



DESARROLLO WEB ENTORNO SERVIDOR

Desarrollo Web Entorno Servidor (DWES)

2º DESARROLLO DE APLICACIONES WEB
15/10/2020

NEREA ÁLVAREZ JUSTEL

Contenido

1. Protocolos de comunicaciones: IP, TCP, HTTP y HTTPS.	2
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.	4
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.	5
4. Modelo de desarrollo de aplicaciones multicapa.	6
5. Modelo de dividir funcional front-end / back-end para aplicaciones web.	7
6. Página web estática – página web dinámica – aplicación web – mashup.	8
7. Componentes de una aplicación web.	9
8. Programas lado del cliente y lado del servidor – lenguaje en cada caso.	10
9. Lenguajes de programación utilizados en el lado del servidor en aplicaciones web.	11
10. Características y posibilidades de desarrollo de una plataforma XAMPP.	12
11. Instalación de JVM y JDK.	13
12. IDE más utilizados.	14
13. Servidores HTTP /HTTPS más utilizados.....	16
14. Apache HTTP vs Apache Tomcat.	17
15. Navegadores HTTP / HTTPS más utilizados.	18
16. Generadores de documentación HTML: PHPDoc – ApiGen.	19
17. Repositorios de software – sistemas de control de versiones	20
18. Propuesta de configuración del entorno desarrollo en este curso.	21
19. Propuesta de configuración del entorno de explotación en este curso.	22
20. Realiza un estudio de los siguientes conceptos y relación con el desarrollo de aplicaciones web. .	23
21. Elegir y realizar un estudio sobre una de las arquitecturas de Desarrollo de Aplicaciones Web.	24
GLOSARIO	29
BIBLIOGRAFIA.....	30

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

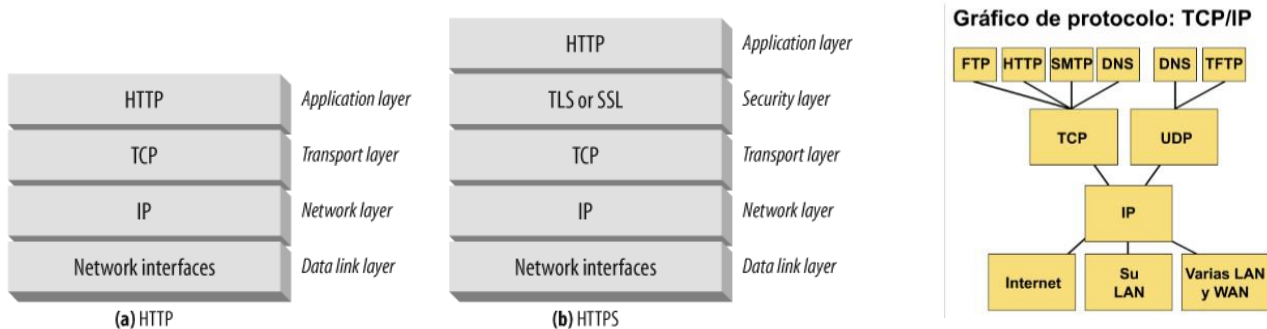
Se trata del **conjunto de pautas que posibilitan que distintos elementos que forman parte de un sistema establezcan comunicaciones entre sí, intercambiando información.**

IP: Se encarga del **transporte de paquetes desde el origen hasta el destino en una comunicación**, y a estos paquetes de información se les llama **datagrama**. **Socket** designa un concepto abstracto por el cual dos procesos pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

TCP: (Protocolo de Control de Transmisión) es **uno de los principales protocolos de la capa de transporte del modelo TCP/IP**. En el nivel de aplicación, **posibilita la administración de datos que vienen del nivel más bajo del modelo, o van hacia él**. TCP es un protocolo orientado a conexión, es decir, que **permite que dos máquinas que están comunicadas controlen el estado de la transmisión**.

HTTP: (Protocolo de transferencia de hipertexto - Hypertext Transfer Protocol) es el **protocolo de comunicación que permite las transferencias de información**. El más importante es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web para comunicarse. Es un **protocolo sin estado**, es decir, no guarda ninguna información sobre conexiones anteriores. Las aplicaciones web necesitan mantener estado, para esto se usan las cookies. Salió al mercado en 1991. Sigue el clásico **modelo cliente-servidor**, un cliente establece una conexión, realizando una petición a un servidor y espera una respuesta de este.

HTTPS: (Protocolo seguro de transferencia de hipertexto - Hypertext Transfer Protocol Secure), 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. Tiene a mayores la capa **TLS o SSL** que consta de **cifrado, integridad de datos y autenticación**. Salió al mercado en 1992.



Diferencias entre HTTP y HTTPS

En el protocolo **HTTP** las **URLs** comienzan con **"http://"** y utilizan por omisión el **puerto 80**, las URLs de **HTTPS** comienzan con **"https://"** y utilizan el **puerto 443** por omisión.

HTTP es inseguro y está sujeto a ataques man-in-the-middle y eavesdropping que pueden permitir al atacante obtener acceso a bancos y a cuentas de un sitio web e información confidencial. **HTTPS** está diseñado para resistir esos ataques y ser más seguro.

URI: (Uniform Resource Identifier -Identificador Uniforme de Recursos) es una **cadena de caracteres que identifica los recursos de una red** de forma unívoca. **Normalmente estos recursos son accesibles en una red o sistema.** Las URI engloban los otros dos conceptos, pudiendo clasificarlas entre URL, URN o ambas.

Un URI consta de las siguientes **partes**:

Esquema: nombre para asignar los identificadores, *e.g. urn;* *tag;* *cid:*. También identifica el protocolo de acceso al recurso, *http;* *mailto;* *ftp;* *etc.*

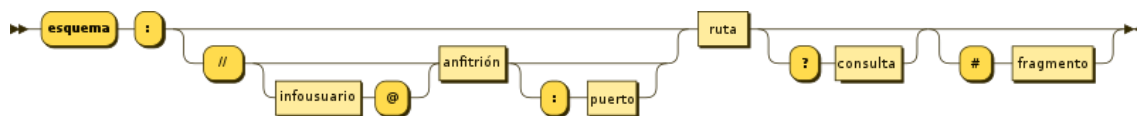
Autoridad: elemento jerárquico que identifica la autoridad de nombres, *//www.example.com*.

Ruta: información usualmente organizada en forma jerárquica, que identifica al recurso en el ámbito del esquema URI y la autoridad de nombres, *e.g.* `/domains/example`.

Consulta: información con estructura no jerárquica que identifica al recurso en el ámbito del esquema URI y la autoridad de nombres. Comienza mediante el carácter '?'.
Ejemplo: `?q=información sobre el recurso`

Fragmento: Permite identificar una parte del recurso principal, o vista de una representación de este. El comienzo se indica mediante el carácter '#'.
Ejemplo: #Código de identificación

Un URI se diferencia de un URL en que permite incluir en la dirección una subdirección, determinada por el “fragmento”.

