

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, JavaScript, etc.
- ✓ **Protocolos estándar:** éstos permiten que cualquier navegador pueda comunicarse con cualquier servidor web. El más comúnmente usado en WWW es HTTP (Hypertext Transfer Protocol), 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 capas**:

1. **Capa 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 MySQL, PostgreSQL, etc.
2. En una segunda capa estarían los **servidores de aplicaciones web**, ejecutando aplicaciones de tipo Apache, Tomcat, Resin, etc.
3. En una tercera capa estarían **los clientes** del servicio web al que accederían mediante un navegador web como Firefox, Internet Explorer, Opera, etc.

1.1. 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.

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.

1.2. 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.

- ✓ **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:** 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.
 - 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.

- Desarrollado por la empresa Sun Microsystems, pero posteriormente liberado bajo licencia GNU GPL
- ✓ **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.
- ✓ **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 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.

1.3. Tipos de aplicaciones web.

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.
- ✓ **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 y ASP. 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.

1.4. 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.
- ✓ **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 (componentes software que tienen la particularidad de ser reutilizables 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.

1.5. 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.

XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB/MySQL, PHP, Perl. A partir de la versión 5.6.15, XAMPP cambió la base de datos MySQL por MariaDB, un fork de MySQL con licencia GPL.

El programa se distribuye con la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. A esta fecha, XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

1.6. 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.

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.

Cluster.

Con la aparición de los servidores de aplicaciones en cluster se abrió 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.

Cloud computing.

La computación en la nube (del inglés cloud computing), conocida también como servicios en la nube, informática en la nube, nube de cómputo, nube de conceptos o simplemente "la nube", es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es Internet.

En este tipo de computación todo lo que puede ofrecer un sistema informático se ofrece como servicio, de modo que los usuarios puedan acceder a los servicios disponibles "en la nube de Internet" sin conocimientos en la gestión de los recursos que usan.

La computación en la nube son servidores desde Internet encargados de atender las peticiones en cualquier momento. Se puede tener acceso a su información o servicio, mediante una conexión a internet desde cualquier dispositivo móvil o fijo ubicado en cualquier lugar. Sirven a sus usuarios desde varios proveedores de alojamiento repartidos frecuentemente por todo el mundo. Esta medida reduce los costos, garantiza un mejor tiempo de actividad y que los sitios web sean invulnerables a los delincuentes informáticos, a los gobiernos locales y a sus redadas policiales pertenecientes.

Cloud computing es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite incluso al usuario acceder a un catálogo de servicios estandarizados y responder con ellos a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado, o incluso gratuitamente en caso de proveedores que se financian mediante publicidad o de organizaciones sin ánimo de lucro.

El cambio que ofrece la computación desde la nube es que permite aumentar el número de servicios basados en la red. Esto genera beneficios tanto para los proveedores, que pueden ofrecer, de forma más rápida y eficiente, un mayor número de servicios, como para los usuarios que tienen la posibilidad de acceder a ellos, disfrutando de la ‘transparencia’ e inmediatez del sistema y de un modelo de pago por consumo. Así mismo, el consumidor ahorra los costes salariales o los costes en inversión económica (locales, material especializado, etc.).

La computación en la nube consigue aportar estas ventajas, apoyándose sobre una infraestructura tecnológica dinámica que se caracteriza, entre otros factores, por un alto grado de automatización, una rápida movilización de los recursos, una elevada capacidad de adaptación para atender a una demanda variable, así como virtualización avanzada y un precio flexible en función del consumo realizado, evitando además el uso fraudulento del software y la piratería.

El concepto de “nube informática” es muy amplio, y abarca casi todos los posibles tipo de servicio en línea, pero por lo general se refieren a alguna de estas tres modalidades: el software como servicio (por sus siglas en inglés SaaS —Software as a Service—), Plataforma como Servicio (PaaS) e Infraestructura como Servicio (IaaS).

La computación en nube presenta las siguientes características clave:

- ✓ Agilidad: Capacidad de mejora para ofrecer recursos tecnológicos al usuario por parte del proveedor.
- ✓ Costo: Los recursos en la nube suelen tener costos menores a los que un aprovisionamiento físico local podría representar. Generalmente, la inversión inicial que representaría tener un aprovisionamiento local se ve anulada debido a la naturaleza bajo demanda de la nube.

