Unidad 1: Instalación de servidores de aplicaciones web

Índice

Instalación de servidores de aplicaciones web

Internet se ha convertido en la principal herramienta de intercambio de información a la sociedad actual. La conexión de los ordenadores a las redes locales y estas redes a otros de grandes dimensiones ha posibilitado la comunicación global de información entre los ordenadores en todo el mundo. Para poder ofrecer servicios web, los servidores tienen que tener instalados una serie de servicios. El servidor web permite el envío de contenidos a otros ordenadores a través de la red. Los preprocesadores de hipertexto (cómo PHP) permiten crear contenidos web de forma dinámica. Los servidores de bases de datos permiten almacenar información de forma estructurada que se puede usar para generar contenido web. Estos servicios tienen que ser instalados y configurados correctamente antes de empezar a crear el contenido web dinámico de nuestro web.

1. Análisis de requerimientos

Un servidor web es una pieza de software que responde a las peticiones de los navegadores y entrega la página para el navegador a través de Internet. Cuando se llama en una página web por la dirección –la URL (*uniform resource locator*), por ejemplo, www.ioc.org/index.html–, la comunicación entre el navegador y el servidor es posible gracias a tres protocolos:

• **TCP**(*Transmission Control Protocol*, protocolo de control de transmisión): es el responsable de hacer que el mensaje llegue al destino sin errores.

• **IP**(*Internet Protocol*): es el responsable de hacer que el mensaje encuentre el camino hasta el servidor.

• **HTTP**(*Hypertext Transfer Protocol*, protocolo de transferencia de hipertexto): es el protocolo que ha indicado el usuario a la hora de pedir el recurso al servidor. La primera parte de un recurso URL corresponde al protocolo que utilizarán cliente y servidor para intercambiar datos.

1.1 La URL. Acceso a recursos

Una dirección URL tiene la finalidad de acceder a un recurso en un servidor.
Todas las direcciones URL tienen diferentes partes; cogemos como ejemplo http://ioc.xtec.cat:80/educacio/fp-ciclos-formativos/cfgs/index.html:

• **Protocolo**: se indica antes de :// y en este caso es HTTP. Hay otros como son FTP o HTTPS

• **Dirección del servidor**: el lugar de donde se quiere obtener el recurso (ioc.xtec.cat).

• **Puerto**: el puerto del servidor destinado a la aplicación encargada de gestionar las peticiones HTTP (puerto del servidor web). En caso de que no esté especificado se usa el puerto estándar por HTTP (el puerto 80).

• **Ruta hasta el recurso:** corresponde a la ruta en la carpeta dentro del servidor web donde se encuentra el recurso (educacio/fp-ciclos-formativos/cfgs/).

• **Recurso:** es el recurso o fichero que pedimos al servidor (index.html).

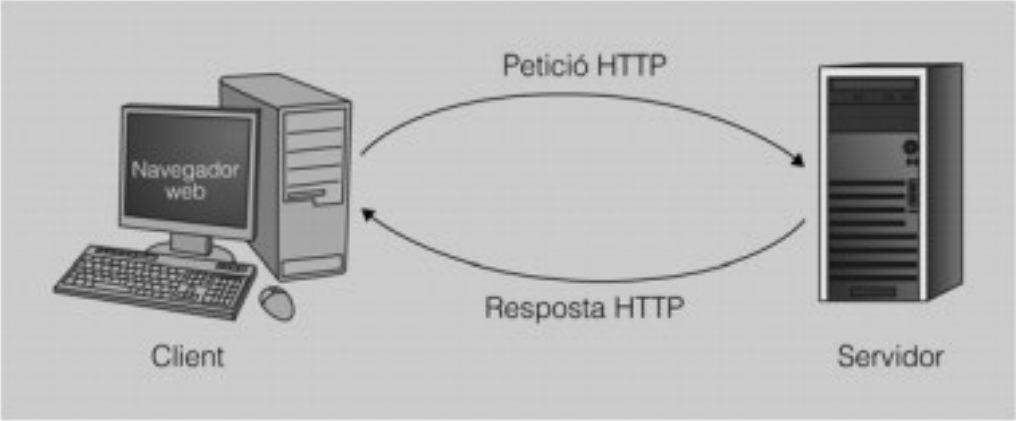
Lo primero que hace el navegador es comunicarse con un servidor de nombres para obtener la dirección IP que corresponde al nombre de la máquina ioc.xtec.cat. Con esta IP se conecta a la máquina servidor. A continuación, el navegador crea una conexión con la dirección IP del servidor al puerto 80, que es el puerto utilizado por defecto en los servidores web. Una vez el cliente y el servidor están conectados, “hablan” usando el protocolo HTTP para que el cliente indique qué

recursos quiere del servidor.

El servidor enviará estos recursos al cliente a través de la conexión establecida.

1.2 HTTP: comunicación entre servidor y cliente

Una vez establecida la conexión entra en juego el protocolo HTTP: el navegador envíauna petición al servidor en que solicita el recurso index.html (figura 1.1). El servidor procesa esta petición y devuelve la página solicitada al navegador, que interpreta las etiquetas HTML y la presenta al usuario.



Sol·licitud client :El client envia, utilitzant el protocol HTTP, una capçalera amb informació sobre el recurs que vol, el mètode que s'ha d'utilitzar. Després de la capçalera, s'envia, si n'hi ha, la informació d'usuari.

La información que el cliente envía al servidor en la petición es la siguiente:

- El método HTTP: la acción que se tiene que hacer.
- El recurso a que se tiene que acceder (una parte de la URL).
- La información que el usuario envía al servidor.

La información que el servidor envía en la respuesta tiene dos partes muy diferenciadas: la cabecera y el contenido.

- La cabecera contiene el código que indica si la petición se ha cumplido. También contiene el tipo de contenido que enviará al cliente.
- El contenido (texto, código HTML, imágenes, etc.) del recurso pedido.

Métodos GET y POST

La primera parte de una petición HTTP es el tipo de método, que sirve para indicarle el tipo de petición que se ha hecho. Los métodos pueden ser HEAD, TRACE, DELETE, OPTIONS, CONNECT, GET y POST . Estos dos últimos son los que realmente utilizaréis.

Tanto con GET cómo con POST se puede enviar información del cliente al servidor, pero hay algunas diferencias.

La información enviada con el método GET se envía después de la dirección URL, por ejemplo:

ioc.xtec.cat/index.php?opc=edu&lang=cat.

