

UD 1 Implantación de arquitecturas web.

1.- Aspectos generales de arquitecturas web.

La arquitectura World Wide Web ([WWW](#)) de Internet provee un modelo de programación sumamente poderoso y flexible. Las aplicaciones y los contenidos son presentados en formatos de datos estándar y son localizados por aplicaciones conocidas como "web browsers", que envían requerimientos de objetos a un servidor y éste responde con el dato codificado según un formato estándar.

Los estándares WWW especifican muchos de los mecanismos necesarios para construir un ambiente de aplicación de propósito general, por ejemplo:

- **Modelo estándar de nombres:** todos los servidores, así como el contenido de la WWW se denominan según un Localizador Uniforme de Recursos (Uniform Resource Locator: [URL](#)).
- **Contenido:** a todos los contenidos en la WWW se les especifica un determinado tipo permitiendo de esta forma que los [browsers](#) (navegadores) los interpreten correctamente.
- **Formatos de contenidos estándar:** todos los navegadores soportan un conjunto de formatos estándar, por ejemplo [HTML](#), [ECMA Script](#), [JavaScript](#), etc.
- **Protocolos estándar:** éstos permiten que cualquier navegador pueda comunicarse con cualquier servidor web. El más comúnmente usado en WWW es [HTML](#) (Protocolo de Transporte de HiperTexto), que opera sobre el conjunto de protocolos [TCP/IP](#).

Esta infraestructura permite a los usuarios acceder a una gran cantidad de aplicaciones y servicios de terceros. También permite a los desarrolladores crear aplicaciones y servicios para una gran comunidad de clientes.

Los aspectos generales a destacar en una arquitectura web son los siguientes:

- Escalabilidad.
- Separación de responsabilidades.
- Portabilidad.
- Utilización de componentes en los servicios de infraestructura.
- Gestión de las sesiones del usuario.
- Aplicación de patrones de diseño.



El esquema de funcionamiento de los servicios web requiere de tres elementos fundamentales:

1. **Proveedor del servicio web**, que es quien lo diseña, desarrolla e implementa y lo pone disponible para su uso, ya sea dentro de la misma organización o en público.
2. **Consumidor del servicio**, que es quien accede al componente para utilizar los servicios que éste presta.
3. **Agente del servicio**, que sirve como enlace entre proveedor y consumidor para efectos de publicación, búsqueda y localización del servicio.

De forma genérica podríamos decir que la arquitectura web es un modelo compuesto de tres niveles:

1. En primer nivel estarían los **clientes del servicio web** al que accederían mediante un navegador web como Chrome, Firefox, Edge, Opera, etc....
2. En una segunda nivel estarían los **servidores de aplicaciones web**, ejecutando aplicaciones de tipo Apache, IIS, Tomcat, Resin, etc.
3. En un tercer nivel **de Base de Datos**, donde estaría toda la documentación de la información que se pretende administrar mediante el servicio web y emplearía una plataforma del tipo MariaDB, PostgreSQL, MySQL, SQL Server, Oracle, DB/2, MongoDB etc.

NIVEL 1



Arquitectura a 3 niveles

Actividad 1.1

Buscar en internet Otros navegadores, Servidores Web y Bases de datos

1.1 Protocolos

Un **protocolo** es un método estándar que permite la comunicación entre procesos (que potencialmente se ejecutan en diferentes equipos), es decir, es un conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos a través de una red. Existen diversos protocolos de acuerdo a cómo se espera que sea la comunicación. Algunos protocolos, por ejemplo, se especializarán en el intercambio de archivos, como el FTP (File Transfer Protocol, Protocolo de transferencia de ficheros); otros pueden utilizarse simplemente para administrar el estado de la transmisión y los errores (como es el caso de ICMP), etc.

En Internet, los protocolos utilizados pertenecen a una sucesión de protocolos o a un conjunto de protocolos relacionados entre sí. Este conjunto de protocolos se denomina TCP/IP. Entre otros, contiene los siguientes protocolos:

1.1.1.- El protocolo DNS

El sistema de nombres de dominio (Domain Name System o **DNS**, por sus siglas en inglés) Es un sistema de nomenclatura jerárquico descentralizado para dispositivos conectados a redes IP como Internet o una red privada.

El servidor DNS utiliza una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP.

1.1.2.- El protocolo HTTP

Desde 1990, el protocolo [HTTP](#) (Hiper Text Transfer Protocol, Protocolo de transferencia de hipertexto) es el protocolo más utilizado en Internet. La versión 0.9 solo tenía la finalidad de transferir los datos a través de Internet (en particular páginas web escritas en HTML). La versión 1.0 del protocolo (la más utilizada) permite la transferencia de mensajes con encabezados que describen el contenido de los mensajes mediante la codificación MIME.