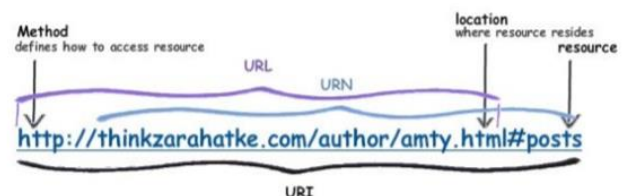
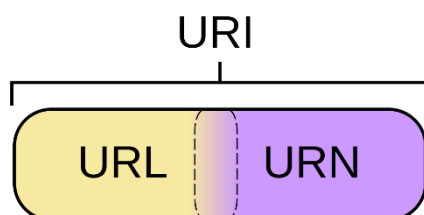


URL: (Uniform Resource Locator -Localizador Uniforme de Recursos) Son unas **cadena de texto** que se usan para nombrar recursos en Internet para su localización, cuyos recursos referidos **pueden cambiar**, esto es, la dirección puede apuntar a recursos variables en el tiempo.

Ejemplo: <http://es.wikipedia.org:80/wiki/Special:Search?search=tren&go=Go>

URN: (Uniform Resource Name -Nombre Uniforme de Recursos) Son unas **cadena de texto que se usan para nombrar recursos en Internet para su identificación, no indican exactamente dónde se encuentra ese objeto.**

Ejemplo: urn:isbn:0451450523



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

La arquitectura cliente-servidor es un modelo de aplicación distribuida 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 en un solo ordenador, aunque es mejor en un sistema operativo multiusuario distribuido a través de una red de ordenadores.

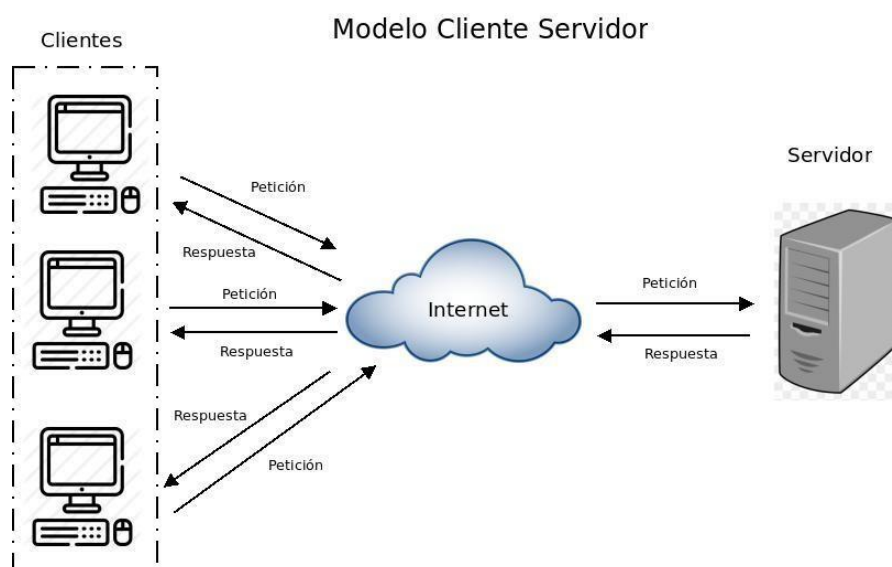
Teniendo varias ventajas y desventajas:

Ventajas:

- Centralización del control.
- Estabilidad.
- Fácil mantenimiento.
- Gran seguridad.

Desventajas:

- Congestión de tráfico.
- Imposible brindar el servicio si el servidor se cae.
- Dependiente de hardware y software.
- Dependencia del servidor para obtener información.



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

- El método **GET** se emplea para **adquirir un recurso** ubicado en un servidor web, que **devuelve los metadatos y el recurso** que se precisa.
 - El método **HEAD** realiza una función similar a la anterior, pero **sólo se solicitan los metadatos** y no el recurso en sí.
 - El método **POST** se usa para **enviar información** a un recurso web del servidor, **no para crear un elemento nuevo**.
 - El método **PUT** **crea un recurso nuevo** en el servidor o si éste ya existe **lo reemplaza**.
 - El método **DELETE** solicita **borrar el recurso especificado**.
 - El método **TRACE** sirve para el **diagnóstico de errores o detección de servidores intermedios** en la conexión mediante la monitorización de mensajes.
 - El método **CONNECT** solicita **saber si se tiene acceso a un host**.
 - El método **PATCH** solicita **sobrescribir completamente un recurso**. Se utiliza para **actualizar, de manera parcial, una o varias partes**.
- El método **OPTIONS** es utilizado para describir las **opciones de comunicación** para el recurso de destino.



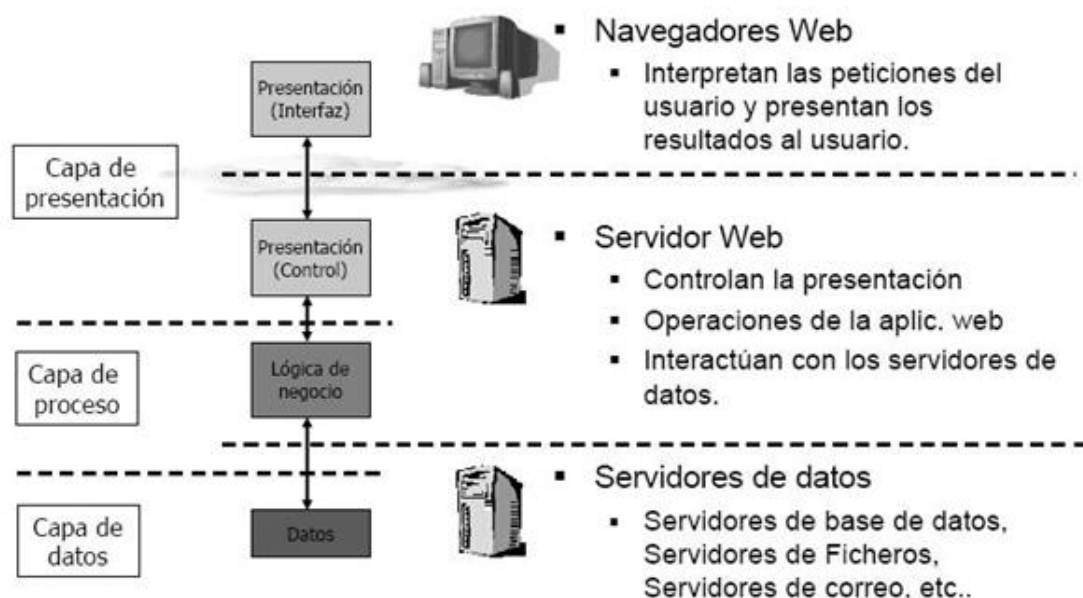
4. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes- funcionalidad de cada capa.

Es un tipo de desarrollo software donde **se persigue la separación de las partes de las que está compuesto un sistema** software en capas, éstas **sólo pueden comunicarse con la capa anterior, de la que reciben información y con la capa posterior, a la que envían información.**

Capa de Presentación: En esta capa se crea la **interfaz del usuario**. Su única función es pasarle las acciones que realice el usuario a la capa de negocio. Y a su vez devolver la petición al usuario con el contenido correspondiente.