La información de usuario son los datos que hay a partir del símbolo ?. Este hecho implica que la cantidad de datos que se pueden enviar con GET esté limitada. Además, la información que se ha enviado de este modo es pública puesto que aparece a la propia URL, que es fácilmente accesible. Por lo tanto no se puede usar para enviar datos privados, como una clave de acceso, por ejemplo.

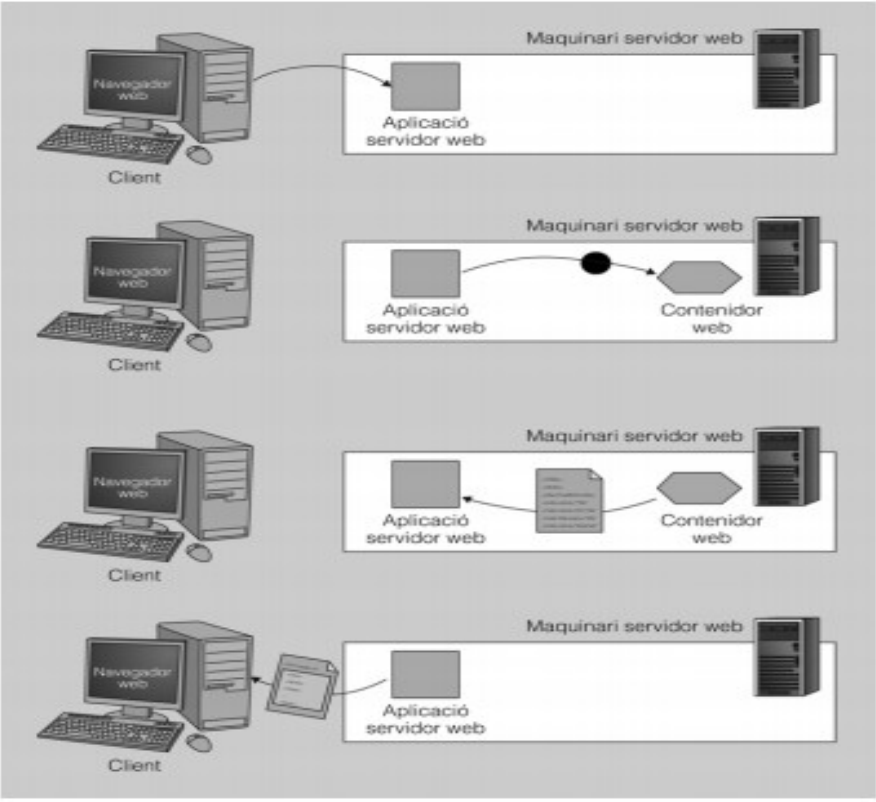
Con el método POST, en cambio, la información se envía como parte del contenido y no es visible para el usuario.

1.3 Contenedores web

El problema de los primeros servidores web es que solamente entregaban recursos estáticos. Recibían una petición, miraban si el recurso estaba disponible y lo enviaban.

La ventaja de las páginas web dinámicas es que el contenido se genera dinámicamente y, por lo tanto, no existe estática y previamente, en el servidor.

Los servidores web tan sólo saben enviar páginas estáticas. Para que un servidor web pueda trabajar con contenido dinámico, es necesario instalar otra **aplicación que ayude el servidor web a servir contenido dinámico** en los clientes (cómo se observa a la figura 1.2). Las aplicaciones encargadas de hacer esta tarea se denominan **contenedores web** y son las que permiten trabajar con PHP, Java, ASPE, C, Python, etc. Su funcionamiento es transparente para el cliente.



1.2 Servidors i contenidors: Els servidors només saben processar continguts estàtics; per ser capaços de treballar amb continguts dinàmics, com una base de dades, necessiten l'ajuda de programes externs: els contenidors

En la figura 1.2 podéis observar el funcionamiento de un servidor web junto con el contenedor. Cuando el servidor encuentra un recurso que no es estático, traspasa la petición al contenedor web. Este contenedor procesa la petición, genera dinámicamente una respuesta en forma de contenido web y la pasa al servidor.
Finalmente el servidor envía el contenido al cliente.

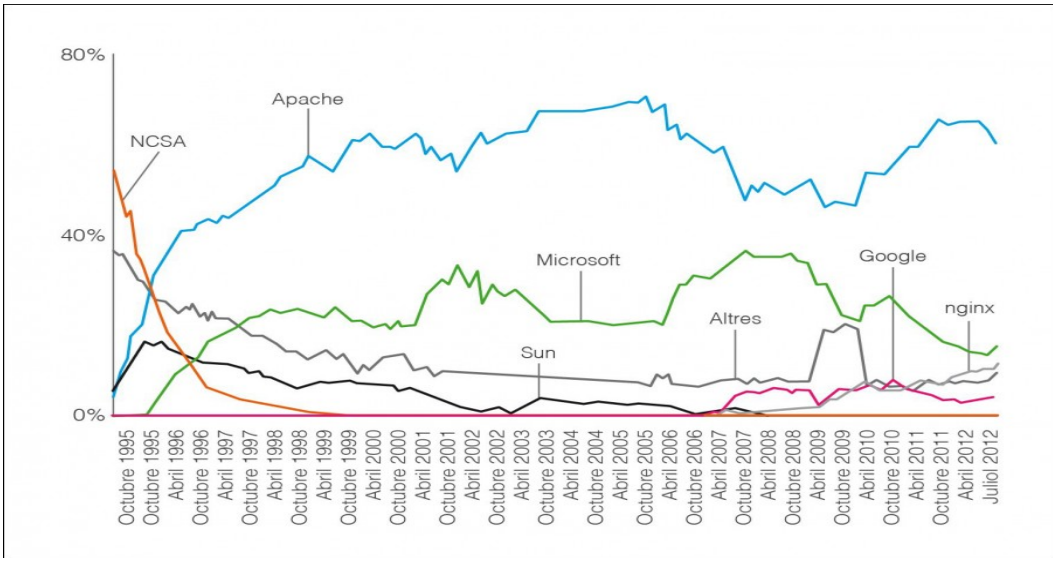
2. Servidores web

Los servidores web pueden estar implementados dentro del *kernel* del sistema operativo o dentro del espacio de memoria de usuario (como cualquier otra aplicación del sistema).

Los **servidores web dentro del *kernel***, como por ejemplo el IIS de Windows, generalmente se ejecutan más rápido puesto que son parte integrando del sistema y pueden usar todos los recursos de hardware directamente cuando los necesitan (como por ejemplo la memoria sin paginar o la memoria caché).

