

Unidad 1

Servidores de Aplicaciones Web

Implantación de Aplicaciones Web. 2º de ASIR



Esta obra está bajo una licencia de Creative Commons.
Autor: Jorge Sánchez Asenjo (año 2011) <http://www.jorgesanchez.net>
e-mail: info@jorgesanchez.net

Esta obra está bajo una licencia de Reconocimiento-NoComercial-CompartirIgual de Creative Commons
Para ver una copia de esta licencia, visite:
<http://creativecommons.org/licenses/by-nc-sa/2.5/es/legalcode.es>
o envíe una carta a:
Creative Commons, 559 Nathan Abbot



Reconocimiento-NoComercial-CompartirIgual 2.5 España

Usted es libre de:



copiar, distribuir y comunicar públicamente la obra



hacer obras derivadas

Bajo las condiciones siguientes:



Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).



No comercial. No puede utilizar esta obra para fines comerciales.



Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Apart from the remix rights granted under this license, nothing in this license impairs or restricts the author's moral rights.

Advertencia

Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.
Esto es un resumen legible por humanos del texto legal (la licencia completa) disponible en los idiomas siguientes:

Catalán Castellano Euskera Gallego

Para ver una copia completa de la licencia, acudir a la dirección
<http://creativecommons.org/licenses/by-nc-sa/2.5/es/legalcode.es>

índice

(1.1) servicios en la web	5
(1.1.1) perspectiva histórica	5
(1.1.2) historia de la Web	5
(1.1.3) servicios web 1.0	9
(1.1.4) servicios web 2.0	9
(1.1.5) aplicaciones en la nube. <i>cloud computing</i>	10
(1.2) servidores de aplicaciones web	11
(1.2.1) servidores web	11
(1.2.2) servidores de aplicaciones web	11
(1.2.3) tecnologías para crear aplicaciones web.....	12
(1.2.4) tecnologías para crear aplicaciones web	12
(1.2.5) servidores de aplicaciones web comerciales	13
(1.3) instalación de Apache	14
(1.3.1) distribuciones	14
(1.3.2) instalación binaria en Windows	15
(1.3.3) instalación de código fuente en Linux.....	18
(1.3.4) instalación del paquete desde el sistema Linux	19
(1.4) configuración de Apache	20
(1.4.1) opciones de httpd.....	20
(1.4.2) instalar Apache como servicio al arranque.....	21
(1.4.3) archivos de configuración.....	21
(1.5) Instalación de MySQL	23
(1.5.1) pasos previos	23
(1.5.2) documentación	23
(1.5.3) instalación en Windows.....	23
(1.5.4) instalación en Linux/Unix.....	28
(1.5.5) asegurando la instalación	33
(1.6) PHP	34
(1.6.1) ¿qué es PHP?	34
(1.6.2) instalación en Windows.....	34
(1.6.3) instalar en Linux	35
(1.7) paquetes completos Apache/MySQL/PHP	38
(1.7.1) AppServ	38
(1.7.2) XAMPP	39
(1.7.3) Wampserver	41

(1)

servidores de aplicaciones web

(1.1) servicios en la web

(1.1.1) perspectiva histórica

El éxito de Internet está absolutamente ligado a la web. Tanto que hoy en día para la inmensa mayoría de las personas es indistinguible qué es la web y qué es Internet. Todo se hace en la web. Dicho de otro modo (más coherente con esta asignatura), la web ofrece todo tipo de servicios.; de hecho ofrece todos los servicios que ofrece Internet.

Esto no ha sido así siempre. Inicialmente Internet no tenía web. Tenía servicios como el correo electrónico, la transmisión de ficheros, los grupos de noticias,... Todos ellos se manejaban de forma tan incómoda que sólo los profesionales de la informática podían utilizar Internet.

Todo cambio cuando **Tim Bernes Lee** ideó las páginas web. El éxito de Internet se disparó gracias a la facilidad de manejo que ofreció la web. De tal manera que en poco tiempo los usuarios y usuarias de las páginas web, demandaban que estas ofrecieran más servicios: no sólo mostrar estáticamente un texto fijo, sino que contuvieran vídeo, animaciones,... y poco a poco: acceso a servidores de bases de datos, manejo del correo electrónico, transmisión de ficheros, compra y venta de productos,...

Hoy en día desde la Web se puede hacer cualquier tarea, editar documentos, leer el correo electrónico, enviar mensajes, retocar fotos, ver películas, escuchar radio,...

(1.1.2) historia de la Web

Señalamos algunos hechos importantes en la historia de la web

- 1989. **Tim Bernes Lee** científico del **CERN**¹ crea la **World Wide Web** uniendo el lenguaje **HTML** y el protocolo **HTTP**.

