS, también lo hace sobre Apache.

Ventajas:

- Completamente orientado a objetos.
- Controles de usuario y personalizados.
- División entre la capa de aplicación o diseño y el código.
- Facilita el mantenimiento de grandes aplicaciones.
- Incremento de velocidad de respuesta del servidor.
- Mayor velocidad.
- Mayor seguridad.

Desventajas:

- Mayor consumo de recursos.
- Tecnología propietaria.
- Hospedaje de sitios web costosos.

JSP:

(Java Server Pages) Es un lenguaje para la creación de sitios web dinámicos. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma.

Ventajas:

- Ejecución rápida del servlets.
- Crear páginas del lado del servidor.
- Multiplataforma.
- Código bien estructurado.
- Integridad con los módulos de Java.
- La parte dinámica está escrita en Java.

Desventajas:

- Complejidad de aprendizaje.

PERL:

(Practical Extraction and Report Language) No establece ninguna filosofía de programación concreta. No se puede decir que sea orientado a objetos, modular o estructurado aunque soporta directamente todos estos paradigmas; su punto fuerte son las labores de procesamiento de textos y archivos. Lenguaje de programación basado en scripts portable a casi cualquier plataforma. Es muy utilizado para escribir CGI's (Common Gateway Interface). Uno de sus elementos más potentes son las expresiones regulares, que a partir de su versión en Perl han sido adoptadas por otros lenguajes y plataformas como .NET o Javascript.

Ventajas:

- Se pueden juntar varios programas de una forma sencilla para alcanzar una meta determinada.
- Es relativamente rápido para un lenguaje tipo script.
- Está disponible en múltiples plataformas y sistemas operativos (UNIX, Linux y Windows).
- El desarrollo de aplicaciones es muy rápido.
- Hay una colección enorme de módulos que pueden ser incorporados a cualquier Script de Perl.
- Perl es gratuito. Es Software Libre.
- Le otorga al programador mucha libertad para que haga el programa como quiera.

Desventajas:

- Es lento para algunas aplicaciones.
- La libertad que se le otorga al programador puede significar que el resultado sea un programa ilegible.
- Utiliza muchos recursos de computó.

PHP:

Es de código abierto muy usado en el desarrollo web, ya que puede incrustarse en HTML. Genera HTML y se lo envía al cliente.

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de base de datos: MySQL, PostgreSQL, Oracle, MS ,SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- Todo el trabajo lo realiza el servidor y no delega al cliente.
- Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
 - La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
 - Dificulta la modulación. - Dificulta la organización por capas de la aplicación.

Python:

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Ventajas:

- Simplificado y rápido: este lenguaje simplifica mucho la programación.
- Elegante y es un lenguaje tan flexible: no hay que preocuparse tanto por los detalles.
- Programación sana y productiva: es direccionado a las reglas perfectas, le hace como dependiente de mejorar, cumplir las reglas, el uso de las líneas, de variables.
- Ordenado y limpio: siendo muy legible cuando cualquier otro programador necesita leer y trabajar sobre el programa escrito .
- Portable: se puede usar en cualquier plataforma.
- Comunidad: la misma comunidad de Python cuida el lenguaje y casi todas las actualizaciones se hacen de manera democrática.

Desventajas:

- Curva de aprendizaje: La "curva de aprendizaje cuando ya estás en la parte web no es tan sencilla".
- Hosting: La mayoría de los servidores no tienen soporte a Python, y si lo soportan, la configuración es un poco difícil.
- Librerías incluidas Algunas librerías que trae por defecto no son del gusto de amplio de la comunidad, y optan a usar librerías de terceros.

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

Es un servidor de plataforma libre, es un software que integra en una sola aplicación, un servidor web Apache, intérpretes de lenguaje de scripts PHP, un servidor de base de datos MySQL, un servidor de FTP FileZilla, base de datos escrito en PHP, MySQL, entre otros módulos convirtiendo nuestro equipo en un componente perfecto para el desarrollo de páginas web, o para la ejecución de cualquier software.