Capa de Negocio: En esta capa se gestiona la **lógica de la aplicación**. Es donde se dice qué se hace con los datos. Estará **conectada con la capa de persistencia** para poder realizar sus funciones. De forma simple seria, Se recibe la información del usuario y se envían los datos tras el procesamiento.

Capa de Persistencia: Esta capa se encarga de guardar los datos. Será donde se **gestione todo lo relativo a la base de datos** y a la creación, edición y borrado de datos de ésta. Se puede tener **más de un gestor de bases de datos**.



5. Modelo de dividir funcional front-end / back-end para aplicaciones web. (NAJ)

Front-end

Es la parte que ve el usuario y en la que sí se incluyen la línea de diseño y los elementos gráficos de la página. Será aquí donde se incluyan **los estilos, los colores, los fondos, tamaños y las animaciones del sitio web**.

Para conseguir todas estas funciones, **es necesario trabajar sobre el código de la página**. Trabaja con otro tipo de lenguajes, más cercanos a la comprensión y la interacción del usuario.

Estos especialistas **son los encargados de maquetar la estructura semántica del contenido**, a través del lenguaje de **HTML**, simplificar todo en hojas de estilo con CSS y permitir que el usuario interactúe con nuestra web gracias a JavaScript.

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**.

Los encargados del desarrollo del Back End deben manejar eficientemente varios códigos de programación, en lenguajes como **PHP, Ruby, Python, Java, SQL, MongoDB, MySQL, etc.**

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.

Ambas tienen tanto programación del lado del servidor como del lado de cliente



En la actualidad se está hablando de **full-stack**.

Esa mentalidad full-stack **permite trabajar en cualquier faceta de una funcionalidad end-to-end**. Es decir, ser capaz de programar tanto en el lado **cliente como servidor**, conociendo cómo se comportan cada parte en conjunto. Arreglar y solucionar cualquier reto, y no la mitad de él.

6. Página web estática – página web dinámica – aplicación web – mashup. (NAJ)

Aplicación web estática: son las **más sencillas** y no suelen tener muchos cambios al no ser sencillos. Están enfocadas principalmente a **mostrar una información permanente**, donde el navegante se limita a obtener dicha información, **sin poder interactuar** con la página web visitada. Están desarrolladas con **código HTML y CSS** y pudiendo mostrar algún banners o vídeos, entre otros. La razón para no tener muchas variaciones es la descarga del HTML, modificado y subida del mismo.

Ejemplo: - Web Quest, El principal objetivo reside en aprender los contenidos del tema elegido de una manera interactiva.

Aplicación web dinámica: son mucho más complejas que las anteriores técnicamente hablando, contiene **aplicaciones dentro de la propia web**, otorgando mayor interactividad con el navegante y **actualiza información y contenido cada entrada de un usuario a la web**. Existen muchos **lenguajes** de programación, pero los **más comunes son PHP y JavaScript**. Además de actualizar también se puede **modificar el diseño** de la web. Es importante **no confundir multimedia e interactividad con páginas dinámicas**.

Ejemplo: -Encuestas y votaciones

En las páginas **dinámicas**, el contenido suele **generarse en el momento de visualizarse**, pudiendo variar por tanto este, mientras que en las **estáticas** el contenido **suele estar predeterminado**.

Aplicación web: software que los usuarios pueden utilizar accediendo a un servidor web a través de Internet (protocolo HTTP) o de una intranet mediante un navegador.

Las aplicaciones web están íntimamente **relacionadas con el almacenamiento de datos**, la información se guarda en **servidores web**, además de alojar, nos la envían en el momento que sea requerida, realizando copias temporales. Las empresas que alquilan estos espacios en los servidores web son conocidas como empresas o **servicios de Hosting**.

Mashup o aplicación web híbrida 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**. Es habitual que la integración de datos se obtenga desde **APIs** abiertas.

Funcionalidad:

- > La información disponible en Web y al interior de las empresas crece exponencialmente. Se necesita usar esa información para manipularla de manera rápida y sencilla.
- > La información se encuentra distribuida en diferentes fuentes de información.
- > Integración dinámica temporal a fuentes de información.
- > Integración realizada por el usuario y no necesariamente por el área de sistemas.
- > Acceso rápido a la información significa mayor competitividad y productividad.

7. Componentes de una aplicación web.

- Servidor web: Generalmente suele ser Apache, se encarga de la comunicación a través de la red con el navegador del usuario.
- Sistema gestor de bases de datos: Se encarga de organizar toda la información con la que se trabaja, las bases de datos más empleadas son las relaciones, especialmente MySQL.
- Módulo encargado de ejecutar el código: Se encarga de interpretar los comandos en el lenguaje utilizados (de Php, Asp, Perl, Python, Ruby) en la aplicación para procesarlos antes de enviar la página HTML al cliente.

- El Código para desarrollar las aplicaciones

Lenguaje de marcas:

HTML, XML, CSS, XSL.

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

8. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor – lenguaje de programación utilizado en cada caso.

Programas del lado del **cliente**

Navegador:

Es un **software, aplicación o programa que permite el acceso a la Web**, interpretando la información de distintos tipos de archivos y sitios web para que estos puedan ser vistos.

JavaScript:

Es un **lenguaje de programación interpretado**.

Flash Player:

Es una **aplicación informática englobada en la categoría de reproductor multimedia**.

Programas del lado el **servidor**

Apache:

Es un **servidor web HTTP de código abierto**.

MySQL:

Es un **sistema de gestión de bases de datos**.

PHP:

Es un **lenguaje de programación**, originalmente para el preprocesador de texto plano en UTF-8.

Nginx:

Es un **servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico**.

9. Lenguajes de programación utilizados en el lado del servidor en aplicaciones web.

- **Python:** Es un lenguaje multiplataforma con diversas herramientas y reduce considerablemente los procesos de desarrollo. En el año 2017 fue el lenguaje de programación más utilizado.
- **Java:** Está orientado a objetos y es uno de los lenguajes de programación más utilizados, es fácil de utilizar, multiplataforma, seguro fiable y eficiente. En el año 2017 fue el tercer lenguaje de programación más utilizado después de C.
- **JSP:** es un lenguaje para la creación de sitios web dinámicos con gran facilidad. Es similar a PHP, pero utiliza el lenguaje de programación de Java.
- **PHP:** Es un lenguaje interpretado, se usa para generar páginas web de contenido dinámico, es robusto y es fácil de aprender. Actualmente la implantación de PHP ha incrementado considerablemente en los 2 últimos años.
- **Perl:** Es un lenguaje interpretado, es muy dinámico y es especialmente útil para manipular cadenas de caracteres, archivos y procesos. Es el menos utilizado de los 4.

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

• **XAMPP:** Una de sus mejores características es que es **multiplataforma** teniendo versiones para muchos sistemas operativos, cuenta **con licencia libre y consta principalmente del servidor web Apache, la base de datos MySQL e intérpretes para PHP y Perl**, es **fácil de utilizar y permite interpretar páginas dinámicas**. Es una de las plataformas más intuitivas y utilizadas en Windows para desarrollo web.