¹ **Conseil Européen pour la Recherche Nucléaire**, es decir, **Consejo Europeo para la Investigación Nuclear**), es un laboratorio de investigación en física de partículas situado en Suiza.

■ 1992.

- Se crea el navegador **Mosaic** que facilita el manejo de la web al permitir leer páginas web desde un entorno gráfico con el ratón. A partir de él se crearán todos los demás navegadores.
- Hay 50 servidores web a finales de año

■ 1993.

- El CERN anuncia que la web será libre para todo el mundo, renuncia a sus posibles patentes.

■ 1994.

- Aparece **Netscape Navigator**, que en poco tiempo se convertiría en el navegador más utilizado en el mundo
- A finales de año hay 10000 servidores web
- Se crea la web **Yahoo!** que se convertirá en el sitio web más popular y el primer índice de la Web.
- Aparece **CGI**, **Common Gateway Interface**, Interfaz de Pasarela Común que permite realizar programación en otros lenguajes y asociar los programas a páginas web a fin de darlas más potencia.

■ 1995.

- Aparece **Windows 95**, la informática de consumo llega a casi todos los hogares del mundo desarrollado.
- **Explosión de la web**, Internet se empieza a hacer muy popular y la web se empieza a convertir en el servicio más habitual para acceder a Internet. Poco a poco aparecen miles y miles de sitios web en Internet
- **Microsoft** crea **Internet Explorer**, su navegador para acceder a Internet. Comienza la primera guerra de navegadores entre Explorer y Navigator.
- Se crea el lenguaje Java que tiene una relación casi directa con Internet
- Aparece el lenguaje **Javascript** que permite insertar código dentro de las páginas web.
- Aparece el sitio web **Geocities**, antecesor de las redes sociales, en el que la gente publica páginas web y las agrupa por gustos en ciudades virtuales.
- Se crea el lenguaje **PHP** que permite integrar código de servidor en las páginas web a fin de crear webs más potentes.

■ 1996.

- Se crea el lenguaje **CSS** que permite dar formato a las páginas web de forma más avanzada.
- Se lanza **Hotmail** el sitio web que permite utilizar correo electrónico desde la web. Se convierte en el correo electrónico más popular y acaba siendo comprado por **Microsoft**.

■ 1997.

- Escándalo Lewinsky. Los informes oficiales y detalles sobre el escándalo aparecen antes en la web que en los medios digitales, el público se acostumbra a leer noticias en la web lo que provoca una crisis en los medios tradicionales.

■ 1997.

- El organismo de estandarización **W3C** lanza la versión 4 de **HTML**. Hasta este momento la última versión oficial de HTML. A su vez **Explorer** y **Navigator** lanzan la versión 4, poco a poco Explorer le va comiendo el mercado a Navigator.
- La empresa Macromedia crea **Flash**, que permite incrustar objetos gráficos avanzados en las páginas web, por lo que las posibilidades visuales de las mismas se disparan a cambio de que los navegadores integren este famoso plugin.

■ 1998.

- Microsoft crea el lenguaje script de servidor **ASP**, que se convierte en un éxito inmediato a pesar de ser de código propietario.
- El **W3C** lanza la versión 2 de CSS, la última lanzada hasta la fecha.

■ 1999.

- **Sun** crea el lenguaje de scripts de servidor basado en Java, **JSP**; para competir con ASP y PHP que se han popularizado.
- **Microsoft** crea para **Explorer 5** un objeto XML llamado **XMLHttpRequest** que será copiado en los demás navegadores en los años sucesivos. El acceso a este objeto desde JavaScript permite crear páginas muy dinámicas y potentes (como **Google Maps** por ejemplo). A esta forma de programar se le llamará (en 2005) **AJAX**, acrónimo *asynchronous JavaScript and XML*

■ 2000.

- Aparece la **Wikipedia**, enciclopedia creada con la colaboración desinteresada de miles de personas cuyos artículos crecen de manera exponencial cada año.
- **Google** convierte a su buscador en el más rápido y poco a poco en el más popular desplazando a **Yahoo!** y **Altavista** como los más populares.
- La **W3C** define **XHTML** versión de HTML que sigue de manera estricta las reglas de XML. Se considera actualmente la única versión válida de HTML.

■ 2001.

