



Índice

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.....	2
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.....	3
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.....	4
4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme)/URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.....	5
5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.....	6
6. Modelo de división funcional front-end / back-end para aplicaciones web.....	7
7. Página web estática – página web dinámica – aplicación web – mashup	8
8. Componentes de una aplicación web.....	9
9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación.....	10
utilizados en cada caso.....	10
10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).....	11
11. Características y posibilidades de desarrollo de una plataforma XAMPP.....	12
12. 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.....	13
13. IDE más utilizados (características y grado de implantación actual).....	14
14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual)..	15
15. Apache HTTP vs Apache Tomcat.....	16
16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).	17
17. Generadores de documentación HTML (PHPDocPHPDocumentor, ApiGen, ...).	18
18. Repositorios de software – sistemas de control de versiones: GIT , CVS, Subversion,	19

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

Los protocolos de comunicación son una serie de normas que se siguen durante un intercambio de información entre 2 máquinas (pc-pc, pc-móvil, pc-servidor...)

Protocolo TCP: El protocolo TCP descompone los archivos a enviar en paquetes y posteriormente los envía al protocolo IP.

Protocolo IP: Suele ir ligado al protocolo TCP, este protocolo asigna la información de la IP destino a los paquetes enviados, cada equipo cuenta con una dirección IP.



Protocolo HTTP: Su propósito es transferir información entre dos programas (texto, imágenes, video, sonido, etc.) mediante protocolo TCP/IP usando distintos métodos (post, get, delete...)

Protocolo HTTPS: Igual que HTTP solo que cifrando la información que se envía.

Aplicación	HTTP
Transporte	TCP
Red	IP

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

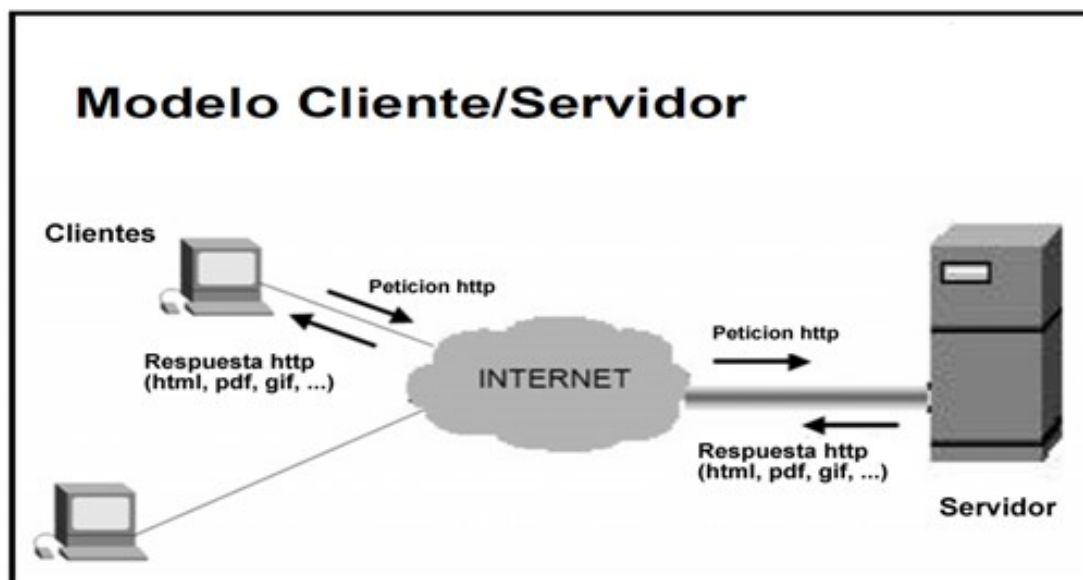
Cómo interactúa un usuario con una aplicación

A través del front end el usuario interactúa con la aplicación, el back end realiza funciones en base a dicha interacción. Un ejemplo de esto es, por ejemplo, realizar una búsqueda en google.

Nosotros introducimos ciertos parámetros en la barra de búsqueda y el servidor nos envía una serie de sitios web acordes a nuestra búsqueda.