• **Java EE:** Utiliza el lenguaje de programación **Java y cualquier base de datos**.

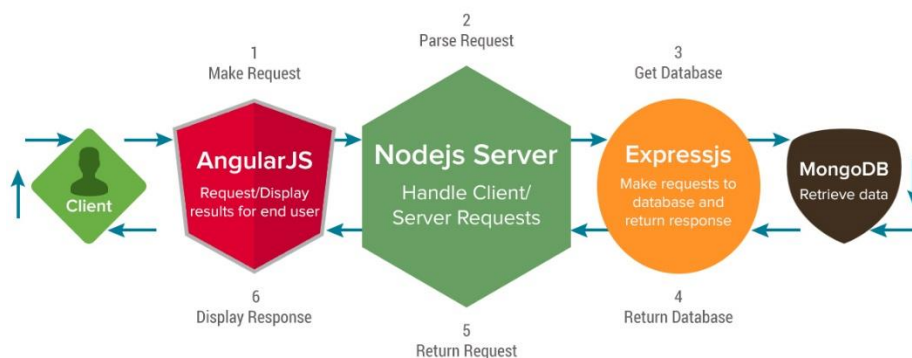
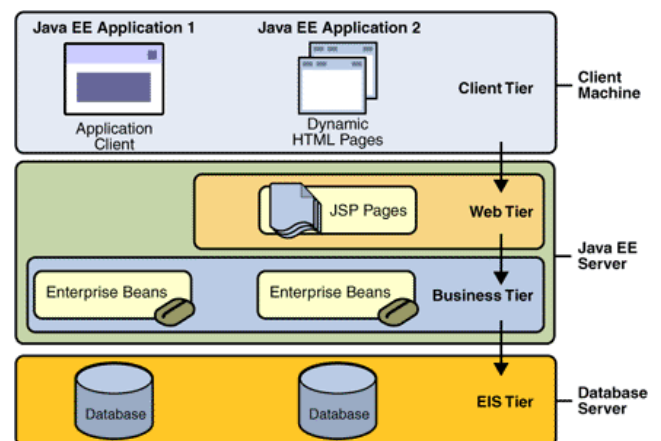
• **MEAN:** Utiliza el lenguaje de programación **JavaScript y la base de datos Mongo**.

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.

Desventajas:

- 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.



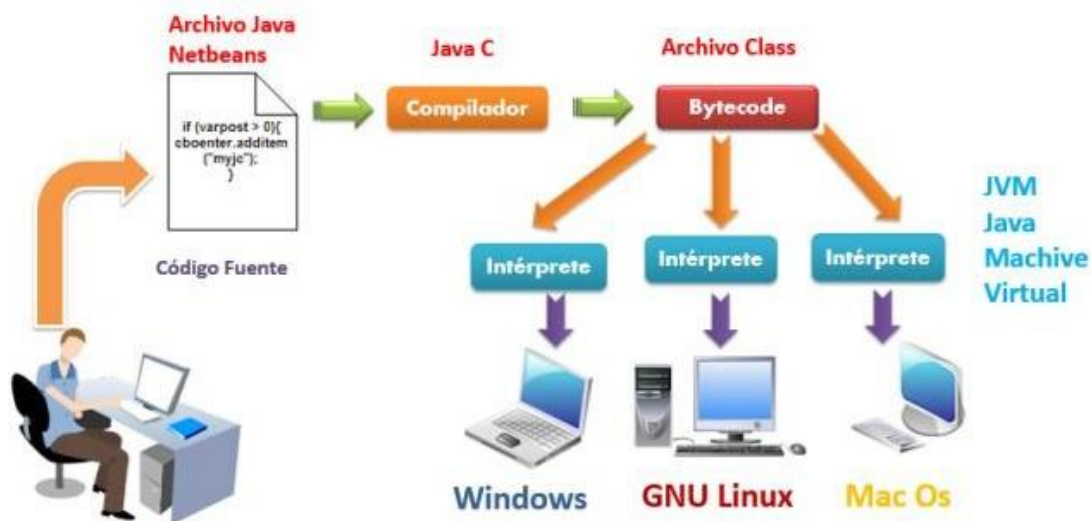
11. Instalación de JVM y JDK. En qué casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo en el entorno de explotación.

Sólo es necesaria la instalación del JDK en el entorno de desarrollo y del JVM en el entorno de explotación **cuando estamos desarrollando o ejecutando**, respectivamente, una aplicación web escrita o que contenga **lenguaje Java**, para otros lenguajes distintos no es necesario.

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 Developer Kit (JDK): se trata de un paquete de software provee herramientas de Desarrollo para la creación de programas en **Java**

Java Runtime Environment (JRE): 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.

Eclipse

Eclipse es uno de los entornos más conocidos y utilizados por los programadores, ya que se trata de un **entorno de programación de código abierto y multiplataforma**. Sirve para Java, C++, PHP, Perl y un largo etcétera. También nos permite realizar aplicaciones de escritorio y aplicaciones web por lo que nos brinda una gran versatilidad.

Además, está en constante evolución y hay muchos tutoriales por la red que nos guían en su instalación y utilización.



NetBeans

Netbeans también es un entorno de programación muy utilizado por los programadores. **Se trata de otro entorno multilenguaje y multiplataforma** en el cual podemos desarrollar software de calidad. Con él podemos crear **aplicaciones web y de escritorio**, además de contar con plugins para trabajar en **Android**.

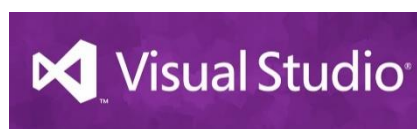
El lenguaje que mejor soporta es **Java**, ya que fue creado por Oracle y su creación fue para ser el IDE de Java. Aunque como hemos dicho, es multilenguaje debido a que soporta JavaScript, HTML5, PHP, C/C++ etc.



Visual Studio

Visual Studio fue diseñado por Microsoft y es uno de los mejores entornos de programación que existe siempre y cuando utilices sus lenguajes. Antiguamente tenían una versión de pago que incluía todos los lenguajes, y versiones express que eran gratuitas para un lenguaje en concreto.