El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato HTML) entre un navegador (el cliente) y un servidor web localizado mediante una cadena de caracteres denominada dirección URL (Uniform Resource Locator, localizador uniforme de recursos).

La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:

- El navegador realiza una solicitud HTTP.
- El servidor procesa la solicitud y después envía una respuesta HTTP

1.1.3.- El protocolo HTTPS

El protocolo seguro de Transferencia de hipertexto ([HTTPS](#), Hiper Text Transfer Protocol Secure) es la versión segura del protocolo [HTTP](#). La diferencia es que [HTTPS](#) permite realizar transacciones de forma segura. Por lo tanto, podremos desarrollar actividades de tipo e-commerce, acceso a cuentas bancarias on line, trámites con la administración pública, etc.

En los navegadores comunes como Firefox, Edge o Chrome, cuando estamos empleando un protocolo HTTPS podemos ver el icono de un candado que aparece en la barra principal de nuestro navegador. Además, en la barra de direcciones podremos ver que “<http://>” será sustituido por “<https://>”.

Y, ¿cómo funciona la conexión exactamente? ¿Por qué es más segura? Básicamente, lo que ocurre es que la página web codifica la sesión con certificado digital. De este modo, el usuario tiene ciertas garantías de que la información que envíe desde dicha página no podrá ser interceptada y utilizada por terceros.

Estos certificados de seguridad son conocidos como [SSL](#). Cuando estos están instalados en la página web veremos el candado del que hablábamos anteriormente. Por otro lado, si hay instalados Certificados de Validación Extendida, además del candado, los usuarios podremos ver que la barra de URL del navegador toma un fondo verdoso.

1.1.4.- El protocolo FTP

El protocolo [FTP](#) (File Transfer Protocol, Protocolo de transferencia de archivos) es, como su propio nombre indica, un protocolo para transferir archivos.

La implementación del [FTP](#) se remonta a 1971, cuando se desarrolló un sistema de transferencia de archivos (descrito en RFC141) entre equipos del Instituto Tecnológico de Massachusetts (MIT, Massachusetts Institute of Technology). Desde entonces, diversos documentos de RFC han mejorado el protocolo básico, pero las innovaciones más importantes se llevaron a cabo en julio de 1973.

El protocolo FTP define la manera en que los datos deben ser transferidos a través de una red TCP/IP. El objetivo del protocolo FTP es:

- Permitir que equipos remotos puedan compartir archivos.
- Permitir la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor.
- Permitir una transferencia de datos eficaz

1.1.4.- El protocolo SMTP

El protocolo [SMTP](#) (Simple Mail Transfer Protocol, Protocolo simple de transferencia de correo) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores. La comunicación entre el cliente y el servidor consiste enteramente en líneas de texto compuestas por caracteres ASCII. El tamaño máximo permitido para estas líneas es de 1.000 caracteres.

Actividad 1.2

Explica con tus palabras los protocolos ARP, UDP, TCP, TFTP, POP y NTP

1.2.- Evolución de los servicios web.

La evolución del uso de Servicios web en las organizaciones está fuertemente ligada al desarrollo de Internet como red prestadora de servicios. Entre los factores que han impulsado el uso de los servicios web se encuentran:

- ✓ **El contenido se está volviendo más dinámico:** Los sitios web actuales proporcionan contenidos "instantáneos". Un Servicio web debe ser capaz de combinar contenido proveniente de fuentes muy diferentes.
- ✓ **El ancho de banda es menos costoso:** Actualmente un Servicio web puede entregar tipos variables de contenidos como vídeo o audio. A medida que crezca el ancho de banda, los servicios web deben adaptarse a nuevos tipos de contenidos.
- ✓ **El almacenamiento es más barato:** Un Servicio web debe ser capaz de manejar cantidades masivas de datos, y debe poder hacerlo de forma inteligente.
- ✓ **El éxito de la computación extendida se está volviendo más importante:** Con cientos de millones de dispositivos como teléfonos móviles, agendas electrónicas, etc. existentes actualmente, estamos llegando a un momento en el cual las computadoras están dejando de ser el dispositivo más común en Internet. A medida que las plataformas se hacen más diversas, tecnologías como XML se volverán más importantes. Un servicio web no puede exigir que los usuarios ejecuten, por ejemplo, un navegador web tradicional en alguna versión de Microsoft Windows; por el contrario, los servicios web deben servir a todo tipo de dispositivos, plataformas y navegadores, entregando contenido sobre una amplia variedad de tipos de conexión.