El modelo HTTP Es un buen ejemplo de modelo cliente-servidor

En una aplicación de escritorio, tanto el cliente y el servidor están en una misma maquina, sin embargo en las aplicaciones web, el cliente y el servidor están en maquinas separadas, pudiendo el servidor dividirse en varias maquinas, lo que facilita y hace mas claro el diseño del sistema



<https://es.wikipedia.org/wiki/Cliente-servidor>

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

Por excelencia, los métodos mas usados son Get y Post

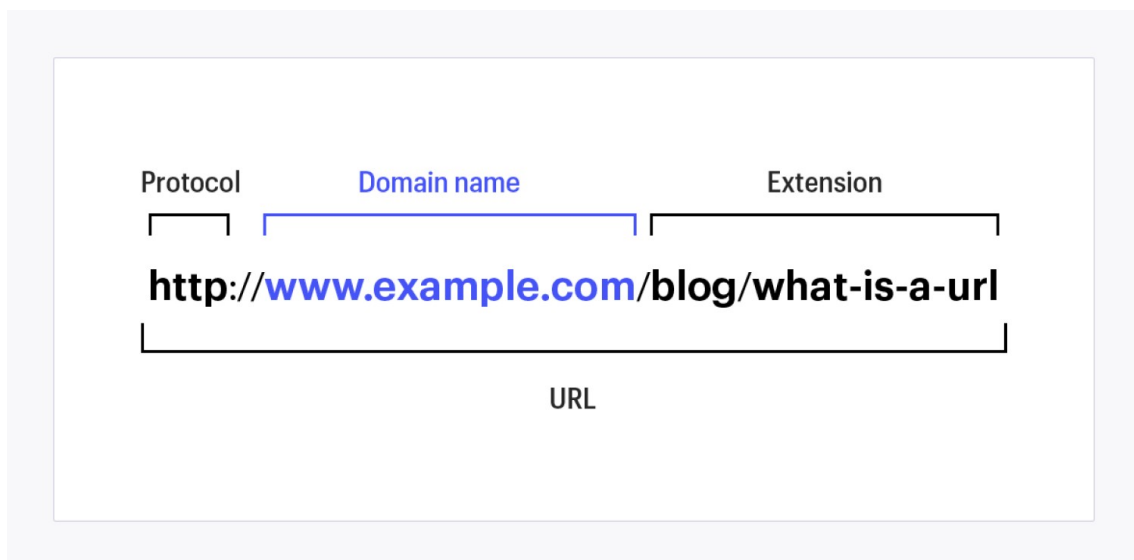
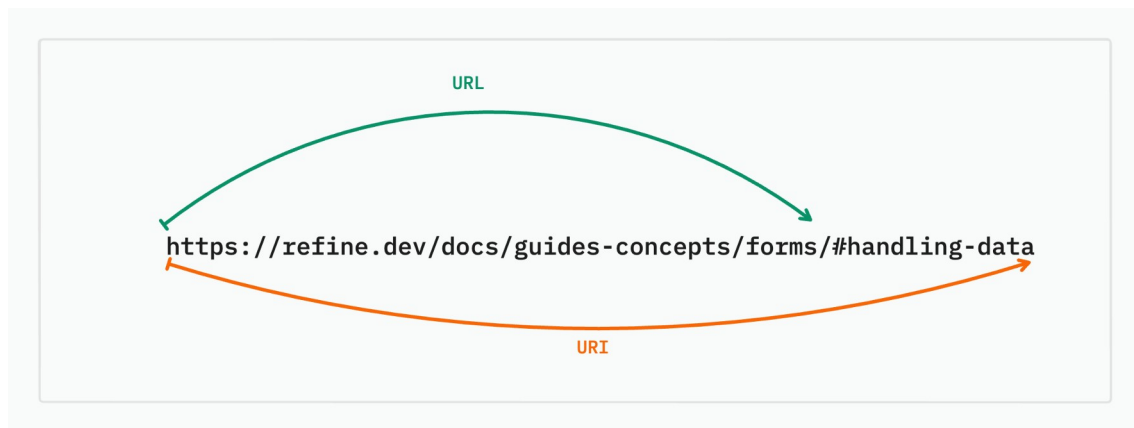
GET	Solicita al servidor un objeto, recurso o documento
POST	Envía al servidor un documento
PUT	Solicita al servidor que haga accesible el documento enviado en una URI específica
DELETE	Solicita al servidor que elimine un documento
TRACE	Solicita una copia de la petición enviada
HEAD	Solicita al servidor el encabezado del recurso
OPTIONS	Solicita los métodos HTTP soportados por el servidor
CONNECT	Pregunta si se tiene acceso al servidor