- ✓ Escalabilidad y elasticidad: aprovisionamiento de recursos sobre una base de autoservicio casi en tiempo real, sin que los usuarios necesiten cargas de alta duración.
- ✓ Independencia entre el dispositivo y la ubicación: permite a los usuarios acceder a los sistemas utilizando un navegador web, independientemente de su ubicación o del dispositivo que utilice (por ejemplo, PC, teléfono móvil).
- ✓ La tecnología de virtualización permite compartir servidores y dispositivos de almacenamiento y una mayor utilización. Las aplicaciones pueden ser fácilmente migradas de un servidor físico a otro.
- ✓ Rendimiento: Los sistemas en la nube controlan y optimizan el uso de los recursos de manera automática, dicha característica permite un seguimiento, control y notificación del mismo. Esta capacidad aporta transparencia tanto para el consumidor o el proveedor de servicio.
- ✓ Seguridad: puede mejorar debido a la centralización de los datos. La seguridad es a menudo tan buena o mejor que otros sistemas tradicionales, en parte porque los proveedores son capaces de dedicar recursos a la solución de los problemas de seguridad que muchos clientes no pueden permitirse el lujo de abordar. El usuario de la nube es responsable de la seguridad a nivel de aplicación. El proveedor de la nube es responsable de la seguridad física.⁴
- ✓ Mantenimiento: en el caso de las aplicaciones de computación en la nube, es más sencillo, ya que no necesitan ser instalados en el ordenador de cada usuario y se puede acceder desde diferentes lugares.

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.

2.1. Instalación, configuración y arranque.

Empezamos por identificarnos en la máquina con el usuario root (o utilizar sudo) y ejecutar:

```
# apt-get install apache2  
  
# /etc/init.d/apache2 start  
# /etc/init.d/apache2 stop  
# /etc/init.d/apache2 restart
```

El archivo de configuración predeterminado de Apache2 es /etc/apache2/apache2.conf. Se puede editar este archivo para configurar el servidor Apache2, para configurar el número de puerto, la raíz de documentos, los módulos, los archivos de registros, los hosts virtuales, etc.

Otro método de operar es descargar el código fuente de la aplicación desde la web del proyecto Apache; luego descomprimir, compilar e instalar; realizar el proceso empleando los siguientes comandos:

```
# apt-get update  
# apt-get install build-essential  
# apt-get install libexpat1-dev  
# apt-get install libpcre3-dev  
  
# cd /usr/local/src/  
# wget apache.rediris.es/apr/apr-1.6.5.tar.gz  
# wget apache.rediris.es/apr/apr-util-1.6.1.tar.gz
```

```
# wget apache.rediris.es/apr/apr-iconv-1.2.2.tar.gz
# wget apache.rediris.es/httpd/httpd-2.4.41.tar.gz (o la última versión disponible)

# tar xvfz apr-1.6.5.tar.gz
# cd apr-1.6.5
# ./configure --prefix=/usr/local/apr
# make
# make install
# cd ..

# tar xvfz apr-util-1.6.1.tar.gz
# cd apr-util-1.6.1
# ./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr
# make
# make install
# cd ..

# tar xvfz apr-iconv-1.2.2.tar.gz
# cd apr-iconv-1.2.2
# ./configure --prefix=/usr/local/apr-iconv --with-apr=/usr/local/apr
# make
# make install
# cd ..

# tar xvfz httpd-2.4.35.tar.gz
# cd httpd
# ./configure --prefix=/usr/local/apache --with-apr=/usr/local/apr --with-apr-util=/usr/local/apr-util
--enable-so
# make
# make install

# /usr/local/apache/bin/apachectl configtest
```

Si todo está correcto debería devolver un mensaje del tipo "Syntax Ok"

```
# /usr/local/apache/bin/apachectl start
# /usr/local/apache/bin/apachectl stop
# /usr/local/apache/bin/apachectl restart
```

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.