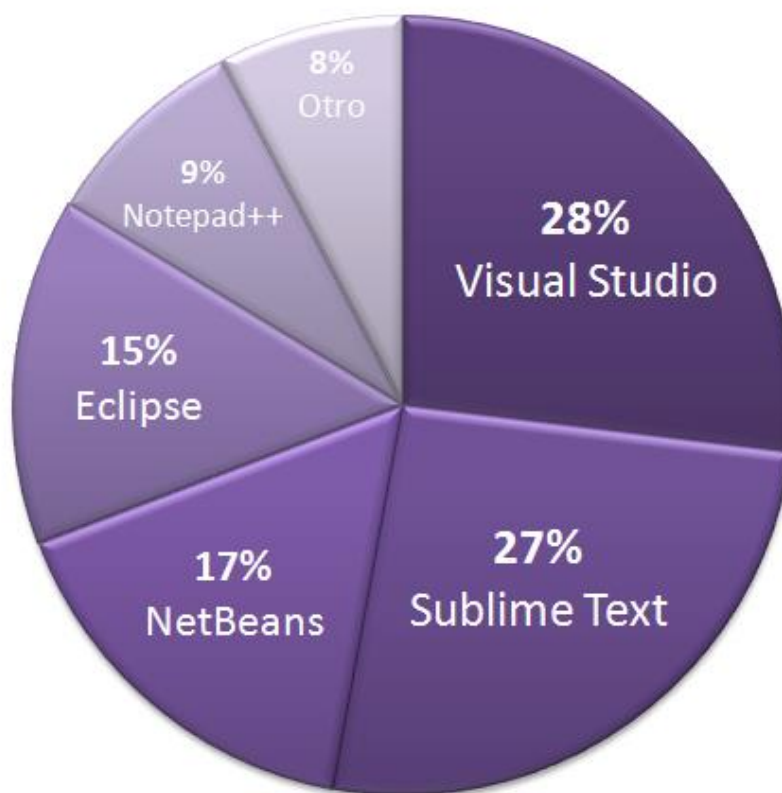
Ahora como Microsoft quiere pasarse al software libre, ha creado también un Visual Studio Community que es muy parecido al Visual Studio de pago, sólo que este está soportado por la comunidad. Este entorno nos permite hacer **aplicaciones web y de escritorio** y ayuda mucho al programador. El inconveniente que tiene es que **solo es válido para lenguajes de Microsoft**.



SublimeText

Como **software de pago** encontramos este editor de texto muy potente. **SublimeText** incorpora funcionalidades básicas y avanzadas. Uno de los puntos que se puede resaltar es la **capacidad de personalizar su entorno de trabajo**.

Sublime Text es multiplataforma y multilenguaje, sus colores te ayudarán a dar una mejor vista y organización a tu código.



13. Servidores HTTP /HTTPS más utilizados.

- **Apache:** Es el web server de referencia para Internet. Nació en 1996 y hasta el día de hoy sigue vigente. **Es el más usado.**

Ventajas:

- Código abierto
- Software gratuito
- Multiplataforma (Windows, Linux y Unix).

Desventajas:

- Bajo rendimiento cuando se reciben miles de requests.

- **Nginx:** Es un **servidor web de código abierto y gratuito** (aunque también existe una versión comercial) que se **destaca por su alto rendimiento**. Está **diseñado para ofrecer un bajo uso de memoria y alta concurrencia. Segundo más usado**

Ventajas:

- Software multiplataforma.
- Consume menos recursos que la mayoría de los servicios que hacen su misma función.
- 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.

Desventajas:

- No se integra con PHP de forma nativa. Es necesario usar FastCGI para poder integrarlo

- **Microsoft IIS:** Internet Information Services, también conocido como IIS, es un tipo de **servidor web creado por Microsoft específicamente para su plataforma de sistemas operativos Windows.**

Ventajas:

- 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.
- En la instalación permite elegir sobre que servidor web va a correr (Apache o IIS).

Desventajas:

- Este servidor no es multiplataforma, solo funciona bajo Windows.
- Posee limitaciones en las versiones que no son de la familia "Server".
- Posee vulnerabilidades.

- **GWS:** Google Web Server, se trata de un **servidor web privado escrito en C++, que es utilizado por Google para la mayoría de su infraestructura web. Está basado en Linux y no disponible para el público.** Alberga aproximadamente un 10 % de todas las páginas web activas del mundo.

14. Apache HTTP vs Apache Tomcat.

El servidor **HTTP Apache** es un servidor web HTTP de código abierto.

Apache presenta entre otras **características altamente configurables, bases de datos de autenticación y negociado de contenido**, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Las principales **ventajas** de Apache HTTP son la siguientes:

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/soporte)

Apache Tomcat funciona como un **contenedor de servlets**.

Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation.

Tomcat **fue escrito en Java**, funciona en cualquier sistema operativo que disponga de la **máquina virtual Java**.

15. Navegadores HHTP / HTTPS más utilizados.

- **Chrome:** Es un navegador web de **código cerrado** desarrollado por Google, aunque derivado de proyectos de código abierto (como el motor de renderizado Blink). Está disponible gratuitamente.

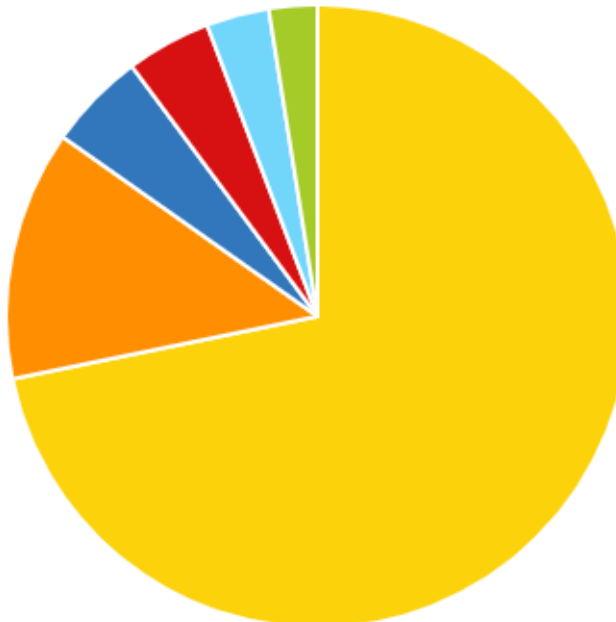
Alguna de sus **características**:

- RLZ identifier: Una cadena codificada enviada junto con todas las consultas a Google.
 - Posee un identificador (ID) único («clientID») para identificar al usuario en los registros de accesos.
 - Cuenta con una marca de tiempo de cuándo fue instalado el navegador.
 - Páginas de error alojadas en servidores de Google, cuando no se encuentra un servidor.
- **Mozilla Firefox:** es un navegador web **libre y de código abierto**, desarrollado para multiplataforma coordinado por la Corporación Mozilla y la Fundación Mozilla. Usa el motor Gecko para renderizar páginas web, el cual implementa actuales y futuros estándares web.

Alguna de sus **características**:

- Multiplataforma.
- Navegación con pestañas.
- Restauración de sesión.
- Corrector ortográfico.

Chrome Firefox Edge Opera Safari Otros



16. Generadores de documentación HTML: PHPDoc – ApiGen.

Es una herramienta de programación que **genera documentación destinada a los programadores (documentación de API) o a usuarios finales, o a ambos, a partir de un conjunto de código fuente especialmente documentado, y en algunos casos, archivos binarios.**

PHPDoc es una adaptación de javadoc para php que define un **estándar oficial para comentar código php**. **PhpDocumentor** es un **generador de documentación de código abierto escrito en PHP**.

ApiGen es una herramienta que **genera documentación para proyectos PHP**.