4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme)/URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS

La URI es una dirección que sirve para localizar recursos en la web y en la máquina local, sin embargo la URL solo sirve para localizar recursos en la web

La URI está compuesta por: protocolo de comunicación+servidor+puerto(si es necesario)+ruta del archivo

La URL está compuesta por: protocolo+dominio+ruta



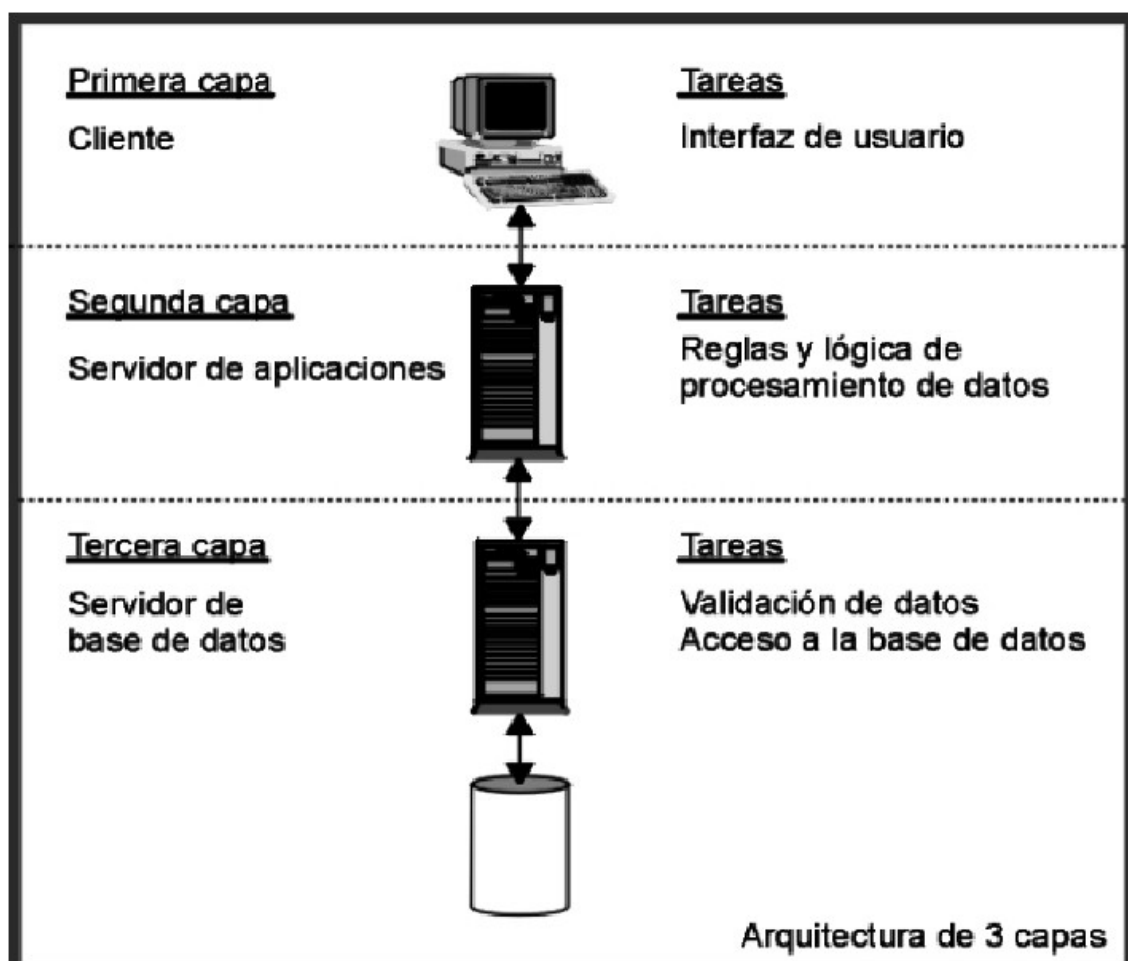
5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.