- Aparece **Explorer 6** integrado en el popularísimo sistema **Windows XP**. Con ello Microsoft gana la primera guerra de navegadores.
- Los sucesos del **11S** provocan un colapso en Internet.
- Estallido punto com, cientos de empresas que habían tenido ganancias enormes pero especulativas comienzan a desplomarse y desaparecer. Se habla de redefinir la web. Al año siguiente ya se habla de la web 2.0

- Se populariza el uso de blogs (o weblogs) para publicar en Internet al ser más fáciles de crear por todo tipo de usuarios
- Se crea **Drupal**, herramienta de gestión de contenidos (CMS) que permite crear sitios colaborativos. Su éxito verdadero comienza a partir del 2007 compitiendo con Joomla

■ 2002.

- Aparece **MySpace** popular web social de éxito espectacular en esos años.

■ 2003.

- Aparece la red social gráfica y virtual **Second Life**.
- Se crean las redes sociales profesionales **Linkedin** y **Xing** (llamada inicialmente **Open Social Club**)

■ 2004.

- Aparece **Facebook** verdadera red social digital cuyo éxito desborda en los siguientes años hasta llegar a cientos de millones de usuarios.
- Aparece **Gmail** servicio de correo web de Google que poco a poco se populariza por el tamaño de su buzón claramente superior a la competencia y por sus grandes prestaciones y experiencia de usuario.
- Se comienza a trabajar en los borradores de **HTML5** y **CSS3**

■ 2005.

- Tras liberar Netscape el código de su navegador, se crea el navegador **Mozilla Firefox** basado en código abierto. Se inicia la segunda guerra de navegadores que, por ahora, actualmente gana (por poco) Firefox.
- Aparece **Google maps** con mapas de Estados Unidos y Canadá que poco a poco incluirán información de todo el planeta y le convierten en el buscador de rutas más popular.
- Se crea **Joomla**, que se acaba convirtiendo en el CMS más popular.

■ 2006.

- Aparece el sitio de **microblogging twitter**. En poco tiempo se convierte en uno de los más populares por su velocidad y número espectacular de usuarios.

■ 2007.

- Google lanza **Street View** como complemento a Google Maps desde el que se ven fotografías de todas las calles de varias ciudades estadounidenses. Poco a poco consigue fotografiar la mayoría de las carreteras y calles de diversos países, entre ellos España-

■ 2008.

- Google lanza su navegador **Chrome**. Actualmente está a punto de pasar a Firefox como el navegador más popular de Internet.

■ 2010.

- Explosión móvil. El crecimiento desorbitado del uso de **smartphones** hace que cada vez más gente acceda a la web desde dispositivos móviles. En China casi un 70% de los usuarios utiliza un móvil para acceder a Internet.

■ 2011.

- **HTML5** y **CSS3** empiezan a imponerse a pesar de seguir en estado de borrador. Parece que el futuro pasa por ambas tecnologías.

(1.1.3) servicios web 1.0

Hoy en día se llaman así a los servicios que ofrecían los sitios web clásicos. En realidad la inmensa mayoría de sitios actuales siguen siendo web 1.0 puesto que solo utilizan tecnologías clásicas.

La realidad es que web 1.0 se define como lo que no es web 2.0, una definición vaga pero que en el punto siguiente se pretende aclarar mejor.

(1.1.4) servicios web 2.0

Se trata de la forma de denominar a las páginas web que ofrecen servicios orientados al usuario. Las páginas web tradicionales ofrecían la misma información para todos los usuarios, ahora varían para cada usuario, le permiten un manejo más rico e incluso le hacen partícipe del contenido.

Inicialmente la web era simplemente textos e imágenes junto con los hipervínculos que permiten saltar hacia otros contenidos. Ahora el tipo de contenido es mucho más variado: animaciones, vídeo, sonido, juegos, aplicaciones ofimáticas,... Este tipo de páginas empezaron a proliferar tras el estallido de la burbuja punto com y es en el año 2004 cuando la empresa **O'Reilly Media** le utiliza en una conferencia e incluso llega a registrarlo y lo hace popular.

Hoy en día es difícil de definir, pero sin duda hace referencia a un tipo de servicios web muy concretos. Podemos decir que hay tres pilares que conforman las páginas web 2.0:

- **Aplicaciones Ricas de Internet.** Es quizá el término más tecnológico relacionado con la web 2.0. Se llaman Aplicaciones Ricas de Internet (también conocidas con las siglas **RIA**) a aquellas páginas web que ofrecen servicios que las asemejan con las aplicaciones de escritorio.

Permiten en definitiva utilizar la web como si estuviéramos utilizando una aplicación con toda su potencia en un ordenador local.

Inicialmente estas aplicaciones web requerían que el usuario instalara plugins (como **Java** o **Flash** por ejemplo) para poder utilizarlas; pero hoy en día con **AJAX**, basta con el propio navegador.