Alguna de las **ventajas** que se pueden obtener usando **generadores de documentación** son:

- La ayuda que recibe el programador para que todas sus aplicaciones tengan un formato de documentación similar.
- Se obliga al programador a ir llenando los campos necesarios para documentar.
- Es fácil estandarizar la documentación sobre qué hace una función, qué parámetros recibe, que valores devuelve, etc....

```
/**
 * @param int $disp_mode
 * @param $disp_option
 *
 * @internal param $total
 * @return int|null
 */
function setDisplayMode($disp_mode=0, $disp_option)
```

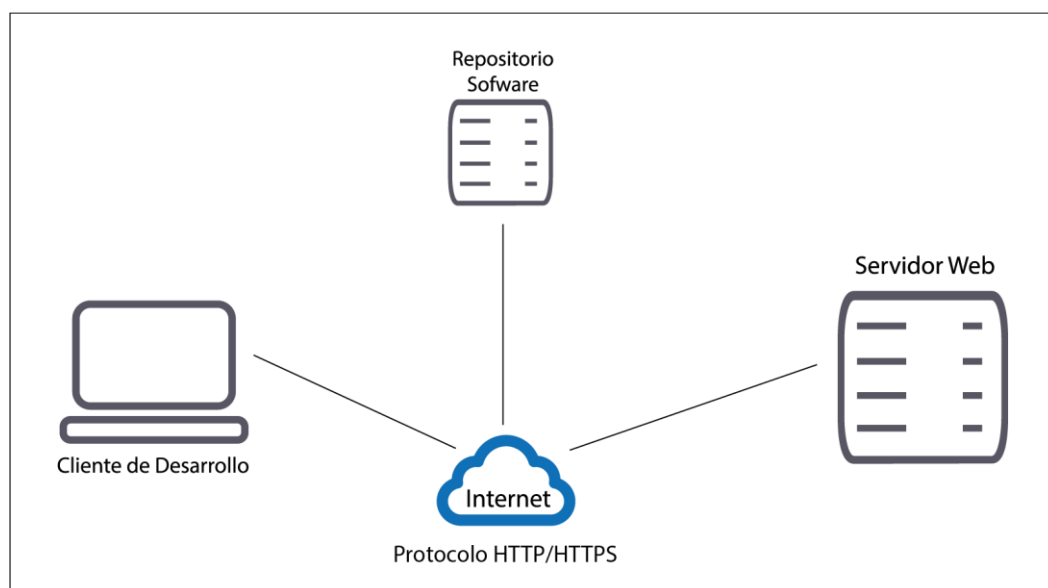
17. Repositorios de software – sistemas de control de versiones

Un repositorio de software es un lugar de almacenamiento del cual pueden ser recuperados e instalados los paquetes de software en un ordenador.

- **GIT:** es un **software de control de versiones** diseñado por Linus Torvalds. Su propósito es **llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos**.
- **CVS:** Concurrent Versions System , 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.
- **Subversion:** Apache Subversion es una **herramienta de control de versiones de código abierto basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros**. Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Entre dos revisiones sólo guarda el conjunto de modificaciones (delta), optimizando así al máximo el uso de espacio en disco. SVN permite al usuario crear, copiar y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local.

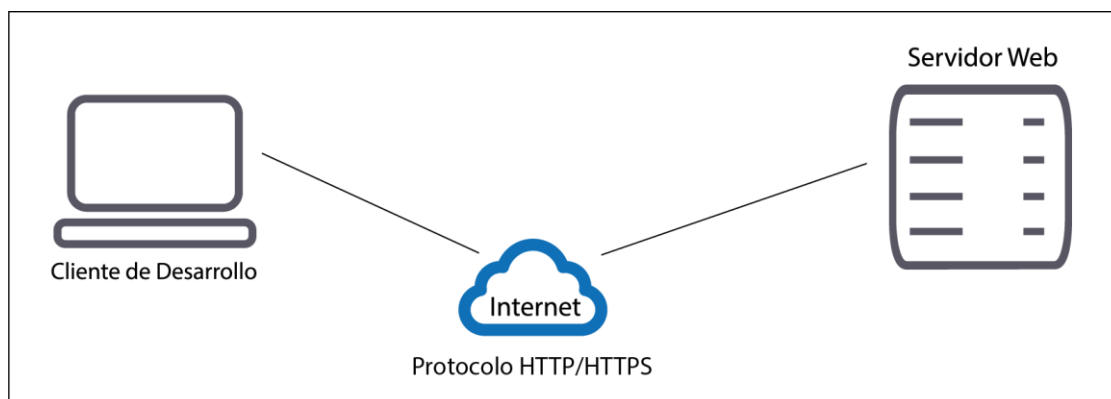
18. Propuesta de configuración del entorno desarrollo para la asignatura de Desarrollo web del lado servidor en este curso. (NAJ)

NAJ-W7ED	NAJ-USED
<p>CARACTERISTICAS</p> <p>Sistema Operativo windows 7 professional 64 bits</p> <p>Memoria RAM 4 GB</p> <p><u>Particiones</u></p> <p>CAPACIDAD TIPO</p> <p>150 GB NTFS</p> <p>350 GB NTFS</p> <p>4 GB</p> <p><u>Nombre de la máquina NAJ-W7ED</u></p> <p>Usuarios y contraseñas</p> <p>miadmin / paso (administrador del sistema)</p> <p>usuario / paso (usuario maquina)</p> <p>Configuración de red</p> <p>IP 192.168.3.XXX</p> <p>GATEWAY 192.168.3.1</p> <p>NETMASK 255.255.255.0</p> <p>DNS 192.168.20.20 / 8.8.8.8</p> <p>RED 192.168.3.0/24</p>	<p>CARACTERISTICAS</p> <p>Sistema Operativo Ubuntu Server 18.04.2 LTS</p> <p>Memoria RAM 2 GB</p> <p><u>Particiones</u></p> <p>CAPACIDAD 150 GB</p> <p>TIPO ext4</p> <p>RUTA DE MONTAJE /</p> <p>CAPACIDAD 350 GB</p> <p>TIPO ext4</p> <p>RUTA DE MONTAJE /var</p> <p><u>Nombre de la máquina NAJ-USED</u></p> <p>Usuarios y contraseñas</p> <p>miadmin / paso (administrador de sistema)</p> <p>operadorweb / paso (administrador sitios web)</p> <p>adminisDB / paso (administrador BBDD)</p> <p>Configuración de red</p> <p>IP 192.168.3.XXX</p> <p>GATEWAY 192.168.3.1</p> <p>NETMASK 255.255.255.0</p> <p>DNS 192.168.20.20 / 8.8.8.8</p> <p>RED 192.168.3.0/24</p>
<p>SOFTWARE USADO</p> <p>IDE Netbeans 12</p> <p>Navegador Chrome 84.0.4147.94</p> <p>CientesFTP Filezilla 3.37</p> <p>Cliente SSH Putty 0.74</p> <p>Herramientas Multimedia Gimp 2.10</p> <p>Ofimatica Libre Office 6.3.2</p> <p>Software PhpDoc</p> <p>MySQL Workbench</p>	<p>SOFTWARE USADO</p> <p>Sistema operativo Ubuntu Server 18.04.3 LTS</p> <p>Servidor administración remota SSH 7.6</p> <p>Servidor de transferencia de ficheros SFTP (SSH)</p> <p>Servidor Web Apache 2.5</p> <p>SGBD MySQL 8,0</p> <p>Módulos PHP 7.4</p>



19. Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado del servidor en este curso. (NAJ)