Las aplicaciones multicapa se separan en 3 capas que se comunican entre ellas

Presentación: Es la capa que ve el usuario, comunica y recoge información del mismo, se comunica con la capa de negocio

Negocio: Donde están alojados los programas que se ejecutan, se reciben y procesan las respuestas del usuario y se envían respuestas al mismo tras procesarlas, esta capa se comunica con la capa de presentación y la de datos

Datos: Es la encargada de acceder a los datos necesarios, se comunica con la capa de negocio para almacenar y enviar datos

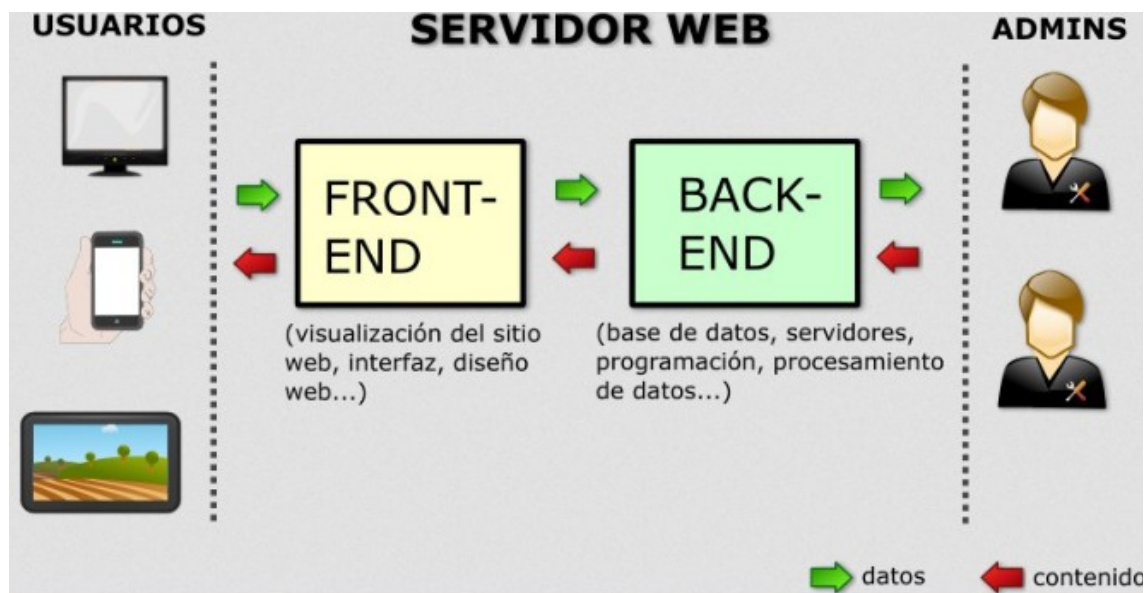


6. Modelo de división funcional front-end / back-end para aplicaciones web

Distintas funcionalidades en una aplicación

Las funcionalidades front end corresponden a usuarios normales, registrados o no, tales como acceder a una aplicación, administrar su usuario, o leer publicaciones

Las funcionalidades back end corresponden a usuarios especiales como administradores o publicadores, usuarios con mas privilegios que por ejemplo, pueden administrar usuarios, eliminarlos, publicar artículos, etc...