Ventajas:

- Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris)
- Es multiplataforma.
- Es sencillo de instalar y de configurar. - Es Software libre.

Deventajas:

- No soporta MySQL desde la consola.
- No se pueden actualizar individualmente las versiones de los programas que instala toca reinstalarlo entero para poder actualizar.
- Dificultad para configurar aplicaciones de terceros.

11º 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.

Máquina virtual de java JVM:

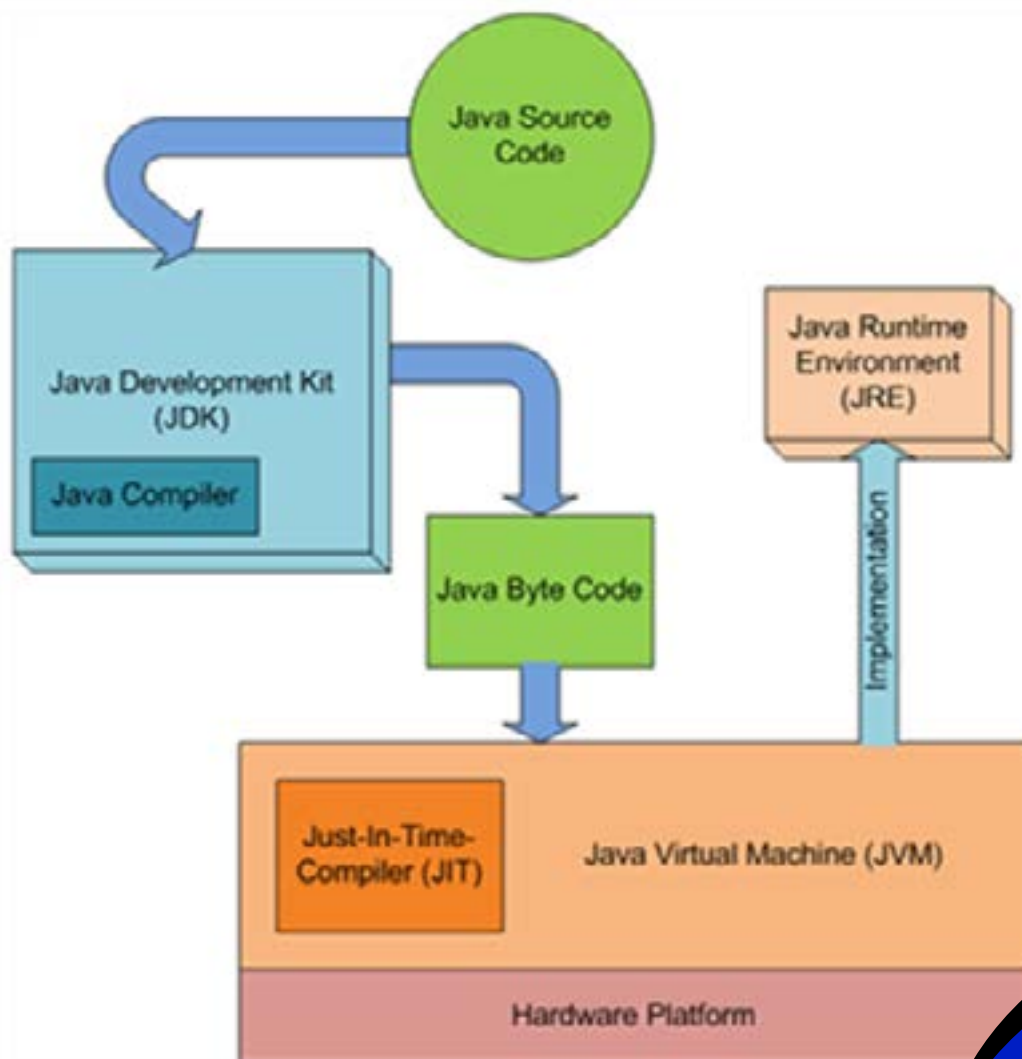
Es una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el bytecode Java), el cual es generado por el compilador del lenguaje Java.