NAJ-W7EE	NAJ-USEE
CARACTERISTICAS	CARACTERISTICAS
<p>Sistema Operativo windows 7 professional 64 bits</p> <p>Memoria RAM 4 GB</p> <p><u>Particiones</u></p> <p>CAPACIDAD TIPO</p> <p>150 GB NTFS</p> <p>350 GB NTFS</p> <p>4 GB</p> <p><u>Nombre de la máquina NAJ-W7EE</u></p> <p>Usuarios y contraseñas</p> <p>miadmin / paso (administrador del sistema)</p> <p>usuario / paso (usuario maquina)</p> <p>Configuración de red</p> <p>IP 192.168.3.XXX</p> <p>GATEWAY 192.168.3.1</p> <p>NETMASK 255.255.255.0</p> <p>DNS 192.168.20.20 / 8.8.8.8</p> <p>RED 192.168.3.0/24</p>	<p>Sistema Operativo Ubuntu Server 18.04.2 LTS</p> <p>Memoria RAM 2 GB</p> <p><u>Particiones</u></p> <p>CAPACIDAD 150 GB</p> <p>TIPO ext4</p> <p>RUTA DE MONTAJE /</p> <p>CAPACIDAD 350 GB</p> <p>TIPO ext4</p> <p>RUTA DE MONTAJE /var</p> <p><u>Nombre de la máquina NAJ-USEE</u></p> <p>Usuarios y contraseñas</p> <p>miadmin / paso (administrador de sistema)</p> <p>operadorweb / paso (administrador sitios web)</p> <p>adminisDB / paso (administrador BBDD)</p> <p>Configuración de red</p> <p>IP 192.168.3.XXX</p> <p>GATEWAY 192.168.3.1</p> <p>NETMASK 255.255.255.0</p> <p>DNS 192.168.20.20 / 8.8.8.8</p> <p>RED 192.168.3.0/24</p>
SOFTWARE USADO	SOFTWARE USADO
<p>Navegador Chrome 84.0.4147.94</p> <p>Cookies</p> <p>JavaScript</p>	<p>Sistema operativo Ubuntu Server 20.04.1 LTS</p> <p>Servidor administración remota SSH 7.6</p> <p>Servidor de transferencia de ficheros SFTP (SSH)</p> <p>Servidor Web Apache 2.5</p> <p>SGBD MySQL 8,0</p> <p>Módulos PHP 7.4</p>



20. Realiza un estudio de los siguientes conceptos y relación con el desarrollo de aplicaciones web.

CMS – Sistema de gestión de contenidos

Es un software que ayuda a los usuarios a crear, administrar y modificar contenido en un sitio web sin la necesidad de conocimientos técnicos especializados.

Se compone de dos partes principales:

Una Aplicación de Gestión de Contenido (CMA) – esta es la parte que le permite agregar y administrar el contenido de su sitio (como se visualiza arriba).

Una Aplicación de Entrega de Contenido (CDA) – este es el proceso de backend y detrás de escena que toma el contenido que ingresa en el CMA, lo almacena correctamente y lo hace visible para sus visitantes.

Ejemplos:

WordPress

Joomla

Drupal

Magento (for eCommerce stores)

Squarespace

Wix

TYPO3

ERP – Sistema de planificación de los recursos empresariales

Sistema de planificación de recursos empresariales. Estos programas se hacen cargo de distintas operaciones internas de una empresa, desde producción a distribución o incluso recursos humanos.

Ejemplos:

SAP

SAGE

MICROSOFT

ORACLE

21. Elegir y realizar un estudio sobre una de las arquitecturas de Desarrollo de Aplicaciones Web.

MEAN (con MongoDB y con MySQL)

MEAN Stack (acrónimo para: MongoDB, ExpressJS, AngularJS, NodeJS), es un framework o conjunto de subsistemas de software para el desarrollo de aplicaciones, y páginas web dinámicas, que están basadas, cada una de estas en el popular lenguaje de programación conocido como JavaScript. Gracias a esta característica el conjunto se integra exitosamente en una plataforma auto-suficiente.

Componentes

MongoDB: es un sistema de base de datos NoSQL, que almacena los datos en estructuras o “documentos”, los cuales están definidos con la notación JSON, lo que permite una rápida manipulación y transferencia de los datos. La mayor característica de esta plataforma es su escalabilidad, lo que significa que puede aumentar en forma considerable la cantidad de datos que almacena sin que esto afecte su funcionamiento en general.

ExpressJS: es un módulo de NodeJS y como tal funciona sobre esta plataforma; este módulo ofrece los métodos suficientes en JavaScript, para poder manejar las solicitudes o peticiones que se hacen por medio de los métodos del **protocolo HTTP** (GET, POST, etc.). También ofrece un sistema simple de enrutamiento, que dentro del mean stack es aprovechado en el **back-end** o en el lado del servidor.

AngularJS: es un framework que facilita la manipulación del DOM y por lo tanto en el mean stack es la plataforma que se usa para trabajar en el **front-end**. Este framework permite crear una gran variedad de efectos, de una forma sencilla, reduciendo contundentemente la cantidad de código, lo que permite que sea mucho más sencillo de mantener.

Angular también tiene la capacidad de hacer el enrutamiento del sistema y claro está también incorpora la función de ajax (asincronico javascript y xml), para hacer peticiones al servidor sin tener que cargar toda la página.

NodeJS: (es la base de todo el conjunto) Es la plataforma encargada del funcionamiento del servidor. Funciona totalmente con JavaScript, un lenguaje de programación que en un principio era dedicado a correr en **el lado del cliente**, pero su uso se ha ampliado considerablemente en todos los aspectos de un sitio web.

Java EE vs Spring

Java Platform, Enterprise Edition o Java EE, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación.

Java EE tiene varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, servlets, JavaServer Pages y varias tecnologías de servicios web. Ello permite al desarrollador crear una aplicación de empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar

transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento del bajo nivel.

Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.2

El framework fue lanzado inicialmente bajo la licencia Apache 2.0 en junio de 2003

Si bien las características fundamentales de Spring Framework pueden ser usadas en cualquier aplicación desarrollada en Java, existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. A pesar de que no impone ningún modelo de programación en particular, este framework se ha vuelto popular en la comunidad al ser considerado una alternativa, sustituto, e incluso un complemento al modelo EJB (Enterprise JavaBean).

Microsoft .NET

NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones

Los principales componentes del marco de trabajo son:

- El conjunto de lenguajes de programación.
- La biblioteca de clases base o BCL.
- El entorno común de ejecución para lenguajes.

.NET soporta ya más de 20 lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos, lo que elimina las diferencias que existían entre lo que era posible hacer con uno u otro lenguaje.

Algunos de los lenguajes desarrollados para el marco de trabajo .NET son: C#, Visual Basic .NET, Delphi (Object Pascal), C++, F#, J#, Perl, Python, Fortran, Prolog (existen al menos dos implementaciones, el P#1 y el Prolog.NET2), Cobol y PowerBuilder.