- **Arquitectura orientada al servicio (SOA).** Otra de las claves, se trata de una tecnología que permite diseñar aplicaciones basándose en peticiones a un determinado servicio. De esta forma se puede crear pequeños elementos software muy reutilizables y además independientes del lenguaje con el que fueron creados.

- **Web social.** Es la parte más evidente y entendible de la web 2.0. El término se refiere a que el usuario posee una interacción mucho mayor en la web siendo partícipe de lo que en ella ocurre.

En definitiva las páginas web 2.0 tienen estas características:

- El contenido no solo es creado por los propietarios de las mismas, sino que es creado y modificado por los propios usuarios.
- Las páginas permiten controles más potentes de manejo por los usuarios
- Los diferentes servicios ofrecidos por distintas páginas tienen facilidad para interaccionar (podemos colocar un mapa de Google fácilmente o acceder a Facebook para compartir un detalle de la página,...)
- El contenido no está centralizado sino que residen en una maraña (nube) de múltiples servidores colocados incluso en diferentes puntos geográficos.

Es habitual comparar servicios ofrecidos por la web 1.0 con respecto a los de la web 2.0, por ejemplo:

○ Web 1.0	○ Web 2.0
○ Hotmail (clásico)	○ Gmail
○ Doubleclick	○ AdWords
○ Geocities	○ Wordpress
○ Yahoo!	○ Google
○ Enciclopedia Británica	○ Wikipedia
○ Messenger	○ Twitter, Facebook
○ Akamai	○ Torrents
○ ...	○ Google docs
○ ...	○ Aviary

(1.1.5) aplicaciones en la nube. *cloud computing*

Es otro de los términos fundamentales para definir los servicios ofrecidos por las páginas actuales. Se basa en ofrecer al usuario servicios de modo que el usuario pueda acceder a ellos desde cualquier dispositivo conectado a Internet, ofreciendo una altísima disponibilidad del mismo.

Se basa en la programación distribuida de aplicaciones, pero a tal nivel que las aplicaciones se distribuyen incluso por cientos de servidores de partes distintas del planeta y así poder responder a una demanda ingente de peticiones de servicio y además poseer una altísima capacidad de tolerar fallos.

De esta manera es la tecnología que permite utilizar el software como un servicio (**SaaS**, **Software as a Service**), al igual que la web 2.0 (con la que está íntimamente relacionada), al usuario le basta un navegador o una pequeña aplicación (como una **App** de un dispositivo móvil) para acceder y utilizar el servicio.

La computación en la nube apareció para responder tecnológicamente a los retos de empresas como Google o Facebook que tenían que responder a gran velocidad a las peticiones de millones de usuario. Hoy en día el concepto permite que se utilice Internet como la base de trabajo, sustituyendo así al propio ordenador personal que puede ser un equipo de menor potencia al delegar en la **nube** el proceso de las tareas y el almacenamiento de la información.

Ejemplos de servicios en la nube serían:

- **Discos duros virtuales.** Como **DropBox** o **SugarSync** que sirven para almacenar todo tipo de datos y que estos estén accesibles desde cualquier dispositivo conectado.
- **Aplicaciones de ofimática web.** Con capacidad para crear e incluso compartir documentos de texto, hojas de cálculo,... Entre ellas están **Google Docs** y **Microsoft Web Apps**.
- **Copias de seguridad en línea.** Al estilo del disco virtual, pero pensado para que las empresas tengan un respaldo en caso de pérdida de información.
- **Calendarios.** Permiten disponibilidad permanente de los datos de agenda personal.
- **Sistemas operativos web.** Permiten utilizar un ordenador virtual disponible a través de Internet. El más famoso es **EyeOS** y en España **TribalOS**.
- **Servicios VPN.** Permiten a través de un navegador u otra aplicación cliente ligera, controlar uno o más ordenadores remoto de modo que no sea necesario acceder físicamente a él.
- **Redes sociales.** Como Facebook, tuenti o LinkedIn.
- **Bibliotecas multimedia.** Como **Youtube** o **Flickr**.
- **Marcadores en línea.** Como **del.icio.us** o **Digg**.

Existen detractores de este tipo de servicios, puesto que al final información personal o empresarial crítica queda en poder de empresas privadas. E incluso hay quien afirma (especialmente los movimientos a favor del software libre) que supone un retroceso que asemeja el funcionamiento de los sistemas de los años 70 en los que los trabajadores utilizaban un terminal muy ligero que se conectaba a un ordenador central de cuya potencia y software dependían.

(1.2) servidores de aplicaciones web

(1.2.1) servidores web

Los servidores web son los encargados de recibir las peticiones referidas a páginas o elementos de la web a través del protocolo **http**. Normalmente es un software alojado en un ordenador servidor.