Java Development Kit (JDK):

Se trata de un paquete de software que puede utilizar para desarrollar aplicaciones basadas en Java.

Java Runtime Environment JRE:

es una máquina virtual de Java y su función es hacer de intermediario entre una aplicación programada en Java y el sistema operativo que se esté usando. De este modo, cualquier aplicación puede funcionar en cualquier sistema operativo que disponga del JRE.



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 %
9		Sublime Text	3.81 %	+0.1 %
10		Atom	3.78 %	+0.7 %

Estos son los IDE más usados actualmente en 2020, aun que algunos son de desarrollo de android. Por lo cual solo voy a desarrollar los más usado para desarrollo web.

Visual Studio:

Microsoft Visual Studio es un entorno de IDE para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP.

Fue desarrollado por microsoft en 1997.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET

Características:

- Editor de código.
- Depurador de código.
- Diseño multiplataforma:
 - Diseñador de formularios de Windows.
 - Diseñador de WPF (Windows Presentation Foundation).
 - Diseñador / desarrollo web.
 - etc, [wiki](#).
- Es software propietario(de pago).
- Extensibilidad.

Eclipse:

Eclipse es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar aplicaciones.

Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados.

Eclipse fue desarrollado originalmente por **IBM** en el año 2001 como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la **Fundación Eclipse**.

características:

- Importar programas
- Programas de exportación
- Limpieza
- Migración al archivo JAR
- Cree y aplique el script
- Historial
- Construir Automáticamente
- Crear un conjunto de trabajo
- Herramienta de depuración
- Configuración de depuración
- Puntos de ruptura
- Interfaces
- Herramientas externas.

Visual Studio Code:

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows , Linux y macOS.

Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias.

características:

- Resaltado de sintaxis.
- [Snippets](#).
- Autocompletado de código.
- Refactorización.
- Depuración.
- Software libre y gratuito.
- Etc [wiki](#).

PyCharm:

PyCharm es un entorno de desarrollo integrado (IDE), específicamente para el lenguaje Python. Está desarrollado por la empresa checa JetBrains en el año 2010.

PyCharm es un IDE multiplataforma, con versiones de Windows, macOS y Linux. La Community Edition se publica bajo la licencia Apache y también hay una edición profesional con características adicionales, lanzadas bajo una licencia propietaria.

características:

- Asistencia y análisis de codificación, con finalización de código, resaltado de sintaxis y errores, integración de linter y soluciones rápidas.
- Navegación ágil por proyectos y códigos: vistas de proyectos especializadas, vistas de estructura de archivos y salto rápido entre archivos, clases, métodos de usos.
- Refactoring de Python: incluye cambio de nombre, método de extracción, introducción de variable, introducción de constante, pull up, push down y otros.
- Soporte para frameworks web: Django, web2py y Flask [solo edición profesional].
- Depurador de Python integrado.
- Pruebas de integridad, con cobertura de código línea por línea.
- Desarrollo de Python de Google App Engine [solo edición profesional].
- Integración de control de versiones: interfaz de usuario unificada para Mercurial, Git, Subversion, Perforce y CVS con listas de cambios y combinación.
- Soporte para herramientas científicas como matplotlib, numpy y scipy [solo edición profesional].

NetBeans:

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE1 es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation).