Los **servidores web que se ejecutan en modo de usuario** tienen que pedir al sistema cuando tienen necesidades de memoria o recursos de CPU. Estas peticiones tardan en realizarse y no siempre pueden ser satisfechas, puesto que el sistema reserva los recursos para su uso y lo tiene que repartir entre todo el resto de aplicaciones en ejecución.

Cómo se puede ver en la figura 1.3 y la tabla 1.1, a pesar de que hoy en día hay muchos servidores web diferentes, Apache es el servidor web más popular en Internet con un 65% de cuota de mercado. El resto del mercado está repartido entre los servidores de otras compañías, los más populares de los cuales son IIS de Microsoft y nginx.



1.3

Taula 1.1. Servidors web més utilitzats per les pàgines web més visitades d'Internet a Juliol de 2012

Desenvolupador	Juny 2012	Percentatge	Juliol 2012	Percentatge	Canvi
Apache	448.452.703	64,33%	409.185.675	61,45%	-2.89
Microsoft	95.891.537	13,76%	97.385.377	14,62%	0,87
nginx	72.881.755	10,46%	73.833.173	11.09%	0,63
Google	22.464.345	3,22%	22.931.169	3.44%	0,22

2.1 Servidor web Apache

El servidor HTTP Apache es un servidor web de código abierto para plataformas Une (BSD, GNU/Linux, etc.), Windows, Macintosh y otros sistemas operativos que implementa el protocolo HTTP y que utiliza el concepto de lugar virtual.

El Apache empezó como una serie de módulos desarrollados para un servidor web en el cual trabajaban en el Centro Nacional de Supercomputación de Aplicaciones de los Estados Unidos (NCSA). Un golpe acabó este proyecto, programadores de todo el mundo encontraron que había la necesidad de tener un repositorio central en que se pudiera mantener el código y los módulos que se habían desarrollado para poder continuar trabajando en el desarrollo del servidor. Así surgió la Fundación de Software Apache (Apache Software Foundation).

El Apache fue diseñado desde el principio de forma modular, de forma que el software podía ser ampliado por otros desarrolladores que escribían pequeñas partes del código de una manera fácil. La clave de este éxito es la creación de una API (interfaz de programación de aplicaciones) modular y una serie de fases muy definidas que atraviesa cada petición que llega al servidor. Estas fases van desde la inicialización del servidor (el Apache cuando lee los ficheros de configuración), hasta la traducción de una dirección URL en un nombre de fichero del ordenador.

2.2 Instalación de Apache

Per instal·lar paquets en distribucions basades en Debian GNU/Linux, s'han d'executar amb drets de root o amb un usuari pertanyent al grup sudo

La forma general de instalar paquetes en los sistemas basados en Debian es:

```
# sudo apt-get install {nombre_del_paquete}
```

Para instalar el servidor web tenéis que instalar el paquete **apache2**. Por defecto el servidor usa el directorio **/var/www** para contener las páginas que servirá.

Una vez instalado podéis comprobar su funcionamiento modificando el fichero **/var/www/index.html** y comprobando que se muestra correctamente si abrís desde el navegador web la dirección http://localhost tal como se puede ver a la figura 1.4.

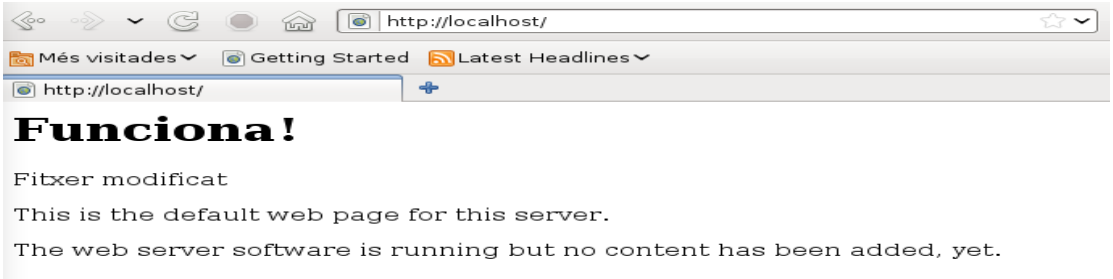


Figura 1.4. Comprobación del funcionamiento del servidor Apache

2.3 Configuración básica de Apache

Se puede controlar el arranque y parada del servidor Apache 2 en función de la versión del sistema operativo instalado bien con un script que se encuentra dentro de la carpeta **/etc/init.d**. La sintaxis de este script es la siguiente:

 # /etc/init.d/apache2 ordre

en el cual ***ordre*** puede ser alguna de las siguientes opciones:

• start: para iniciar el servidor
• stop: para parar el servidor
• restart: para reiniciar el servidor. Para y pone en marcha el servidor.

O bien para versiones como Ubuntu 16.04 con la instrucción :

 # systemctl ordre apache2

Recordáis que tenéis que trabajar con permisos de superusuario para modificar la ejecución del servidor, así que usáis ***sudo*** o trabajáis como usuario *root*. Podéis comprobar como el servidor se para y pone en marcha entrando a la página de localhost .

Una vez finalizada la instalación de Apache 2.0, ya podemos sacar algunas conclusiones. En primer lugar, podemos afirmar que utiliza un diseño modular (basado en módulos) que se utiliza en muchas de las funciones básicas del servidor web. Por otro lado, tal como muestra el proceso de instalación, se ha instalado por defecto el paquete **apache2-mpm-worker**. Este paquete proporciona una versión mucho más ágil y rápida del servidor. De hecho, los módulos de multiprocessament (MPM, multe-processing modulas) son los responsables de conectar con los puertos de red de la máquina, aceptar las peticiones y gestionar las respuestas correspondientes. En el caso de los servidores web, el puerto de funcionamiento estándar es el 80. Justo es decir que dentro del proceso de instalación por defecto del servidor web Apache también hay la instalación del módulo de directorio de usuario (*userdir*), puesto que cada usuario dispondrá de un espacio en su espacio de disco (*home*) para crear y guardar sus páginas web. Con todo, si bien hay un amplio abanico de paquetes que se tienen que instalar con Apache para aumentar la eficiencia, hace falta no dejar de banda la descarga de la documentación del software, puesto que acontecerá de gran utilidad para el administrador. El paquete a instalar es concretamente: apache2-doc.

Considerando que, en el supuesto de que hayamos instalado la documentación, tendremos que volver a arrancar el servidor web, también podemos acceder a la documentación mediante la página principal predefinida en el servidor web Apache. Por lo tanto, desde cualquier navegador web nos podremos documentar sobre el funcionamiento del servidor web escribiendo en la barra de navegación http://localhost/manual/.