3.1. El servidor de aplicaciones Tomcat.

Tomcat es el servidor web (incluye el servidor Apache) y de aplicaciones del proyecto Yakarta, 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 de servlets 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. 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. Esto significa que siempre que despluguemos Tomcat tendremos que decidir cómo usarlo y, si seleccionamos las opciones 2 o 3, también necesitaremos instalar un adaptador de servidor web.

3.1.1. Instalación y configuración básica.

En primer lugar es necesario tener instalado , JVM (Java Virtual Machine), ya que el objetivo es que las peticiones a Apache se redirijan a Tomcat empleando un conector proporcionado por Java en este caso.

Empezamos buscando el paquete de Java que nos pueda interesar. Con el siguiente comando obtendríamos la lista del entorno Java:

```
# apt-get update
```

```
# apt-cache search java-runtime
```

Instalamos el siguiente paquete por ser el que más se adapta a nuestras necesidades:

```
# apt-get install default-jre
```

Llegado este punto descargamos Tomcat:

```
# cd /usr/local
```

```
# wget apache.rediris.es/tomcat/tomcat-9/v9.0.27/bin/apache-tomcat-9.0.27.tar.gz
```

Descomprimos el archivo descargado y lo eliminamos:

```
# tar xvzf apache-tomcat-9.0.12.tar.gz
```

```
# rm apache-tomcat-9.0.12.tar.gz
```

3.1.2. Iniciar y parar Tomcat.

```
# cd /usr/local /apache-tomcat-9.0.12/bin
```

```
# ./startup.sh
```

```
# ./shutdown.sh
```

Para comprobar que nuestro servidor está ya escuchando, introducimos en un navegador la URL <http://127.0.0.1:8080>, y éste debería mostrar la página de inicio de Tomcat.

3.1.3. Tomcat en Debian 9.

Empezamos buscando el paquete de Tomcat en los repositorios de Debian. Con el siguiente comando obtendríamos la lista:

```
# apt-get update
```

```
# apt-cache tomcat
```

Instalamos lo siguiente:

```
# apt-get install tomcat8 tomcat8-*
```

Llegado este punto Tomcat se encontrará funcionando (lo podemos comprobar):

```
# service tomcat status
```

Para iniciar y detener el servicio Tomcat:

```
# service tomcat start
```

```
# service tomcat stop
```

4. Estructura y despliegue de una aplicación web.

Una aplicación web está compuesta de una serie de servlets, páginas jsp, ficheros html, ficheros de imágenes, ficheros de sonidos, texto, clases, etc.; de forma que todos estos recursos se pueden empaquetar y ejecutar en varios contenedores distintos.

El servlet es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, estos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets de Java que se ejecutan en servidores en vez de navegadores web. Este tipo de servlets son la contraparte Java de otras tecnologías de contenido dinámico Web, como PHP y ASP.NET.

La palabra servlet deriva de otra anterior, applet, que se refiere a pequeños programas que se ejecutan en el contexto de un navegador web.

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.

De forma genérica podríamos decir que una aplicación web se estructura en tres capas:

1. Navegador web.
2. Tecnología web dinámica (PHP, Java Servlets, ASP, etc.)
3. Base de datos encargada de almacenar de forma permanente y actualizada la información que la aplicación web necesita.

4.1.- Archivos WAR.

Su nombre procede de Web Application Archive (Archivo de Aplicación Web); permiten empaquetar en una sola unidad aplicaciones web de Java completas, es decir que su contenido:

- ✓ Servlets y JSP.
- ✓ Contenido estático: HTML, imágenes, etc.
- ✓ Otros recursos web.

Aportan como ventaja, la simplificación del despliegue de aplicaciones web, debido a que su instalación es sencilla y solamente es necesario un fichero para cada servidor en un cluster, además de incrementar la seguridad ya que no permite el acceso entre aplicaciones web distintas.

Para generar archivos .WAR se pueden emplear diversas herramientas desde entorno IDE "Integrated Development Environment". Por ejemplo, encontramos: NetBeans y Eclipse, ambos Open-Source y también Jbuilder de Borland, Jdeveloper de Oracle; otro modo de construir archivos war es mediante Apache Ant, se trata de una herramienta Open-Source que facilita la construcción de aplicaciones en Java. No es considerado un IDE pero para los que conocen el entorno Linux, es considerado el make de Java.