Estos factores, unidos a los beneficios proporcionados por los servicios web en la organización y los buenos productos disponibles para su desarrollo, han hecho que su utilización se extienda sin mayores obstáculos.

En términos generales, cuando se empiezan a utilizar servicios web en una organización, estos se desarrollan e implementan como servicios simples, que poco a poco se van integrando hasta llegar a servicios web mucho más complejos.

En los orígenes del mundo web nos situábamos ante un entorno estático, con páginas en formato HTML que raramente sufrían modificaciones o actualizaciones y en las que apenas había interacción con el usuario.

La **Web 2.0** es la transición que se ha dado desde las aplicaciones tradicionales hacia aplicaciones que funcionan a través de la web y que están fuertemente enfocadas al usuario final. En este nuevo entorno existen una serie de nuevas tecnologías que, en general, tienen como objetivo:

- ✓ Transformar software de escritorio hacia la web.
- ✓ Separar hojas de estilo.
- ✓ Potenciar el trabajo colaborativo y la utilización de redes sociales.
- ✓ Dar control total a los usuarios en el manejo de su información.

1.3.- Tecnologías asociadas a las aplicaciones web.

Las aplicaciones web emplean páginas dinámicas, éstas se ejecutan en un servidor web y se muestran en el navegador de un equipo cliente que es el que ha realizado previamente la solicitud. Cuando una página web llega al navegador, es posible que también incluya algún programa o fragmento de código que se deba ejecutar. Ese código, normalmente en lenguaje JavaScript, **lo ejecutará el propio navegador**. Es por ello que en este apartado nos centraremos en las tecnologías asociadas a las aplicaciones web que se ejecutarán tanto del lado del servidor como del cliente, especificando lo que corresponda en cada uno de los casos.

✓ **Perl** es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado bourne shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

✓ **ASP (Active Server Pages)**: Las "Páginas Activas" se ejecutan del lado del servidor, de este modo se forman los resultados que luego se mostrarán en el navegador de cada equipo cliente que ha realizado la solicitud. Un buen ejemplo de ello son los buscadores, donde un usuario realiza una petición de información y el servidor nos entrega un resultado a medida de nuestra petición.

Existen versiones de ASP para Unix y Linux, a pesar de que fue una tecnología desarrollada por Microsoft para la creación dinámica de páginas web ofrecida junto a su servidor IIS.

✓ **CGI (Common Gateway Interface)**: La "Interface Común de Entrada" es uno de los estándares más antiguos en Internet para trasladar información desde una página a un servidor web. **Este estándar** es utilizado para bases de datos, motores de búsqueda, formularios, generadores de email automático, foros, comercio electrónico, rotadores y mapas de imágenes, juegos en línea, etc.

Las rutinas de CGI son habitualmente escritas en lenguajes interpretados como Perl o por lenguajes compilados como C.

✓ **CSS (Cascading Style Sheets)**: Las "Hojas de Estilo en Cascada" se usan para formatear las páginas web; se trata de separar el contenido de un documento de su presentación. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS.

✓ **Java**: Este es un lenguaje que trabaja en el cliente, es decir: se ejecuta en el navegador del equipo cliente y no en el servidor. Es un lenguaje eficiente y muy poderoso, que se caracteriza por:

➔ Una misma aplicación puede funcionar en diversos tipos de ordenadores y sistemas operativos: Windows, Linux, Solaris, MacOS, etc., así como en otros dispositivos inteligentes.

➔ Los programas Java pueden ser aplicaciones independientes (que corren en una ventana propia) o "applets", que son pequeños programas interactivos que se encuentran incrustados en una página web y pueden funcionar con cualquier tipo de navegador: Chrome, Firefox, Ópera, etc.

➔ Se trata de un lenguaje "orientado a objetos". Esto significa que los programas se construyen a partir de módulos independientes, y que estos módulos se pueden transformar o ampliar fácilmente. Un equipo de programadores puede partir de una aplicación existente para extenderla con nuevas funcionalidades.

➔ Desarrollado por la empresa Sun Microsystems, pero posteriormente liberado bajo licencia GNU GPL (La Licencia Pública General de GNU, o más conocida por su nombre en inglés GNU General Public License es una licencia creada por la "Free Software Foundation" y está orientada, principalmente, a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. El proyecto GNU (GNU es un acrónimo recursivo para "GNU No es Unix"). Comenzó en 1984 a desarrollar un sistema operativo completo con la principal propiedad de ser Software Libre), con lo cual es un software libre.