Normalmente es el navegador el que pide al servidor web el recurso que desea el usuario, para finalmente recibir dicho recurso (si fue válida la petición) y traducirle si es necesario a su forma legible por el usuario (es decir la traducción de HTML la hace el navegador).

(1.2.2) servidores de aplicaciones web

Podemos entender que es una ampliación de los anteriores. Es decir son servidores web, pero que tienen capacidad de almacenar y gestionar aplicaciones web. Entendiendo que una **aplicación web** es un servicio al que los usuarios acceden a través de la web.

Este tipo de servidores no sólo sirven para atender peticiones http, sino que además son capaces de entender instrucciones de lenguajes avanzados de la web y traducirlas o bien son capaces de acceder a recursos de otros servidores. Ese proceso se hace de

forma transparente al usuario, es decir el usuario pide el servicio a través, normalmente, de su navegador y el servidor de aplicaciones atiende la petición, e interpreta el código de la aplicación a fin de traducirle y mostrar al usuario el resultado de forma entendible por su navegador (es decir en formato HTML).

A la forma de trabajar de un servidor de aplicaciones, se le conoce normalmente como arquitectura de tres capas (a veces se habla de más capas). Una primera capa es la del navegador que es capaz de traducir código del lado del cliente (HTML, JavaScript, CSS, Flash,...). Para ello esa capa debe de disponer de todos los componentes necesarios para hacer esa labor en el ordenador del usuario.

La segunda capa la forma el servidor de aplicaciones en su labor de traducir código en el lado del servidor (JSP, PHP, Ruby on Rails, Cold Fussion...) y convertirlo al formato entendible por el navegador.

La tercera capa son todos los servicios a los que accede el servidor de aplicaciones para poder realizar la tarea encomendada a la aplicación (por ejemplo el acceso a la base de datos):

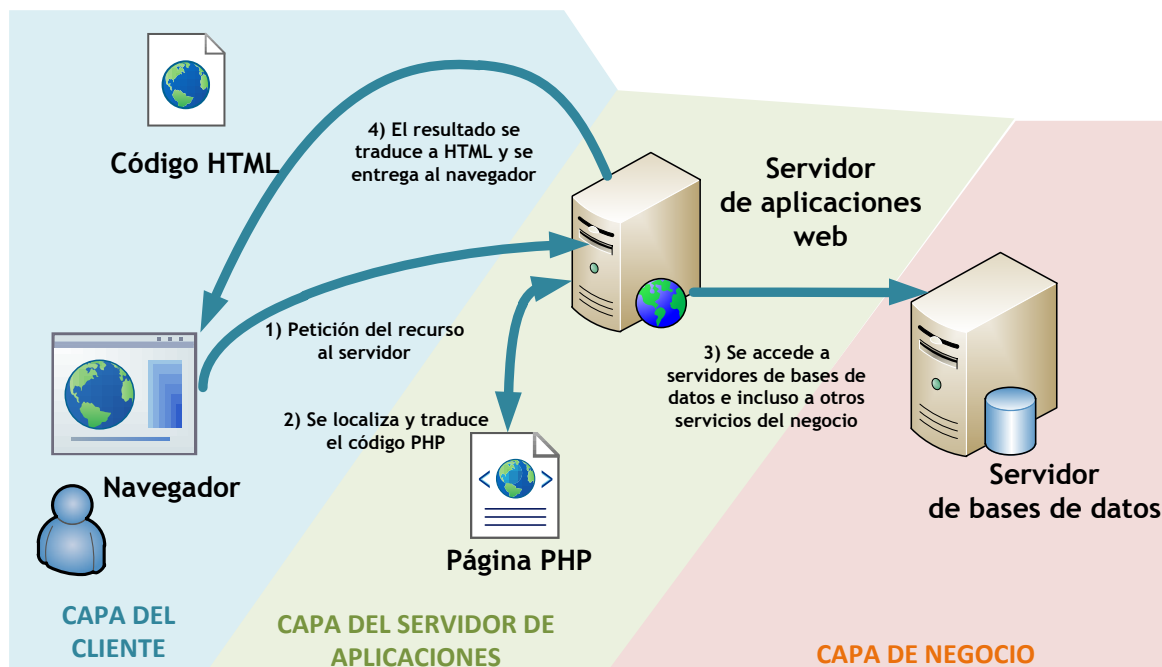


Ilustración 1, Funcionamiento de un servidor de aplicaciones PHP, el servidor de aplicaciones puede acceder a servicios ofrecidos por otros servidores, de este modo la lógica de negocio

(1.2.3) tecnologías para crear aplicaciones web

- PHP. Permite crear páginas HTML a las que

(1.2.4) tecnologías para crear aplicaciones web

lenguajes de script de servidor

- PHP (*Personal Home Pages*). Se trata de un lenguaje de scripts de servidor; es decir código que se incrusta en las páginas HTML y que requiere ser traducido por un servidor de aplicaciones que devolverá un resultado en formato HTML.