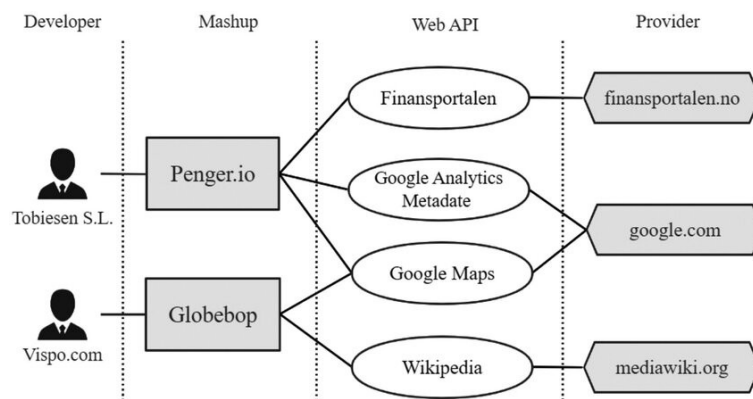
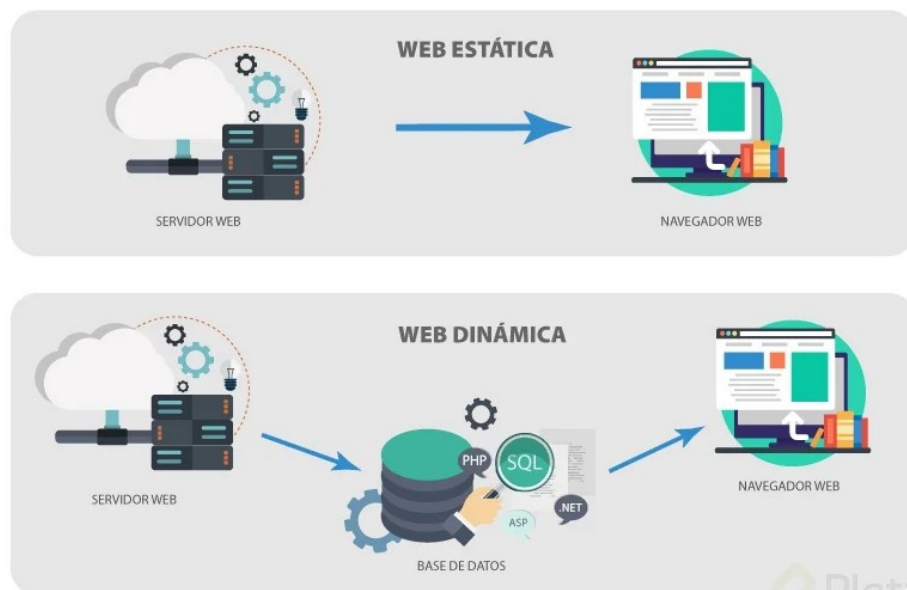
7. Página web estática – página web dinámica – aplicación web – mashup .

Página web estática: Es aquella que no permite al usuario modificar su contenido, la información mostrada es siempre la misma y las actualizaciones siempre vienen por parte del servidor

Página web dinámica: Es aquella que permite la interacción con el usuario y se actualiza a tiempo real, puede automatizarse y esta en cambio constante

Aplicación web: Es un software con interacción cliente-servidor que se ejecuta en el navegador, envía, recibe y procesa información/archivos del usuario

Mashup: Es una aplicación que combina servicios de varias paginas web, usan contenido de terceros a través de interfaces publicas, ejemplos de esto son Google maps y Youtube



8. Componentes de una aplicación web.

Cliente: Donde se visualiza la aplicación web, el usuario interactúa con la aplicación por aquí

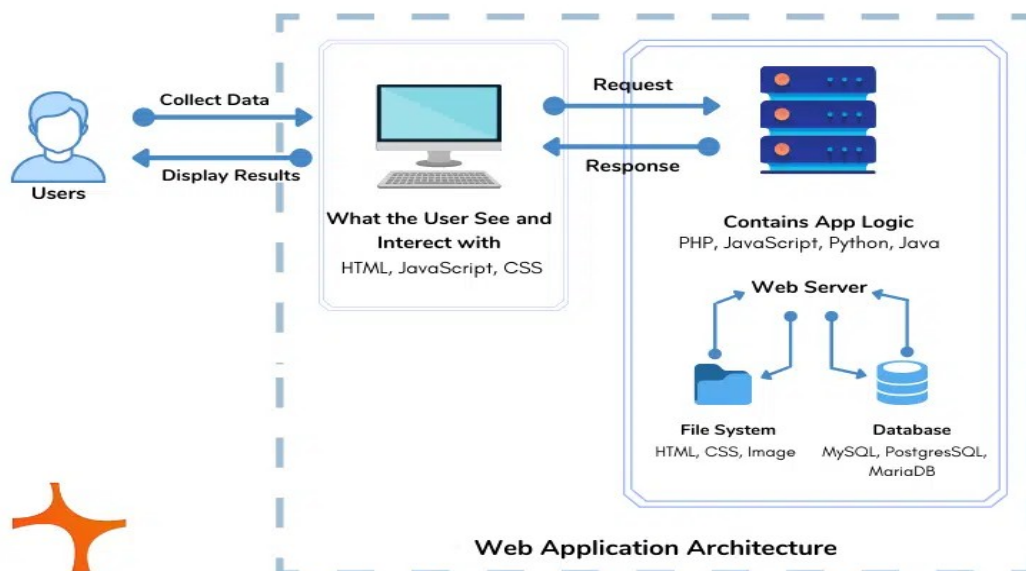
Servidor web: Donde se almacena la aplicación, cambia dependiendo del lenguaje/lenguajes en el que este programada la aplicación