2.4 Configuración de directorios personales

A partir de ahora, cualquier fichero que ponemos en **/var/www** se tiene que visualizar cuando escribimos en el navegador **http://IP/nombre_de el_fichero**, en qué IP es la IP de la máquina en la cual hay el servidor web instalado.

Podéis **activar el directorio personal de los usuarios** de la siguiente manera:

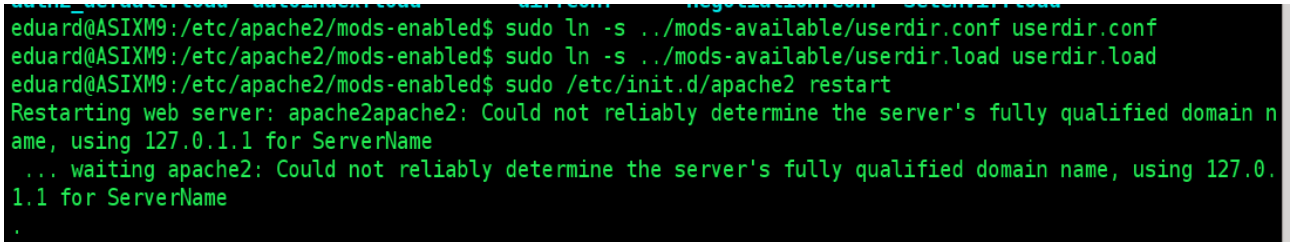
1. Creáis una carpeta dentro del directorio personal del usuario denominada **public_html.** Lo tenéis que hacer con vuestro usuario, **no uséis el usuario *root* o sudo.**

2. Creáis dentro de la carpeta **/mods-enable** de Apache unos enlaces simbólicos hacia los módulos que queréis usar, tal como se puede ver a la figura 1.5.

3. Reiniciáis el servidor.

En este momentos el servidor os dará el mensaje que podéis ver en la figura 6 cuando lo iniciáis. Esto es porque el servidor no tiene un nombre de dominio totalmente calificado. Es normal hasta que no lo configuráis.

Figura 1.5. Activación de los directorios de los usuarios



A partir de este momento el usuario puede meter el contenido web personal dentro de la carpeta public_html que ha creado en su directorio personal. Esta será accesible a través de la dirección http://localhost/nombre_usuario. Por defecto mostrará el fichero index.html o un listado de los ficheros del directorio si no encuentra este fichero.

2.5 Configuración de módulos

Apache es un servidor modular. Esto quiere decir que en el cuerpo del servidor está incluido únicamente las funcionalidades más básicas. Las funcionalidades extras están disponibles mediante los módulos que se pueden cargar al Apache. Si el servidor está compilado para usar los módulos dinámicamente, este se pueden añadir en cualquier momento. Si no es así, Apache se tiene que recompilar para añadir o sacar los módulos. Podéis instalar módulos de Apache con la orden apt-get.

Por ejemplo, **para instalar el módulo** para autentificación MySQL podéis usar la siguiente orden:

#sudo apt-get install libapache2-mod-auth-mysql

Una vez instalado, el módulo estará disponible a ***/etc/apache2/mods-available .***

Los módulos que se están usando están en ***/etc/apache2/mods-enabled****.*

Para activar un módulo podéis hacer un enlace simbólico al módulo de ***mods-available*** .

#sudo ln -s /etc/apache2/mods-availabe/auth_mysql.load /etc/apache2/mods-enabled/auth_mysql.load

o podéis usar la orden:

 # sudo a2enmod auth_mysql

Para **desactivar el módulo** podéis usar:

#sudo a2dismod auth_mysql

Podéis mirar los módulos que está usando Apache mirando directamente el contenido del

directorio de módulos activos (**/etc/apache2/mods-enabled**) o con la orden:
#sudo apache2ctl -M | sort

2.6 Control de accesos

Para autenticar a los usuarios a Apache se puede hacer de dos maneras **Básica y Digest.** A la **Básica** el usuario introduce en el navegador web su **nombre de usuario y contraseña** y se envían al servidor **sin cifrar**, también es la más sencilla de configurar. A la **Digest** el **usuario y contraseña** y se envían al servidor **cifrados.**

Estos métodos **controlan** el acceso a los **recursos**, pero n**o cifran la comunicación cliente-servidor** un golpe se ha validado el acceso.

Autenticación básica

El módulo de Apache que controla este método de autenticación es **mod_auth_basic.**

- Está soportado por todos los navegadores web.
- El usuario y la contraseña no van cifradas del navegador web al servidor.

Por cada directorio que se quiera proteger al archivo **/etc/apache2/sitesavailable/default**, o en el fichero relativo al *host* </ Directory>:

```
1 <Directory "/var/www/privado">
2 AuthType Basic
3 AuthName "Directorio privado"
4 AuthUserFile /etc/apache2/passwd/.htpasswd
5 Require valid user
6 </ Directory>
```

En el cual:

- **AuthName**: indica el nombre del dominio de autenticación y es texto que se le presentará al usuario en el momento de autenticarse.
- **AuthType**: indica el método de autenticación a usar.
- **AuthUserFile**: indica la ruta al archivo de texto que contendrá los nombres de usuario y contraseñas usadas en la autenticación HTTP básica.
- **Require**: indica restricciones sobre los usuarios que tienen acceso a los recursos especificados. Puede ser *valid-user* que identifica cualquier usuario incluido al archivo de contraseñas *.htpasswd* o user *<lista de usuarios>* que especifica la lista de usuarios de .htpasswd que pueden acceder.

3. Lenguajes script

Un lenguaje de script (guiones) es un lenguaje de programación basado en guiones que son seguidos línea por línea por un procesador.

Un script es un conjunto de instrucciones que afectan en la página web, si estas instrucciones afectan al navegador se encuentran junto al cliente y si afectan al servidor se encuentran al ordenador servidor. Si pensáis en páginas web que habéis visitado recientemente que cambian cada vez que las visitáis o incluso que pueden cambiar mientras las estáis visitando es posible

que utilicen algún script.

Por ejemplo, cuando buscáis una palabra al buscador Google, este utiliza scripts para sugerir palabras relacionadas, para poner anuncios y para encontrar el que estáis buscando. Cuando compráis en una tienda virtual, esta utiliza scripts para mostrar los productos y para guardar vuestro carro de compra.

A continuación veréis las características principales de los dos tipos de lenguajes de guiones.