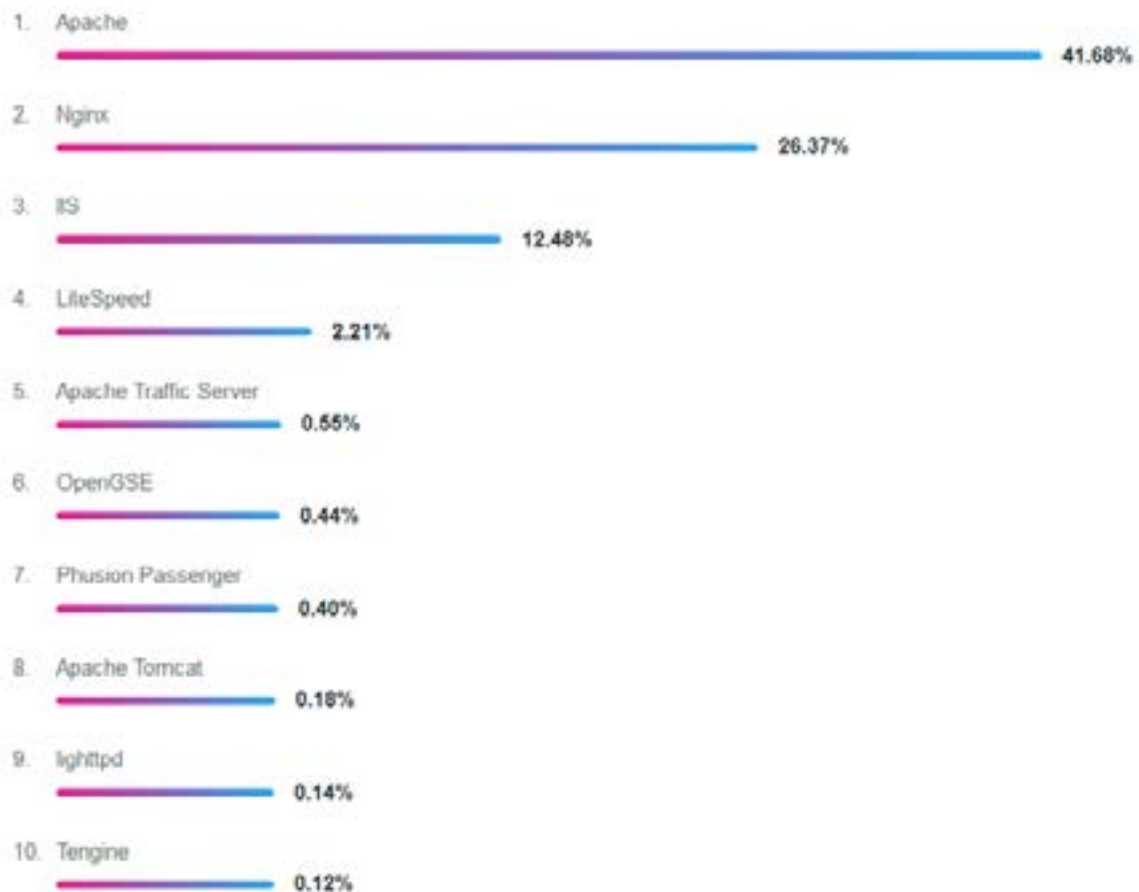
Fue desarrollado por Apache Software Foundation, Oracle Corporation y por Sun Microsystems en el año 1996.

características:

- Edición de código rápido e inteligente.
- Gestión de proyectos fácil y eficiente.
- Desarrollo rápido de la interfaz de usuario
- Escribe códigos libres de errores.
- Editor de código, multilenguaje.
- Depurado de errores.
- Acceso a base de datos.

Entre otros muchos.

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



Nombre	Descripción	Características	Fecha de creación	Desarrollador
HTTP Apache	Es un servidor web HTTP de código abierto, para multiplataforma que implementa el protocolo HTTP.	<ul style="list-style-type: none"> • Modular. • Código abierto. • Multi-plataforma. • Extensible. • Popular (fácil conseguir ayuda/ soporte) 	1995	Apache Software Foundation
Nginx	Es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3).	<ul style="list-style-type: none"> • Software multiplataforma . • Consume menos recursos que la mayoría • Alto rendimiento soportando mayor carga y respondiendo. • Puede ser usado como proxy inverso. • Podemos integrarlo junto con Apache, de forma que Nginx procese contenido estático y Apache contenido dinámico. 	2004	NGINX, Inc. Igor Sysoev