Modulo encargado de ejecutar el código: Se encarga de interpretar/ejecutar los ficheros escritos en lenguajes de programación

Sistema gestor de base de datos(si es necesario): Almacena y administra datos

Ficheros escritos en lenguajes de programación: Archivos en los que se encuentra el código encargado de distintas funciones

Web Application Architecture

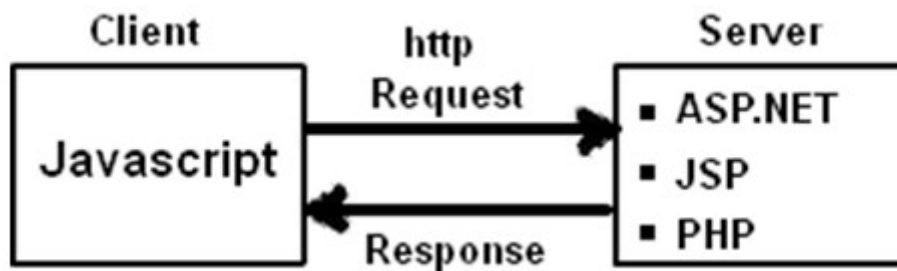


9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.

Los lenguajes de programación ejecutados del lado del cliente son aquellos ejecutados/interpretados por el navegador web: HTML, Flash, CSS...

Los lenguajes de programación ejecutados del lado del servidor son aquellos ejecutados e interpretados por el servidor en el que se aloja la aplicación: PHP, Ruby, ASP.net,

Hay lenguajes que pueden ejecutarse en ambos lados, tales como Java y JavaScript



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

Lenguajes usados en el back-end

Ejemplos de lenguajes usados en el lado del servidor o back-end son:

PHP, que cuenta con la ventaja de tener facilidad para conectarse a todo tipo de bases de datos. Es uno de los lenguajes mas usados

Perl, que trabaja bien procesando tanto archivos como textos a pesar de que peca de ser lento en algunos escenarios

ASP.NET, el cual tiene compatibilidad con todos los navegadores existentes

Ruby, el cual cuenta con la ventaja de contar con herramientas preinstaladas que facilitan el trabajo



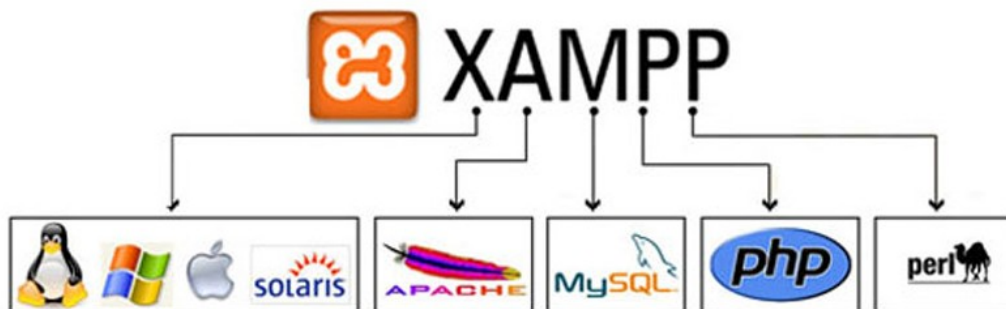
<https://axarnet.es/blog/lenguajes-del-lado-del-servidor>

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

Xampp es un servidor web local que trae integrado Apache HTTP, MariaDB, PHP y Perl

Este servidor es fácil de instalar y usar, así como útil para probar aplicaciones web antes de desplegarlos en un servidor en línea