3.1 Lenguajes script de cliente

Los lenguajes script de cliente se ejecutan al sistema en el cual se encuentra el navegador web (Firefox, Chrome, Opera, Safari, etc.) del usuario y pueden formar parte del código HTML del documento o encontrarse en archivos adjuntos. Los lenguajes script de cliente dan interactividad a las páginas HTML, el código es ligero y al llegar al navegador es interpretado (es decir, los scripts se ejecutan sin una compilación preliminar) por el navegador.

El proceso de un navegador al sistema del cliente es:

1. El usuario pide una página web a un servidor desde su navegador.
2. El servidor busca la página y, si la encuentra, la devuelve al navegador del usuario.
3. El navegador muestra la página y al mismo tiempo ejecuta los scripts de cliente para crear la salida HTML.

Los lenguajes de script de cliente modifican la página web una vez llega al navegador. La ventaja principal de los scripts de cliente es que al no ejecutarse al servidor no consumen CPU del servidor ni ancho de banda. Se utilizan para tareas que ahorran trabajo al servidor, como por ejemplo validar los campos de un formulario, o para mejorar la apariencia y la interacción con el usuario.

El lenguaje de referencia para programar código en el lado del cliente es **Javascript**. No se necesita ninguna licencia para utilizarlo. Al ejecutarse en la máquina del cliente, si ésta es lenta los scripts se pueden ejecutar lentamente, o incluso no se llegarán a ejecutar si el navegador no entiende el lenguaje de script. A las preferencias de los navegadores modernos se puede activar o desactivar la ejecución de Javascript. Cómo el código está en el ordenador del cliente, el usuario lo puede ver para copiarlo o manipularlo.

Con la tecnología **AJAX** y el **Javascript,** la mejora de funcionalidad de las páginas web al navegador hace que se asemejen cada vez más a aplicaciones normales del ordenador.

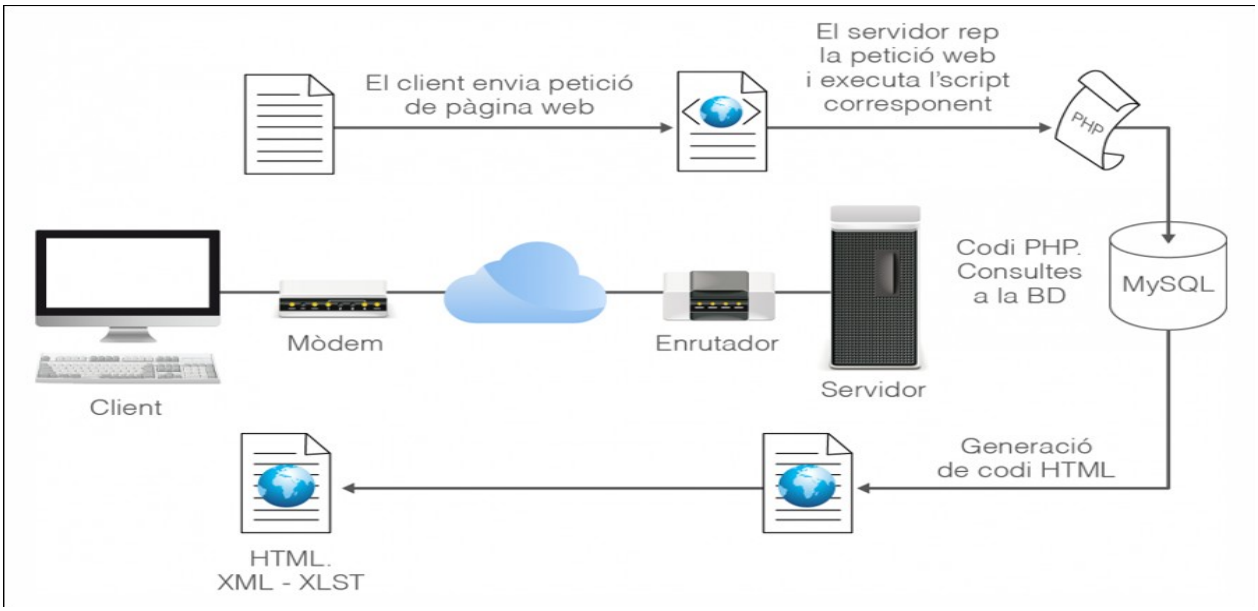
3.2 Lenguajes script de servidor

Las aplicaciones web desarrolladas con lenguajes script de servidor se ejecutan en el servidor web y por lo tanto el usuario no tiene acceso al código fuente del servidor. El servidor contiene estos scripts que son interpretados cada vez que se recibe una petición de una página web. El proceso es el siguiente:

1. El usuario a través de su navegador pide una página al servidor.
2. El script de servidor es interpretado montando el contenido a devolver (la página web).
3. Cuando el script acaba, el contenido montado se devuelve al cliente.

Al esquema de la figura 1.6 se ilustra una máquina cliente que envía una petición de página web a un servidor que ejecuta un script de servidor que requiere consultar una base de datos. Como

respuesta, el cliente recibe una página web HMTL.



Entre las características más significativas de los lenguajes script de servidor se encuentran las de permitir a los usuarios tener cuentas individuales y la de comunicarse con bases de datos (MySQL, PostgreSQL, SQL Server, Oracle, etc). Permiten un nivel de personalización, privacidad y recuperación de información muy importante.

Las páginas web de comercio electrónico y las páginas web sociales hacen un uso intensivo de los scripts de servidor.

Como ejemplos de los principales lenguajes script de servidor tenemos el PHP (páginas con extensión php) y la ASPE/Asp.net (páginas con extensión asp/aspx).

Otros lenguajes son Perl (páginas con extensión pl), Java Server Pages (páginas con extensión jsp), ColdFusion (páginas con extensión cfm) y Python (páginas con extensión py).

AJAX Son las siglas de *Asynchronous Javascript And Xml*, (Javascript asíncrono y XML), un conjunto de tecnologías que permiten actualizar partes de una página web sin tener que volverla a cargar entera

3.3 Instalación de PHP

3.3.1 Instalación de PHP 7

Hay una serie de paquetes necesarios para trabajar con PHP y Apache. Los podéis instalar con apt-get (php7.0, php-common, php7.0-common, php7.0-fpm, php7.0-json, php7.0-opcache, php7.0-readline, libapache2-mod-php7.0, php7.0-cli).

Una vez instalados tenéis que reiniciar el servidor Apache porque los cambios sean efectivos De manera opcional podéis instalar del PHP5 los paquetes referenciados aquí para apoyar en diferentes utilidades. Si sólo necesitáis una instalación básica del PHP, no hace falta instalar los paquetes siguientes, aunque son recomendables.