✓ **JavaScript**: Lenguaje que se interpreta y se ejecuta en el cliente. Útil para realizar tareas como mover imágenes por la pantalla, crear menús de navegación interactivos, utilizar algunos juegos, etc. En las páginas web suele preferirse JavaScript porque es aceptado por muchos más navegadores que VBScript (creado por Microsoft)

✓ **PHP** (Hypertext Preprocessor): Este lenguaje es, como ASP, ejecutado en el lado del servidor. PHP es similar a ASP y puede ser usado en circunstancias similares. Es muy eficiente, permitiendo el acceso a bases de datos empleando servidores como MySQL (potente gestor de bases de datos relacional, sencillo de usar e increíblemente rápido. También es uno de los motores de bases de datos más usados en Internet, la principal razón de esto es que se distribuye bajo la licencia GNU GPL para aplicaciones no comerciales) y, por lo tanto, suele utilizarse para crear páginas dinámicas complejas.

✓ **VBScript** (Visual Basic Scripting): La respuesta de Microsoft a JavaScript. VBScript es una buena herramienta para cualquier sitio destinado a ser mostrado exclusivamente en el navegador Microsoft Internet Explorer. El código en VBScript puede, además, estar diseñado para su ejecución en el lado del cliente o en el del servidor, la diferencia es que un código que se ejecuta en el lado del servidor no es visible en el lado del cliente. Éste recibe los resultados, pero no el código.

✓ **Node.js** es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor [V8 de Google](#).

Actividad 1.3

Busca otras tecnología del lado del servidor y del cliente.

1.4- Tipos de aplicaciones web.

Cualquier proyecto que se quiera desarrollar en Internet, bien sea comercio electrónico, reservas de billetes de vuelo on-line, información meteorológica, registro de usuarios, simuladores de hipotecas, etc, conlleva el desarrollo de una aplicación web. En definitiva, una aplicación web es una plataforma orientada a automatizar los procesos de servicios que se quieran ofrecer a usuarios. Establecer una clasificación de los tipos de aplicaciones web es una tarea compleja debido a la dificultad existente para poder establecer algún parámetro en función del cual establecer dicha clasificación, junto con la innumerable cantidad de aplicaciones existentes en el actual entorno [Web 2.0](#).



En función de cómo se presenta la aplicación web junto con el contenido que pretende mostrar, se ha establecido la siguiente clasificación:

- ✓ **Página web Estática.** Están implementadas en HTML y pueden mostrar en alguna parte de la página objetos en movimiento tales como banners, GIF animados, vídeos, etc.
- ✓ **Página web Animada.** Se realizan con la tecnología FLASH; ésta permite que una página web presente el contenido con ciertos efectos animados continuados. El uso de esta tecnología permite diseños más vanguardistas, modernos y creativos. La tendencia actual son las animaciones con HTML5.
- ✓ **Página web Dinámica.** Existen muchos lenguajes de programación que son la base para la mayoría de páginas web dinámicas. Los que destacamos aquí son los lenguajes PHP, ASP.NET, [Python](#) etc. Estos lenguajes permiten una perfecta estructuración del contenido. Por una parte crearíamos la estructura de las páginas web y por otra, almacenaríamos el contenido en determinados archivos. A partir de ahí, crearíamos el código de llamada, que insertaría el contenido en la propia página web estructurada. Este es el principio básico que siguen los lenguajes de programación. A partir de aquí se desarrollan aplicaciones para poder gestionar el contenido a través de un panel de control.
- ✓ **Portal.** Es un sitio web que en su página principal permite el acceso a múltiples secciones que, por lo general, son foros, chats, cuentas de correo, buscador, acceso registrado para obtener ciertas ventajas, las últimas noticias de actualidad, etc.
- ✓ **Tienda virtual o comercio electrónico.** Sitio web que publica los productos de una tienda en Internet. Permite la compra on-line a través de tarjeta de crédito, domiciliación bancaria o transferencia bancaria en general. Ofrece al administrador un panel de gestión para poder subir los productos, actualizarlos, eliminarlos, etc.
- ✓ **Página web con "Gestor de Contenidos".** Se trata de un sitio web cuyo contenido se actualiza a través de un panel de gestión por parte del administrador del sitio. Este panel de gestión suele ser muy intuitivo y fácil de usar. En aquellas páginas web que requieran una actualización constante, se suele incorporar este panel de gestión para que la web pueda controlarse día a día por parte del cliente.