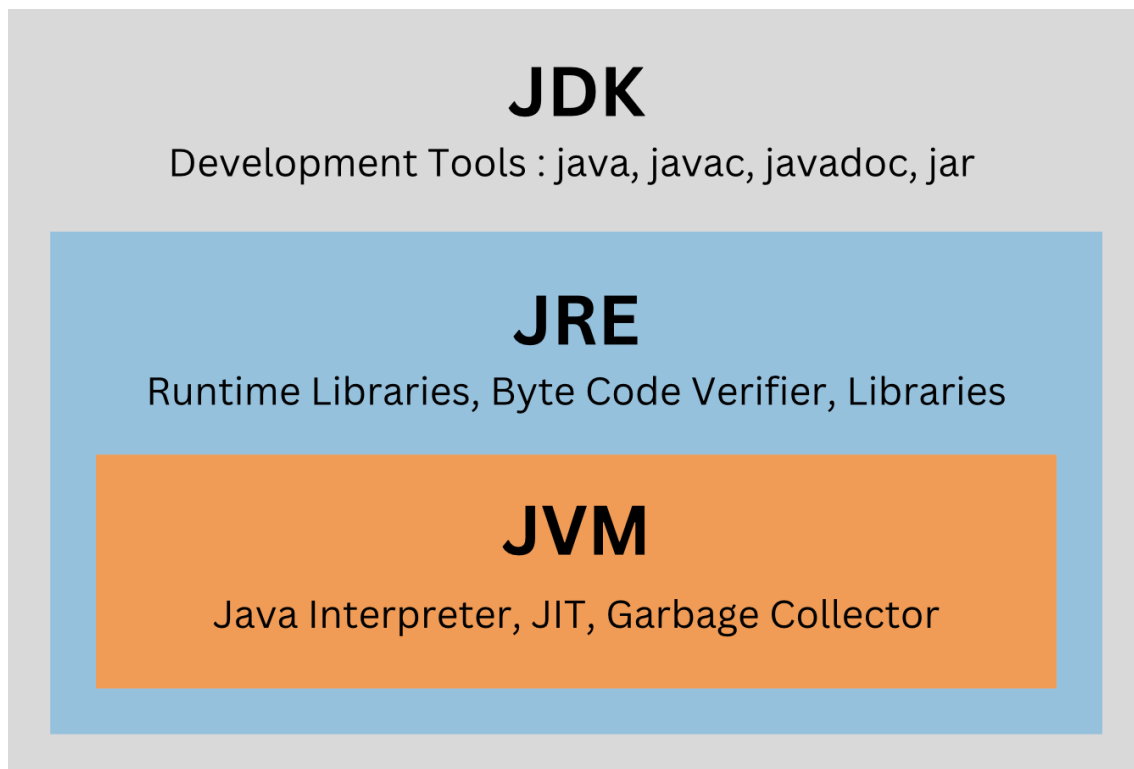
Xampp es multiplataforma, por lo que puede usarse en Windows, Linux y macOS, además al ser muy portable, el traslado de proyectos entre distintos sistemas operativos es muy sencillo



12. 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.

El JDK (Java developers toolkit) permite desarrollar aplicaciones basadas en java, a la par que compilar el código escrito en el mismo lenguaje, será necesario instalarlo si queremos escribir y compilar código escrito en java

La JVM se encarga de ejecutar código java compilado, será necesario instalarlo cuando haga falta ejecutar código java



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

Visual studio code: El IDE mas utilizado con diferencia, su ventaja es la enorme cantidad de extensiones disponibles

Eclipse: Puede usarse para programar en C, C++, Objective-C, PHP, Perl, Java, GNU Pascal, Free Pascal y Java, es de código abierto

NetBeans: Puede usarse para programar en Java, HTML5, CSS, C/C++, JavaScript y PHP, sin embargo no ofrece soporte para aplicaciones web

IntelliJ: Puede usarse para programar en Java, Kotlin, Python o Dart, destaca en su autocompletado y refactorización

Worldwide, Sept 2024 :				
Rank	Change	IDE	Share	1-year trend
1		Visual Studio	27.57 %	-0.3 %
2		Visual Studio Code	14.16 %	+0.3 %
3		Eclipse	11.89 %	+0.6 %
4		pyCharm	10.67 %	+1.2 %
5		Android Studio	9.89 %	+0.7 %
6		IntelliJ	8.03 %	+0.4 %
7		NetBeans	3.81 %	-0.3 %

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

Servidores web mas usados

Apache HTTP: Es de código abierto, uno de los servidores mas usados, cuenta con un gran numero de modulos que permiten ampliar las funcionalidades del servidor, soporta Perl, Python y PHP (No confundir con Apache Tomcat, que es un servidor web que solo admite java)

Ngix: Cuenta con un alto rendimiento, distribuye solicitudes de una manera eficiente, cuenta con soporte nativo para HTTPS, el comportamiento del servidor es muy moldeable, soporta multitud de lenguajes de programación, es el mas usado actualmente