• php7.0-gd: biblioteca para el uso de imágenes.
• libphp-adodb: biblioteca adodb para PHP.
• Smarty: sistema Templates PHP.
• php-pear: bibliotecas Pear para PHP.

• php7.0-json: apoyo para objetos JSON.
• php7-xsl: apoyo para XSLT.
• php7-odbc: para conexiones odbc.

3.3.2 Instalación de PHP 5

Hay una serie de paquetes necesarios para trabajar con PHP y Apache. Los podéis instalar con apt-get (php5, php5-common, libapache2-mod-php5 php5-cli).

Una vez instalados tenéis que reiniciar el servidor Apache porque los cambios sean efectivos De manera opcional podéis instalar del PHP5 los paquetes referenciados aquí para apoyar en diferentes utilidades. Si sólo necesitáis una instalación básica del PHP, no hace falta instalar los paquetes siguientes, aunque son recomendables.

• php5-gd: biblioteca para el uso de imágenes.
• libphp-adodb: biblioteca adodb para PHP.
• Smarty: sistema Templates PHP.
• php-pear: bibliotecas Pear para PHP.
• php5-json: apoyo para objetos JSON.
• php5-xsl: apoyo para XSLT.
• php5-odbc: para conexiones odbc.

4. MySQL

El MySQL es un sistema de gestión de base de datos relacional. Se calcula que tiene más de 6 millones de instalaciones. El programa se ejecuta como un servidor multiusuario y proporciona acceso a una serie de bases de datos.

El proyecto de código fuente está disponible en los términos de la licencia pública general GNU, a pesar de que el MySQL es propiedad de una empresa con ánimo de lucro, que también lo patrocina. El MySQL era propiedad de la compañía sueca MySQL AB, una empresa que Sun Microsystems compró el 2008. El abril de 2009, Oracle *compró Sun e indirectamente MySQL AB, uno de los principales competidores en* algunos de sus productos.

El MySQL es utilizado por una gran cantidad de proyectos que requieren una base de datos con funciones completas, sistema de gestión, administración y es usado, entre otros, por WordPress, phpBB, Google y Facebook.

Muchas veces se hace referencia a la combinación de Linux, el servidor web Apache, el contenedor web de PHP y MySQL con el acrónimo *LAMP*.

4.1 Instalación de MySQL en sistemas basados en Debian (Ubuntu)

La forma más sencilla de instalar MySQL a Debian es mediante apt-get. Tenéis que instalar los paquetes mysql-cliente y mysql-server. Os pedirá qué queréis que sea la contraseña de root de MySQL. MySQL tiene sus propios usuarios, que son independientes de los usuarios del sistema Linux. Por lo tanto **es importante cambiar la contraseña de root y apuntarla.**

Si más adelante queréis volver a **cambiar la contraseña** lo podéis hacer con la orden:

 # sudo mysqladmin -u root -h localhost password ’clave’

Podéis cambiar *localhost* por el nombre de la máquina. Una vez hecho ésto, ya podéis entrar al servidor MySQL con la orden:

 # mysql -u root -p

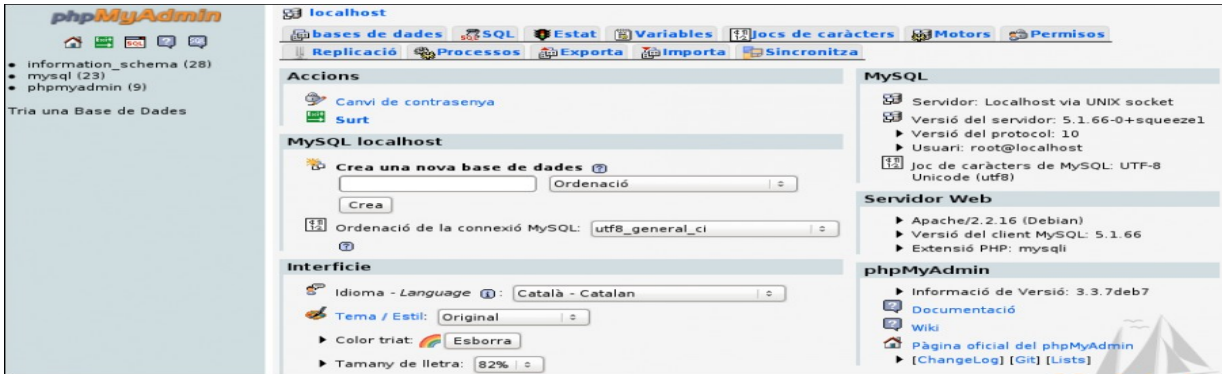
Para parar o poner en marcha el servidor de MySQL tenéis el script que lo controla en **/etc/init.d/mysql** que acepta las típicas opciones (*start*, *stop*, *restart*).

Aseguraos que tenéis instalado el paquete **php7-mysql** o **php5-mysql** puesto que lo necesitaréis para trabajar con MySQL y PHP. Lo podéis **comprobar con la orden:**

#dpkg -s php7-mysql
o
#dpkg -s php5-mysql

5. phpMyAdmin

El phpMyAdmin es un programa de distribución libre en PHP, creado por una comunidad sin ánimo de lucro. Es una **herramienta** muy completa que permite a**cceder a todas las funciones** típicas de la **base de datos MySQL** por medio de una interfaz web muy intuitiva, tal como se puede ver a la figura 1.7.



La aplicación tan sólo es un conjunto de archivos escritos en PHP que se copian en un directorio del servidor web, de forma que, cuando se accede a estos archivos, encontramos una herramienta que nos permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editar y borrar, borrar tablas, etc. Incluso ejecutar sentencias SQL y hacer una copia de seguridad de la base de datos.

5.1 Instalación del phpMyAdmin

La página de inicio del proyecto es www.phpmyadmin.net. Desde aquí podéis bajar los ficheros de la última versión de la aplicación, que después tenéis que colocar en nuestro servidor web.

Para distribuciones GNU/Linux basadas en Debian, también se puede hacer la instalación directamente desde el repositorio:

 # sudo apt-get install phpmyadmin

Durante la instalación os hará una serie de preguntas:

- En cuanto al servidor web que usará tenéis que contestar: apache2.
- Cuando os pregunta si queréis configurar la base de datos phpMyAdmin con *dbconfig-common*, tenéis que contestar que sí.
- Cuando os pregunta la contraseña de root de MySQL (porque phpMyAdmin quiere crear sus tablas a la base de datos MySQL)
- Os pedirá la clave de acceso a la base de datos MySQL del usuario que se ha creado por el phpMyAdmin. Podéis escribir una clave vosotros o, si la dejáis en blanco, creará una clave aleatoria.