Actividad 1.4

Busca que es la WEB 3.0 y WEB 4.0

1.5.- Arquitecturas web. Modelos.

Se puede establecer que la arquitectura de un sitio web comprende los sistemas de organización y estructuración de los contenidos junto con los sistemas de recuperación de información y navegación que provea el sitio web, con el objetivo de servir de ayuda a los usuarios a encontrar y manejar la información.

Centraremos el estudio de los modelos de arquitectura web relacionados, en función de cómo implementan cada una de las capas establecidas en una aplicación web:

1. **Capa de presentación** es la encargada de la navegabilidad, validación de los datos de entrada, formateo de los datos de salida, presentación de la web, etc.; se trata de la capa que se presenta al usuario.
2. **Capa de negocio** es la que recibe las peticiones del usuario y desde donde se le envían las respuestas; en esta capa se verifican que las reglas establecidas se cumplen.
3. **Capa de acceso a datos** es la formada por determinados gestores de datos que se encargan de almacenar, estructurar y recuperar los datos solicitados por la capa de negocio.

La evolución experimentada por los medios informáticos en los últimos años ha convivido con otra evolución paralela, la evolución de la arquitectura de las aplicaciones web, que permite aprovechar las nuevas características que éstas ofrecen. De esta forma, el modelo arquitectónico de las aplicaciones de Internet ha sufrido dos grandes saltos, con algún paso intermedio, desde la aparición de los primeros portales web. Los distintos modelos de aplicación sobre los que se ha ido desarrollando, según diversos autores, se podrían clasificar del siguiente modo:

✓ **Modelo 1**

En este caso las aplicaciones se diseñan en un modelo web CGI, basadas en la ejecución de procesos externos al servidor web, cuya salida por pantalla era el HTML que el navegador recibía en respuesta a su

petición. Presentación, negocio y acceso a datos se confundían en un mismo script perl (Lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado shell (sh), awk, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación).

✓ **Modelo 1.5**

Aplicado a la tecnología java (Lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90, aunque a finales de 2006 liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL), se da con la aparición de las JSP y los servlets (Objetos que se ejecutan dentro del contexto de un contenedor de "servlets", por ejemplo Tomcat y amplían su funcionalidad. La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas que se ejecutan en el contexto de un navegador web. Por contraposición, un servlet es un programa que se ejecuta en un servidor. El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web). En este modelo, las responsabilidades de presentación recaen en las páginas JSP, mientras que los beans (Abreviatura científica del botánico Willian Jackson Bean (18631947). Un bean es un componente software que tiene la particularidad de ser reutilizable y así evitar la tediosa tarea de programar los distintos componentes uno a uno) incrustados en las mismas son los responsables del modelo de negocio y acceso a datos.

✓ **Modelo 2**

Como evolución del modelo anterior, con la incorporación del patrón MVC en este tipo de aplicaciones, se aprecia la incorporación de un elemento controlador de la navegación de la aplicación. El modelo de negocio queda encapsulado en los javabeans (Modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java; se usan para encapsular varios objetos en un único objeto (bean), para hacer uso de un solo objeto en lugar de varios más simples. La especificación de JavaBeans los define como "componentes de software reutilizables que se puedan manipular visualmente en una herramienta de construcción") que se incrustan en las páginas JSP.

✓ **Modelo 2X**

Aparecen con el objetivo de dar respuesta a la necesidad, cada vez más habitual, de desarrollar aplicaciones multicanal, es decir, aplicaciones web que pueden ser atacadas desde distintos tipos de clientes remotos. Así, una aplicación web multicanal podrá ejecutarse desde una PDA, desde un terminal de telefonía móvil, o desde cualquier navegador HTML estándar. El medio para lograr publicar la misma aplicación para distintos dispositivos es emplear plantillas XSL para transformar los datos XML.

Actividad 1.5

Define lo que es Patrón de diseño de software y buscar y describir alguno.

1.6.- Plataformas web libres y propietarias.

Una plataforma web es el entorno de desarrollo de software empleado para diseñar y ejecutar un sitio web. En términos generales, una plataforma web consta de cuatro componentes básicos:

1. El **sistema operativo**, bajo el cual opera el equipo donde se hospedan las páginas web y que representa la base misma del funcionamiento del computador. En ocasiones limita la elección de otros componentes.
2. El **servidor web** es el software que maneja las peticiones desde equipos remotos a través de la Internet. En el caso de páginas estáticas, el servidor web simplemente provee el archivo solicitado, el cual se muestra en el navegador. En el caso de sitios dinámicos, el servidor web se encarga de pasar las solicitudes a otros programas que puedan gestionarlas adecuadamente.
3. El **gestor de bases de datos** se encarga de almacenar sistemáticamente un conjunto de registros de datos relacionados para ser usados posteriormente.
4. Un **lenguaje de programación interpretado** que controla las aplicaciones de software que corren en el sitio web.