LiteSpeed: Cuenta con una arquitectura basada en eventos, lo que significa que para cada petición por parte del front-end usa procesos ya abiertos en vez de abrir nuevos, lo que lo hace muy eficiente, a parte, es compatible con apache, lo que facilita las migraciones desde Apache HTTP



<https://www.sysadminok.es/blog/hosting/servidores-web-mas-utilizados/#:~:text=Apache%2C%20Nginx%2C%20Microsoft%20IIS%20y,web%20m%C3%A1s%20populares%20del%20mundo.>

https://www.twaino.com/es/blog/crear-un-sitio-web/servidor-apache-una-guia-completa-para-principiantes/#21_Ventajas_de_Apache

<https://thepower.education/blog/que-es-nginxhttps://rafarjonilla.com/que-es/servidor-litespeed/>

15. Apache HTTP vs Apache Tomcat

Tanto Apache HTTP como Apache tomcat son servidores web, la diferencia entre estos dos es que Apache tomcat es un servidor pensado para aplicaciones programadas en Java. Mientras que Apache HTTP esta pensado para aplicaciones programadas en PHP, Perl o Python



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

Google chrome: Basado en chromium, es el navegador mas usado con diferencia, es intuitivo, seguro, sencillo de usar, y cuenta con gran cantidad de extensiones

Safari: Navegador de dispositivos apple, destaca en su seguridad y velocidades

Microsoft edge: Tambien basado en chromium, destaaca por su seguridad y por la inclusión de cortana

Firefox: Destaca por estar muy optimizado, ser veloz y seguro. A demas es de codigo abierto

Opera: Destaca por tener un bloqueador de anuncios integrado



17. Generadores de documentación HTML (PHPDoc, PHPDocumentor, ApiGen, ...)

Los generadores de documentación se encargan de generar una api a partir de comentarios que incluyamos en nuestro código, esto es útil para usuarios externos o incluso nosotros mismos si volvemos a mirar ese código tiempo después y no recordamos que hacia cada cosa

PHPDoc: Derivado de Javadoc (generador de documentación para java), tiene la ventaja de que algunos IDEs pueden extraer información de la documentación y mostrarla en tiempo real

PHPDocumentor: Permite documentar código tanto orientado a objetos como procedimental (crear funciones para procesos que se van a repetir múltiples veces). Puede generar documentos HTML, PDF, CHM y Docbook

ApiGen: También para PHP, permite ser configurado a nuestro gusto mediante el fichero "apigen.neon"

18. Repositorios de software – sistemas de control de versiones: GIT , CVS, Subversion, ...

Repositorios

Un repositorio es una útil herramienta que permite llevar un seguimiento en nuestra aplicación y permite deshacer cambios hechos si no nos convencen, al incluir nuestro proyecto a un repositorio tendremos la opción de crear “puntos de guardado” cuando nosotros queramos, hacer cambios en la app y, si no nos convencen, volver a ese punto de guardado. De convencernos, podremos guardar estos cambios creando un nuevo punto de guardado.

También tenemos la opción de usar ramas, las cuales nos permiten crear versiones de la misma aplicación, puede usarse, por ejemplo, para tener una rama “master” en la que hay una versión funcional de la aplicación, y otras ramas en las que se le incluyen funcionalidades pero aun hace falta pulirlas y solucionar errores, cuando esta rama sea funcional, puede unirse con la rama “master”

Git destaca precisamente en esto, sus ramas son fáciles de juntar y ademas puede usarse para subir nuestra app a un repositorio remoto en GitHub. Esto lo hace el repositorio mas usado

Subversion funciona de manera similar y también permite el uso de un repositorio remoto

Por otro lado, CVS es mas tosco, por ejemplo, no permite renombrar los archivos del proyecto sobre el que trabaja, aunque puede manejar proyectos con múltiples ramas



<https://blog.onesaitplatform.com/2022/06/09/repositorios-de-software-primeros-pasos-con-git/>

<https://uniandesdsit.github.io/coding-guidelines/versioning/BRANCHES.html>

https://www.perforce-com.translate.goog/blog/vcs/what-svn?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq#what-is-a-subversion-svn-repository

<https://es.wikipedia.org/wiki/CVS>

<https://hackernoon.com/lang/es/top-10-sistemas-de-control-de-versiones-4d314cf7adea>