- **ASP (Active Server Pages)**. Tecnología de Microsoft similar a la anterior, sólo está pensada para utilizar en servidores de Windows, especialmente en **IIS**.
- **JSP (Java Server Pages)**. Competidor de ASP que usa como base el lenguaje Java.
- **Cold Fussion**. Otro lenguaje de scripts, esta vez propiedad de Adobe. Es el más sencillo de todos, pero es de uso más caro porque requiere servidores especiales (Servidores de **Cold Fussion**).

plataformas de desarrollo de servicios web empresariales

- **J2EE (Java 2 Enterprise Edition)**. Nombre que se da a la plataforma de creación de aplicaciones web empresariales de Java. Está formada fundamentalmente por el propio lenguaje Java, **EJB (Enterprise Java Beans)**, componentes reutilizables empresariales), **servlets** y **JSP** además de otros componentes.
- **.NET**. Plataforma de Microsoft que permite (entre otras muchas posibilidades) crear aplicaciones y servicios web, haciendo especial énfasis en el transporte de datos mediante XML.

frameworks MVC

En inglés **framework** se puede traducir como estructura; en el sentido que nos ocupa un framework sería un marco de trabajo. MVC son las siglas de **Modelo-Vista-Controlador** un paradigma de programación de aplicaciones que separa en tres niveles el trabajo: por un lado el modelo que hace referencia a las aplicaciones empresariales conocido como la lógica de negocio (por ejemplo el Sistema Gestor de la Base de Datos); la vista hace referencia al aspecto visual de la aplicación con el usuario; el controlador finalmente es la parte que controla las acciones del usuario y las comunica a los dos apartados anteriores.

Es, en definitiva, un modelo de trabajo que facilita la creación de aplicaciones web complejas. Se les conoce también con el nombre de **plantillas**.

- **Ruby on Rails**. Se trata de un marco de trabajo muy exitoso por la facilidad que tiene de programar y sus buenos resultados visuales. Se puede ejecutar en casi cualquier servidor web, basta con instalar el componente correspondiente.
- **Apache Struts**. El marco de trabajo más famoso para la creación de aplicaciones J2EE. Muy preparado para utilizar con Apache.
- **Spring**. Otro marco para trabajar en Java J2EE que tiene bastante éxito. Tiene incluso una versión para las aplicaciones .NET
- **Django**. Escrita en **Python** y pensada para utilizar en ese lenguaje que facilita la creación de aplicaciones web.
- **Zend**. Framework escrito para PHP. Uno de los más populares para este lenguaje.
- **Yii**. Otro framework PHP de reciente creación, pero de gran crecimiento comercial.

(1.2.5) servidores de aplicaciones web comerciales

Apache

Indudablemente Apache es el servidor web más popular de la actualidad. Abarca el 70% de todos los servidores web instalados. Se trata de un software de código abierto que

utiliza una licencia de tipo **Apache License** que es una variante de la licencia **GPL** de Linux. Eso significa que se puede distribuir sin problemas e incluso mejorar.

Dispone de multitud de módulos que convierten a Apache en un servidor capaz de gestionar todo tipo de aplicaciones, lo que también le convierte en el servidor de aplicaciones más popular de la actualidad; por ejemplo dispone de módulos para:

- Implementar **SSL**. Protocolo de seguridad en la transferencia de información
- Enlace con el servidor **Tomcat** de aplicaciones, para implementar aplicaciones **Java** de servidor.
- Módulo para **Perl**
- Módulo para **PHP**
- Módulo para **Python**.
- etc.

IIS

Abreviatura de **Internet Applications Server**, es el servidor de aplicaciones de Microsoft que está presente en las versiones profesionales de Windows y en todas las de servidor. Viene con el propio sistema operativo y para instalarle basta con agregar componentes al mismo y elegir las funciones deseadas del servidor IIS.

El servidor incluye un servidor web (tanto **http** como **https**), servidor **ftp**, **webDAV** y **smtp**. Además se comporta como servidor de aplicaciones web **.NET** y admite extensiones para diversos tipos de aplicaciones (incluido PHP).

Es el segundo servidor web más popular tras Apache (20% del mercado).

nginx

Servidor web cada vez más popular. Muy ligero, pero con algún problema. Se puede ampliar con módulos que permiten servir aplicaciones web.