Diferentes combinaciones de los cuatro componentes señalados, basadas en las distintas opciones de software disponibles en el mercado, dan lugar a numerosas plataformas web, aunque, sin duda, hay dos que sobresalen del resto por su popularidad y difusión: **LAMP** y **WISA**.

La plataforma **LAMP** trabaja enteramente con componentes de **software libre** y no está sujeta a restricciones propietarias. El nombre **LAMP** surge de las iniciales de los componentes de software que la integran:

- ✓ **Linux**: Sistema operativo.
- ✓ **Apache**: Servidor web.
- ✓ **MySQL**: Gestor de bases de datos.
- ✓ **PHP**: Lenguaje interpretado PHP, aunque a veces se sustituye por **Perl** o **Python**.

La plataforma **WISA** está basada en tecnologías desarrolladas por la compañía Microsoft; se trata, por lo tanto, de **software propietario**. La componen los siguientes elementos:

- ✓ **Windows**: Sistema operativo.
- ✓ **Internet Information Services**: servidor web.
- ✓ **SQL Server**: gestor de bases de datos.
- ✓ **ASP o ASP.NET**: como lenguaje para scripting del lado del servidor.

Existen otras plataformas, como por ejemplo la configuración Windows-Apache-MySQL-PHP que se conoce como **WAMP**. Es bastante común pero sólo como plataforma de desarrollo local. De forma similar, un servidor Windows puede correr con MySQL y PHP. A esta configuración se la conoce como plataforma **WIMP**. Existen muchas otras plataformas que trabajan con distintos sistemas operativos (Unix, MacOS, Solaris), servidores web (incluyendo algunos que se han cobrado relativa popularidad como Lighttpd y LiteSpeed), bases de datos (PostgreSQL) y lenguajes de programación.

1.7.- Escalabilidad.

Las aplicaciones web se ejecutan en un entorno donde el número de clientes que solicitan el servicio puede variar en gran medida en función del momento. Es por ello que hay una característica de esencial importancia como es la escalabilidad, al que Juan ha dedicado un apartado de su wiki para documentar esta característica.

En el entorno en que se ubican las aplicaciones web, uno de los principales factores que puede afectar al rendimiento de las mismas es el número de usuarios, ya que éste puede verse incrementado de forma vertiginosa en un periodo de tiempo relativamente corto. El éxito o el fracaso de un sitio web orientado al usuario común vendrá determinado, entre otros aspectos, por el dimensionamiento del sistema sobre el que se instala y soporta el software que sustenta dicho sitio. En consecuencia, uno de los requisitos fundamentales de una aplicación web es que sea completamente escalable sin que un aumento de los recursos dedicados a la misma suponga modificación alguna en su comportamiento o capacidades.

La escalabilidad de un sistema web puede ser:

- ✓ Verticalmente: de manera ascendente "upgrades" a cada nodo.
- ✓ Horizontalmente: consiste en aumentar el número de nodos.
- ✓ Cluster: consiste en crear agrupaciones de servidores.

Escalabilidad vertical.

Habitualmente, la separación lógica en capas se implementa de tal forma que se permita una separación física de las mismas. Interponiendo elementos conectores que actúen de middlewares es posible distribuir la aplicación de forma vertical (una máquina por cada capa del sistema), e incluso si esto no fuera suficiente, distribuyendo los elementos de una misma capa entre distintas máquinas servidoras.

Escalabilidad horizontal.

Se trata de clonar el sistema en otra máquina de características similares y balancear la carga de trabajo mediante un dispositivo externo. El balanceador de carga puede ser:

- ✓ **Balanceador Software:** Por ejemplo, habitualmente encontramos un servidor web apache junto con el módulo mod_jk, que permite la redirección de las peticiones http que a tal efecto sean configuradas entre las distintas máquinas que forman la granja de servidores. Este tipo de balanceadores examinan el paquete http e identifican la sesión del usuario, guardando registro de cuál de las máquinas de la granja se está encargando de servir a dicha sesión. Este aspecto es importante, dado que nos permite trabajar (de cara al diseño de la aplicación) apoyándonos en el objeto sesión propio del usuario y almacenando información relativa a la sesión del mismo, puesto que tenemos la garantía de que todas las peticiones de una misma sesión http van a ser redireccionadas hacia la misma máquina.
- ✓ **Balanceador hardware:** Se trata de dispositivos que, respondiendo únicamente a algoritmos de reparto de carga (Round Robin, LRU, etc.), redireccionan una petición http del usuario a la máquina que, según dicho algoritmo, convenga que se haga cargo de la petición. Son mucho más rápidos que los anteriores, dado que se basan en conmutación de circuitos y no examinan ni interpretan el paquete http. Sin embargo, el no garantizar el mantenimiento de la misma sesión de usuario en la misma máquina, condiciona seriamente el diseño, dado que fuerza a que la información relativa a la sesión del usuario sea almacenada por el implementador del mismo, bien en cookies o bien en base de datos.
- ✓ **Balanceador hardware http:** Se trata de dispositivos hardware pero que examinan el paquete http y mantienen la relación usuario-máquina servidora. Mucho más rápidos que los balanceadores software, pero algo menos que los hardware, suponen hoy en día una de las soluciones más aceptadas en el mercado.

Cluster

Con la aparición de los servidores de aplicaciones en cluster se abre una nueva capacidad de escalabilidad que, dependiendo de cómo se aplique, podría clasificarse como vertical u horizontal. Un cluster de servidores de aplicaciones permite el despliegue de una aplicación web corriente, de forma que su carga de trabajo vaya a ser distribuida entre la granja de servidores que forman el cluster, de modo transparente al usuario y al administrador. El cluster, mediante el mecanismo de replicación de sesión, garantiza que sea cual sea la máquina que sirva la petición http, tendrá acceso a la sesión del usuario (objeto HttpSession en java). Este tipo de sistemas, debido precisamente a la replicación de sesión, suele presentar problemas de rendimiento.

2.- Servidor web Apache.

Un servidor web es un programa que se ejecuta de forma continua en un ordenador (también se utiliza el término para referirse al ordenador que lo ejecuta), se mantiene a la espera de peticiones por parte de un cliente (un navegador de Internet) y contesta a estas peticiones de forma adecuada, sirviendo una página web que será mostrada en el navegador o mostrando el mensaje correspondiente si se detectó algún error.

Uno de los servidores web más populares del mercado y el más utilizado actualmente es Apache, de código abierto y gratuito, disponible para Windows y GNU/Linux, entre otros.

En cuanto a su arquitectura podemos destacar lo siguiente:

- ✓ Estructurado en módulos.
- ✓ Cada módulo contiene un conjunto de funciones relativas a un aspecto concreto del servidor.
- ✓ El archivo binario **httpd** contiene un conjunto de módulos que han sido compilados.
- ✓ La funcionalidad de estos módulos puede ser activada o desactivada al arrancar el servidor.

- ✓ Los módulos de Apache se pueden clasificar en tres categorías:
 - ➔ **Módulos base:** Se encargan de las funciones básicas.
 - ➔ **Módulos multiproceso:** Encargados de la unión de los puertos de la máquina, aceptando las peticiones y atendiéndolas.
 - ➔ **Módulos adicionales:** se encargan de añadir funcionalidad al servidor

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. La licencia de software, bajo la cual el software de la fundación Apache es distribuido, es una parte distintiva de la historia de Apache HTTP Server y de la comunidad de código abierto.

La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original.

Esta web sirve como manual de referencia, guía de usuario, tutoriales prácticos, etc., sobre el servidor web Apache. Se trata de la web oficial de Apache Software Foundation. <http://httpd.apache.org/docs/2.0/es/>

3.- Aplicaciones web y servidores de aplicaciones

Se define una aplicación web como una aplicación informática que se ejecuta en un entorno web, de forma que se trata de una aplicación cliente-servidor junto con un protocolo de comunicación previamente establecido:

- ✓ Cliente: navegador.
- ✓ Servidor: servidor web
- ✓ Comunicación: protocolo HTTP

Un servidor de aplicaciones es un software que proporciona aplicaciones a los equipos o dispositivos cliente, por lo general, a través de Internet y utilizando el protocolo http. Los servidores de aplicación se distinguen de los servidores web en el uso extensivo del contenido dinámico y por su frecuente integración con bases de datos.

Un servidor de aplicaciones también es una máquina en una red de computadores que ejecuta determinadas aplicaciones, gestionando la mayor parte de las funciones de acceso a los datos de la aplicación.