IIS	Es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows.	<ul style="list-style-type: none"> • Proporcionan capacidades de servidor web integrado. Es confiable, seguro y administrable en internet. • Desarrolla y es compatible con las aplicaciones beneficiándose con un único entorno de alojamiento de aplicaciones integrado con compatibilidad total. • Al momento de la instalación permite elegir sobre que servidor web va a correr (Apache o IIS). 	1995	Microsoft
LittleSpeed	LiteSpeed Web Server (LSWS), es un software de servidor web propietario. El software es un reemplazo directo de Apache, lo que significa que usa el mismo formato de configuración que Apache.	Compatible con las funciones de Apache de uso común, incluidas mod_rewrite, .htaccess mod_security.	2003	LiteSpeed Technologies
Apache Traffic server	El servidor de tráfico Apache (ATS) es un servidor proxy inverso y proxy de reenvío modular, de alto rendimiento , generalmente comparable a Nginx y Squid	<ul style="list-style-type: none"> • Caching. • Servidor proxy. • Es rápido. • Extensible gracias a las api's y plug-in. Más info web oficial	2019	Fundación de software Apache
OpenGSE	No encuentro información concluyente		2009 fue cuando se libero	Google

Phusion Passenger	Es un servidor web gratuito y un servidor de aplicaciones con soporte para Ruby, Python y Node.js. Está diseñado para integrarse en el servidor HTTP Apache o el servidor web nginx, pero también tiene un modo para ejecutarse de forma independiente sin un servidor web externo.	<ul style="list-style-type: none"> • Desarrolla con confianza. • Usabilidad amigable • Confiable y segura. • Rápido, eficiente y escalable. Más oficial-web	2009	Phusion
Apache Tomcat	Apache Tomcat funciona como un contenedor de servlets, javaserver pages, Java expression language y websocket. También proporciona un entorno de servidor web HTTP	<ul style="list-style-type: none"> • Usa Catalina (a servlet container). • Usa Coyote (an HTTP connector). • Usa Jasper (a JSP engine). • Usa Cluster Más info wiki	1999	Apache Software Foundation
Lighttpd	Es un servidor web de código abierto optimizado para entornos de velocidad crítica mientras que cumple con los estándares, es seguro y flexible.	<ul style="list-style-type: none"> • Autenticación contra un servidor LDAP. • Estadísticas de RRDtool. • Alojamiento virtual flexible. • Soporte de módulos. Más wiki	2003	Jan Kneschke
Tengine	Tengine es un servidor web de comercio electrónico más grande de Asia. Está basado en el servidor HTTP Nginx y tiene muchas funciones avanzadas. Tengine ha demostrado ser muy estable y eficiente	<ul style="list-style-type: none"> • Admite el método conector HTTP para el proxy de reenvío. • Monitor de operaciones mejorado, como registro y reversión asíncronos, almacenamiento en caché de DNS, uso de memoria, etc. • Envía cargas sin búfer directamente a servidores backend HTTP y Fast-CGI, lo que ahorra E / S de disco. Más oficial web	2011	Taobao

14º Apache HTTP vs Apache Tomcat.

Apache es un servidor web HTTPS tradicional, optimizado para manejar contenido web estático y dinámico (muy a menudo basado en PHP), carece de la capacidad de administrar Servlets Java y JSP. Tomcat, por otro lado, está casi totalmente orientado al contenido basado en Java. De hecho, Tomcat se desarrolló originalmente como un medio para proporcionar la funcionalidad JSP que Apache carecía.

Incluso con esto en mente, una comparación entre Tomcat Server y Apache no se reduce a una competencia directa. Esto se debe a que es completamente viable ejecutarlos uno al lado del otro. Entonces, en proyectos que involucran contenido basado en Java y PHP, por ejemplo, tiene sentido que Apache maneje la mayor parte del contenido estático y dinámico, mientras que Tomcat se encarga del JSP.

Sin embargo, por sí solo, Tomcat no es particularmente eficiente como un servidor HTTP tradicional, por lo que Apache es una opción mucho mejor para sitios web dinámicos creados únicamente con un lenguaje como PHP. Pero, por supuesto, cuando se trata de sitios basados completamente en JSP, Tomcat es el líder indiscutible de la pandilla.

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

En construcción