Antes de poder acceder al panel de control de phpMyAdmin se tiene que modificar la configuración de Apache.

Buscáis donde está el fichero **apache.conf**. Generalmente lo tendríais que tener a **/etc/phpmyadmin/apache.conf**

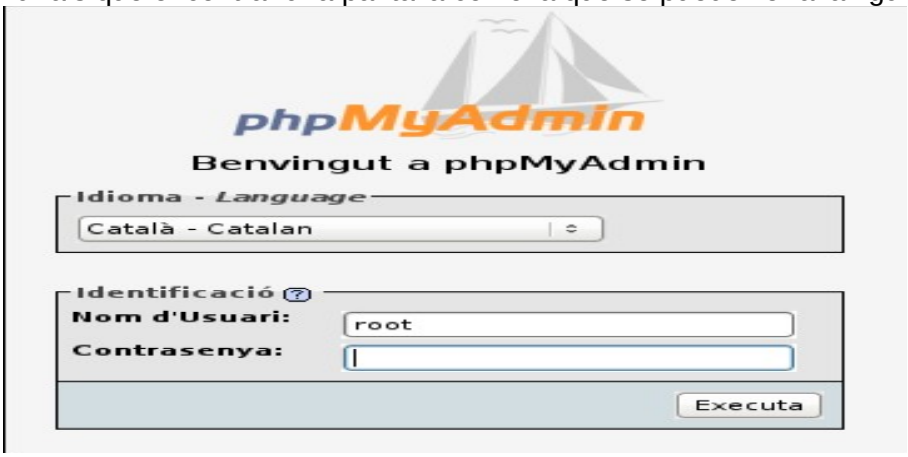
En cualquier caso lo podéis encontrar con las órdenes:
#updatedb
#locate apache.conf

Editáis la **configuración de Apache e incluid el fichero apache.conf de phpMyAdmin.**

Por eso abrís (con derechos de root) el fichero **/etc/apache2/apache2.conf** y añadís al final del fichero la siguiente línea:

Include /etc/phpmyadmin/apache.conf

Ahora ya podéis reinicar el servidor Apache e ir a la dirección http://localhost/phpmyadmin para ver si funciona. Tendríais que encontrar una pantalla como la que se puede ver a la figura 1.8.



6. Utilidades de prueba e instalación integrada

La instalación y posterior configuración de un servidor web Apache, de un servidor de bases de datos MySQL y del lenguaje de programación PHP es una tarea compleja que sólo pueden emprender usuarios con buenos conocimientos informáticos.

Los llamados paquetes LAMP,WAMP o MAMP simplifican la tarea de instalar y configurar automáticamente Apache, PHP y MySQL en Linux, Windows o Mac OS

Estos paquetes proporcionan:

- Servidor Web Apache
- Base de datos MySQL
- Lenguaje de programación PHP
- Accesos por el arranque y la parada de los servicios
- Facilitados para la configuración de los servicios

El paquete XAMPP es un paquete de software libre que tiene versiones por Linux, Windows, Solaris y Mac OSX, además de ofrecer los servicios básicos de un paquete integrado (es decir, Servidor web Apache + Servidor de bases de datos MySQL + Lenguaje PHP). Así pues, ofrece:

- ProFTPD como servidor de archivos por FTP
- Lenguaje Perl
- Servidor de datos SQLite
- SSL por páginas seguras HTTPS (OpenSSL)
- Estadísticas de acceso (Webalizer)

Podéis encontrar la última versión de XAMPP a www.apachefriends.org/en/xampp.html

6.1 Utilidades de prueba.

Para **comprobar si funciona la instalación de Apache2 y PHP:**

- Creáis un documento a la raíz en /var/www, denominado info.php y rellenadlo con la información siguiente:

```
<?php phpinfo ();?>
```

- A continuación, guarda el archivo.
- Probáis el servidor con el navegador y la dirección del tipo siguiente:

  http://localhost/info.php

Si todo ha ido bien, os mostrará una pantalla con información del PHP (ver figura 1.9):

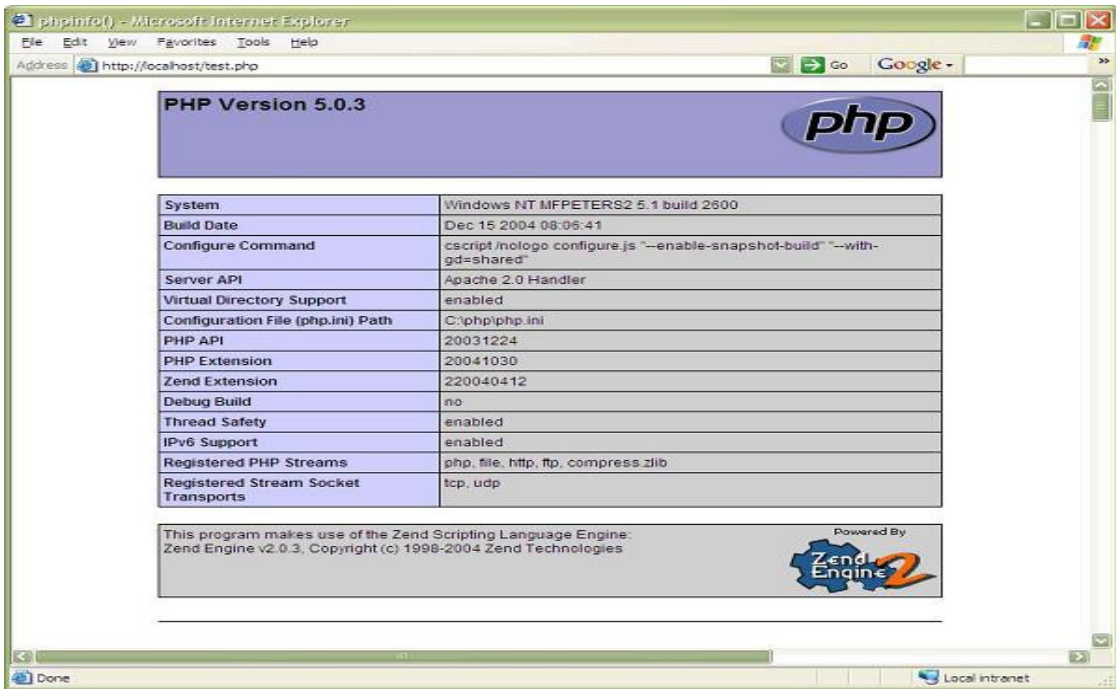


Figura 1.9. Información de configuración de PHP

Para comprobar, una vez acabada la instalación de MySQL, que podéis acceder al entorno, entraréis como usuario *root* y ej**ecutaréis la orden *show Databases*.** Veréis las dos bases de datos que instala MySQL y ejecutaréis *quit* para salir del entorno.