Características

Es el encargado de proveer lo que se llama código administrado, es decir, un entorno que provee servicios automáticos al código que se ejecuta. Los servicios son variados:

- Cargador de clases: permite cargar en memoria las clases.
- Compilador MSIL a nativo: transforma código intermedio de alto nivel independiente del hardware que lo ejecuta a código de máquina propio del dispositivo que lo ejecuta.
- Administrador de código: coordina toda la operación de los distintos subsistemas del Common Language Runtime.
- Recolector de basura: elimina automáticamente de memoria objetos no utilizados.
- Motor de seguridad: administra la seguridad del código que se ejecuta.
- Motor de depuración: permite hacer un seguimiento de la ejecución del código.
- Verificador de tipos: controla que las variables de la aplicación usen el área de memoria que tienen asignado.
- Administrador de excepciones: maneja los errores que se producen durante la ejecución del código.
- Soporte de multiproceso (hilos): permite desarrollar aplicaciones que ejecuten código en forma paralela.
- Biblioteca de Clases Base que incluye soporte para muchas funcionalidades comunes en las aplicaciones.

Angular 7

Angular un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript.

Angular se basa en clases tipo "Componentes", cuyas propiedades son las usadas para hacer el binding de los datos. En dichas clases tenemos propiedades (variables) y métodos (funciones a llamar).

Características

Velocidad y rendimiento

- Generación de código: Angular convierte tus plantillas en código altamente optimizado.
- Universal: Ejecuta la primera vista de tu aplicación en node.js, .NET, PHP, y otros servidores para renderizado de forma casi instantánea obteniendo solo HTML y CSS.
- División del código: Las aplicaciones de Angular se cargan rápidamente gracias al nuevo enrutador de componentes. Éste ofrece una división automática de códigos para que los usuarios sólo carguen el código necesario para procesar la vista que solicitan.

Productividad

- Plantillas: Permite crear rápidamente vistas de interfaz de usuario.
- Angular CLI: Las herramientas de línea de comandos permiten empezar a desarrollar rápidamente, añadir componentes y realizar test, así como previsualizar de forma instantánea la aplicación.
- IDEs: Obtén sugerencias de código inteligente, detección de errores y otros comentarios en la mayoría de los editores populares e IDEs.

Historia completa del desarrollo

- Testing: Utiliza Karma para realizar pruebas unitarias, y Protractor para realizar pruebas end-to-end de forma rápida y estable.
- Animación: Permite crear animaciones complejas y de alto rendimiento con muy poco código a través de la intuitiva API de Angular.
- Accesibilidad: Posee características para crear aplicaciones accesibles con los componentes disponibles para ARIA.

Symfony

Symfony es un framework diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente en PHP 5.3. Es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Características

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de ORM (es una técnica de programación para convertir datos entre el sistema de tipos (clasifica los valores y las expresiones en tipos) utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia.) permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos y características como los espacios de nombres.
- Sencillo de usar en la mayoría de los casos, preferible para grandes aplicaciones web.
- Aunque utiliza Modelo Vista Controlador (MVC), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una línea de comandos que facilita la generación de código.

Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.2

Características

Sistema de ruteo, también RESTful

Blade, Motor de plantillas

Peticiones Fluent

Eloquent ORM

Basado en Composer

Soporte para el caché

Soporte para MVC

Usa componentes de Symfony

Adopta las especificaciones PSR-2 y PSR-4

CakePHP

CakePHP es un framework o marco de trabajo que "facilita" el desarrollo de aplicaciones web, escrito en PHP, creado sobre los conceptos de Ruby on Rails. Al igual que Ruby On Rails, CakePHP facilita al usuario la interacción con la base de datos mediante el uso de Active record.

Utiliza el patrón de diseño MVC (Modelo Vista Controlador). Es de código abierto y se distribuye bajo licencia MIT.

Características

- Compatible con PHP4 y PHP5
- CRUD de la base de datos integrado
- URLs amigables
- Sistema de plantillas rápido y flexible
- Ayudas para AJAX, Javascript, HTML, forms y más
- Trabaja en cualquier subdirectorio del sitio
- Validación integrada
- Scaffolding de las aplicaciones
- Lista de control de acceso
- Sintetización de datos
- Componentes de seguridad y sesión

CodeIgniter

CodeIgniter es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. «Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas. »

También hay que destacar que CodeIgniter es más rápido que muchos otros entornos.

Kohana ha creado una rama a partir de CodeIgniter 1.5.4 en 2007, como resultado del disgusto de la comunidad por la falta de corrección de errores y el deseo de incorporar otras características demandadas por los usuarios.

El día 1 de Marzo se publica la primera beta de la versión 4 de CodeIgniter. Hasta la fecha la más actual en producción es la versión 3.1.10 que se puede descargar desde CodeIgniter

GLOSARIO

Criptografía: se ocupa de las técnicas de cifrado o codificado destinadas a alterar las representaciones lingüísticas de ciertos mensajes con el fin de hacerlos ininteligibles a receptores no autorizados.

Ataque de intermediario: también conocido como *ataque Janus*, es un ataque en el que se adquiere la capacidad de leer, insertar y modificar a voluntad.

Eavesdropping: término inglés que traducido al español significa escuchar secretamente, es el acto de escuchar en secreto o sigilosamente conversaciones privadas o comunicaciones de otros sin su consentimiento.

Metadatos: datos que describen otros datos. En general, un grupo de metadatos se refiere a un grupo de datos que describen el contenido informativo de un objeto al que se denomina recurso.

Proxy: hace de intermediario en las peticiones de recursos que realiza un cliente a otro servidor.

Framework: entorno de trabajo o marco de trabajo, es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Red: es un conjunto de clientes, servidores y base de datos unidos de una manera física o no física en el que existen protocolos de transmisión de información establecidos.

Cliente: hace referencia a un demandante de servicios, este cliente puede ser un ordenador como también una aplicación de informática, la cual requiere información proveniente de la red para funcionar.

Servidor: hace referencia a un proveedor de servicios, este servidor a su vez puede ser un ordenador o una aplicación informática la cual envía información a los demás agentes de la red.

Protocolo: es un conjunto de normas o reglas y pasos establecidos de manera clara y concreta sobre el flujo de información en una red estructurada.

Servicios: es un conjunto de información que busca responder las necesidades de un cliente, donde esta información pueden ser mail, música, mensajes simples entre software, videos, etc.

Base de datos: bancos de información ordenada, categorizada y clasificada que forman parte de la red, que son sitios de almacenaje para la utilización de los servidores y también directamente de los clientes.

Una **API** es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. **API significa** interfaz de programación de aplicaciones.

BIBLIOGRAFIA

Pregunta 3

<https://einatec.com/tipos-de-aplicaciones-web/>

Pregunta 5

<https://nestrategia.com/desarrollo-web-back-end-front-end/>

Pregunta 6

<https://einatec.com/tipos-de-aplicaciones-web/>

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-una-web-app-y-que-clases-hay/>

<https://sg.com.mx/content/view/256>

<https://www.arimetrics.com/glosario-digital/mashup-aplicacion-web-hibrida>

Pregunta 15

<https://programadorwebvalencia.com/cual-es-el-mejor-navegador-web-2020/>