Las principales ventajas de la tecnología de los servidores de aplicaciones es la centralización y disminución de la complejidad en el desarrollo de las aplicaciones, ya que no necesitan ser programadas, sino que son ensambladas desde bloques provistos por el servidor de aplicación.

Otra de las ventajas es la integridad de datos y código ya que, al estar centralizada en una o un pequeño número de máquinas servidoras, las actualizaciones están garantizadas para todos los usuarios.

El término servidor de aplicaciones se aplica a todas las plataformas. Dicho término se utiliza para referirse a los servidores de aplicaciones basadas en web, como el control de las plataformas de comercio electrónico integrado, sistemas de gestión de contenido de sitios web y asistentes o constructores de sitios de Internet.

Uno de los ejemplos destacados es el de Sun Microsystems, la plataforma J2EE. Los servidores de aplicaciones Java se basan en la Plataforma Java TM 2 Enterprise Edition (J2EE TM). J2EE utiliza un modelo de este tipo y en general, incluye un nivel Cliente, un nivel Medio, y un EIS. El servidor de tipo Cliente puede contener una o más aplicaciones o navegadores. La Plataforma J2EE es del Nivel Medio y consiste en un servidor web y un servidor EJB. (Estos servidores son también llamados "contenedores".) También podría haber subniveles adicionales en el nivel intermedio. El nivel del SistemaEnterprise Information System (EIS, o "Sistema de Información Empresarial") contiene las aplicaciones existentes, archivos y bases de datos.

3.1.- El servidor de aplicaciones Tomcat

Tomcat es el servidor web (incluye el servidor Apache) y de aplicaciones del proyecto Jakarta, con lo cual, gestiona las solicitudes y respuestas http y, además, es servidor de aplicaciones o contenedor de Servlets y JSP.

Incluye el compilador Jasper, que compila JSP convirtiéndolas en servlets.

Tomcat es un contenedor de servlets con un entorno JSP. Un contenedor de servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario. Podemos dividir los contenedores deservlets en:

1. Contenedores de servlets stand-alone (independientes): Estos son una parte integral del servidor web.

Este es el caso en el que se usa un servidor web basado en Java, por ejemplo, el contenedor de servlets es parte de JavaWebServer (actualmente sustituido por iPlanet). Por defecto Tomcat trabaja en este modo, sin embargo, la mayoría de los servidores no están basados en Java.

2. Contenedores de servlets dentro-de-proceso: El contenedor servlets es una combinación de un plugin para el servidor web y una implementación de contenedor Java. El plugin del servidor web abre una JVM (Máquina Virtual Java) dentro del espacio de direcciones del servidor web y permite que el contenedor Java se ejecute en él. En el caso de que una petición debiera ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando JNI). Un contenedor de este tipo es adecuado para servidores multi-thread de un sólo proceso y proporciona un buen rendimiento pero está limitado en escalabilidad.

3. Contenedores de servlets fuera-de-proceso: El contenedor servlets es una combinación de un plugin para el servidor web y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor web. El plugin del servidor web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP). Si una cierta petición tuviese que ejecutar un servlets, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando IPCs). El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad, etc.).

Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor web existente (actualmente soporta los servidores Apache, IIS). Esto significa que siempre que despleguemos Tomcat tendremos que decidir cómo usarlo y, si seleccionamos las opciones 2 o 3, también necesitaremos instalar un adaptador de servidor web.

Actividad 1.5

Apache y Tomcat historial de versiones.

3.2.- El servidor de aplicaciones con Python

Python puede actuar como servidor de aplicaciones gracias a frameworks web como Flask o Django, que manejan las peticiones HTTP y entregan el contenido. Se puede ejecutar un servidor web local para desarrollo y pruebas utilizando comandos de Python, pero para un despliegue en producción se necesita un hosting de Python que proporcione la infraestructura de servidor. Plataformas como Render y servicios en la nube como Google Cloud gestionan la pila de software y despliegan automáticamente las aplicaciones web de Python.

Un servidor de aplicaciones en Python es un sistema que procesa solicitudes web (usando HTTP o HTTPS) y sirve contenido a los clientes, como los navegadores web.

- **Frameworks:** Se utilizan frameworks de Python como Flask, Django o FastAPI para construir la lógica de la aplicación y manejar los datos.
- **Servidor Local:** Durante el desarrollo, puedes ejecutar un servidor local directamente desde tu entorno Python para probar la aplicación.
- **Despliegue en Producción:** Para que la aplicación esté disponible para los usuarios en internet, se necesita desplegarla en un servicio de hosting especializado para Python.

Actividad 1.6

Python, Flask y Django historial de versiones.