A un terminal escribiréis la petición siguiente:

elatorre@debian:~$ mysql -h localhost -u root -p
Enter password:
Welcome tono the MySQL monitor. Commands end with ; oro ((/))g.
Your MySQL connection id is 41
Server version: 5.1.49□3 (Debian)

Copyright (c) 2000, 2010, Oráculo and/oro its affiliates. Ajo rights reserved.
This software comas with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome tono modify and redistribute it under the GPL v2 license

*Type 'help;' oro '((/))h' **for** help. Type '((/))c' tono clear the current input statement.*

mysql> show databases;
+□□□□□□□□□□□□□□□□□□□□□□□□□+
Database |
+□□□□□□□□□□□□□□□□□□□□□□□□□+
| information_schema |
| mysql | +□□□□□□□□□□□□□□□□□□□□□□□□□+
rows in siete (0.00 seco)
mysql> quit
Bye
elatorre@debian:~$

7. Documentación

Durante el proceso de instalación y configuración de un servidor de aplicaciones web, hay que tomar nota de todos los pasos que se han seguido: **primero en el momento de la instalación y después en el momento de la configuración**, de forma que tengáis una referencia completa de todo el que habéis hecho. Conocer los detalles de por qué se ha puesto un valor a una variable o a un fichero de configuración puede ser muy útil en el futuro, puesto que es probable que pasado un tiempo os olvidáis.

Crear la documentación es una parte importante del proceso y puede ser muy útil cuando aparecen dificultades, incidencias o en mantenimientos, y se quieren saber rápidamente los valores de los parámetros configurados sin tener que ir a mirar directamente los archivos de configuración al servidor. La documentación también es importante para la transmisión de conocimiento.

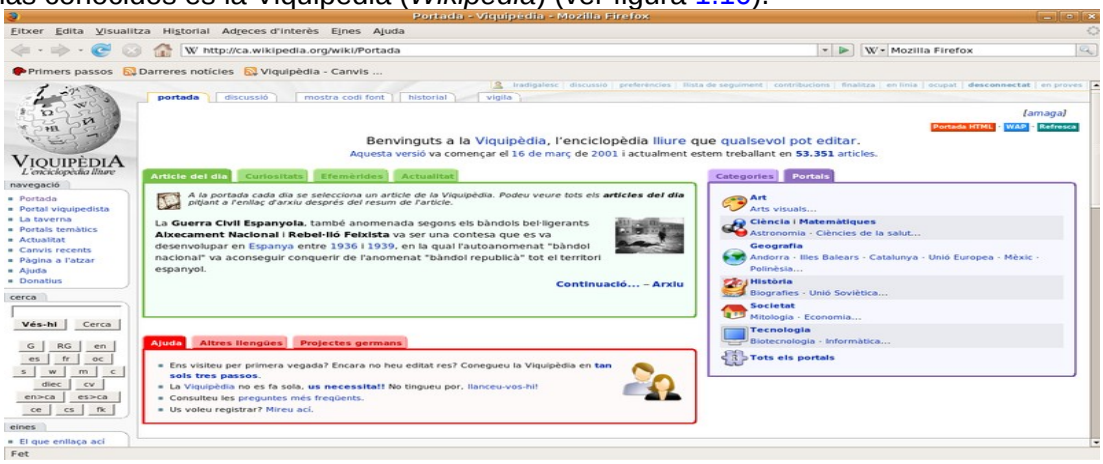
Podéis pasar la documentación a otras personas que pueden aprender como está montado el servidor sin tener que explicar nada por vuestra parte.

Cuando se quiere replicar en otra máquina la instalación y configuración de un servidor web, la documentación es una gran ayuda para saber como se ha hecho y cómo está montado el servidor que ya está en funcionamiento.

Una buena manera de empezar a hacer la documentación es consultar el índice que habéis seguido para hacer la instalación de la aplicación. Se pueden analizar los puntos del índice que se han ido tirando durante el proceso de instalación, hacer un resumen y documentar especialmente las partes en las cuales se difiere de la guía que se ha usado para hacer la instalación.

7.1 Herramientas de ayuda a la creación

Una herramienta útil que podéis usar para crear la documentación es un wiki. Un wiki (del hawaià *wikiwiki*, que quiere decir *rápido*) es un sitio web colaborativo que puede ser editado desde el navegador por los usuarios. Los usuarios de un wiki pueden, de este modo, crear, modificar, enlazar y borrar el contenido de una página web, de forma interactiva, fácil y rápida. Uno de los wikis más conocidos es la Viquipèdia (*Wikipedia*) (ver figura 1.10).



1.10

Los proyectes wiki tienen normalmente historiales con todas las modificaciones que se han hecho de sus contenidos. De este modo se pueden ver todas las versiones y recuperar la información eliminada o deshacer las ediciones incorrectas, puesto que los cambios se aplican normalmente al instante, sin que ningún tipo de usuario de confianza o administrador los haya revisado y confirmado antes.

Hasta ahora, la aplicación de los sistemas wiki deben de su fama sobre todo a la creación de

enciclopedias colectivas como la Viquipèdia. Pero las características de los wikis los convierten en una herramienta efectiva para la escritura colaborativa, y cada vez son más usadas en empresas como webs e intranets económicas y eficaces para la gestión del conocimiento.

Hay muchos softwares de wiki diferentes (podéis consultar el enlace http://en.wikipedia.org/wiki/comparison_of_wiki_software), y que incluso podéis instalar en el mismo servidor web. Entre los más recomendables encontramos los softwares DokuWiki y MediaWiki, que usan una sintaxis muy similar.

El **DokuWiki** (ver logo figura 1.11) está licenciado bajo GPL 2 y escrito en el lenguaje de programación PHP. Funciona con archivos de texto plano y por lo tanto no necesita ninguna base de datos.

El **MediaWiki** (ver logo figura 1.12) es el software de wiki con el cual se ha hecho la Viquipèdia, está licenciado bajo licencia GPL, está escrito en el lenguaje de programación PHP y utiliza MySQL como gestor de base de datos.

Figura 1.12. Logo MediaWiki