Apache Tomcat

Basado en Apache, y desarrollado por la misma fundación; es un servidor web con capacidad para desarrollar aplicaciones **Java J2EE**, el más popular para este tipo de aplicaciones.

(1.3) instalación de Apache

(1.3.1) distribuciones

Apache es un software de código abierto bajo una licencia de tipo Apache License que es una variante de la licencia GPL lo que significa que incluso se puede modificar el código libremente y generar una nueva variante propia de Apache que incluso puede cerrar el código, pero que en todo momento debe de mostrar un aviso de que se trata de una variación hecha sobre el código original que era perteneciente a una licencia Apache.

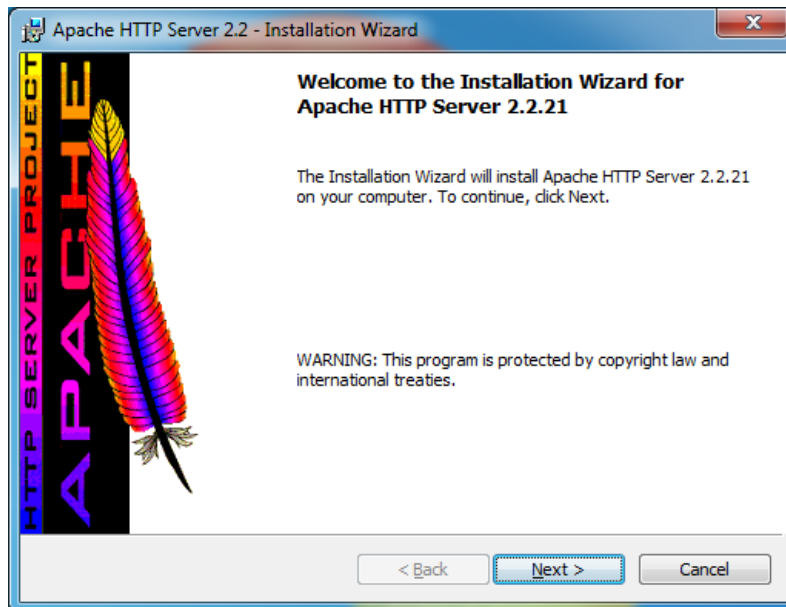
Por ello podemos instalar Apache compilando su código fuente (y así mantener el espíritu del software libre), o bien descargar código ya compilado ejecutable e instalar Apache de esa forma más sencilla.

Las diferentes posibles instalaciones se pueden descargar de <http://http.apache.org>

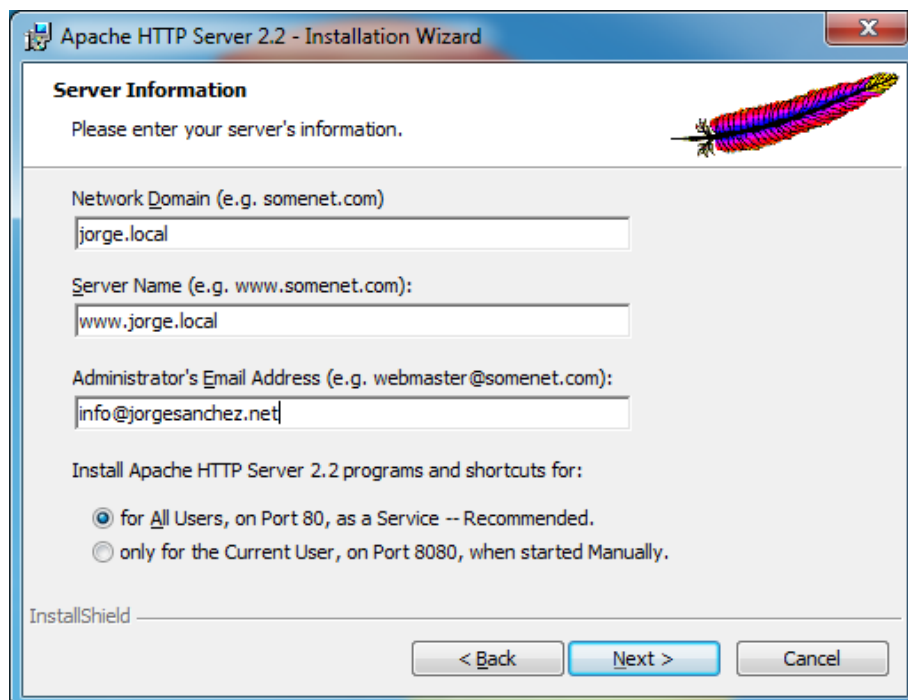
(1.3.2) instalación binaria en Windows

Hay dos posibilidades de binarios para Windows: con o sin servidor SSL. Si optamos por la opción clásica (sin instalar el software **Open SSL**). Los pasos son:

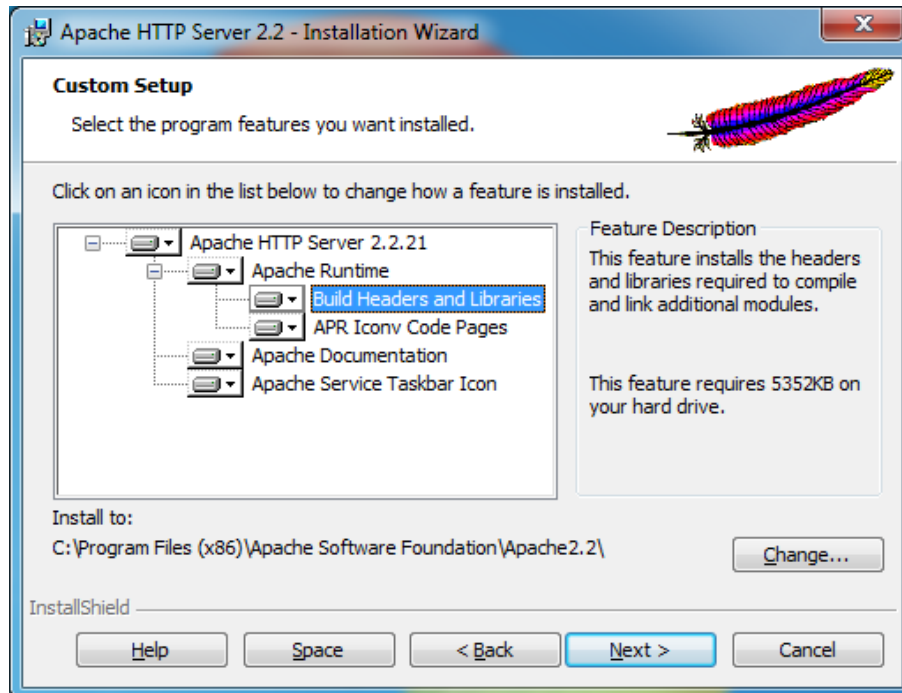
- (1) Descargar y ejecutar el archivo de instalación:



- (2) Aceptar la licencia
- (3) Indicar el nombre del dominio en el que se integra al servidor, nombre del servidor, correo del administrador y si deseamos instalar el servidor para todos los usuarios



- (4) Elegir el tipo de instalación (es aconsejable elegir la personalizada, *Custom*) y seleccionar los componentes de Apache a instalar (recomendable todos)

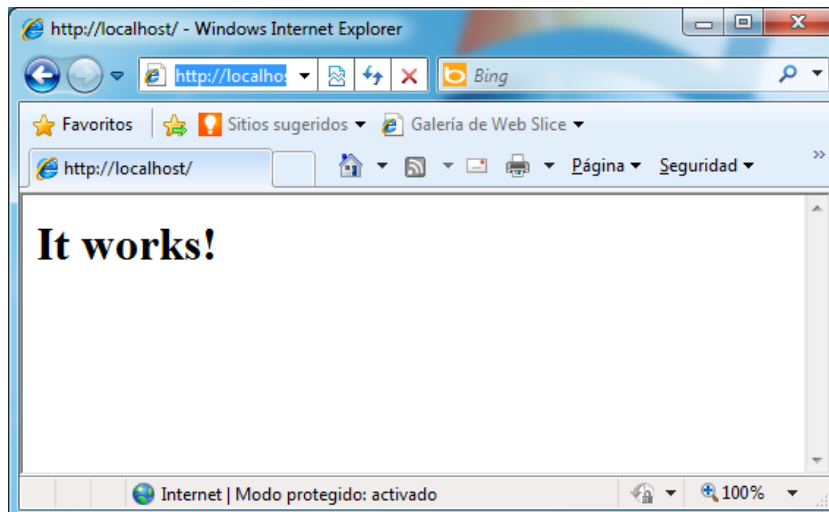


- (5) Finalmente realizar la instalación propiamente dicha

comprobar la instalación

Para comprobar que realmente está el servidor en funcionamiento, basta abrir un navegador y navegar a la dirección <http://localhost>

Si todo va bien, aparecerá el siguiente mensaje:



It works! (funciona) es el texto de la página de inicio del servidor Apache.

modificar el PATH de Windows

Para poder utilizar las aplicaciones de Apache desde la línea de comandos de Windows, es más cómodo que aparezca la ruta a los programas de apache dentro del PATH de Windows.

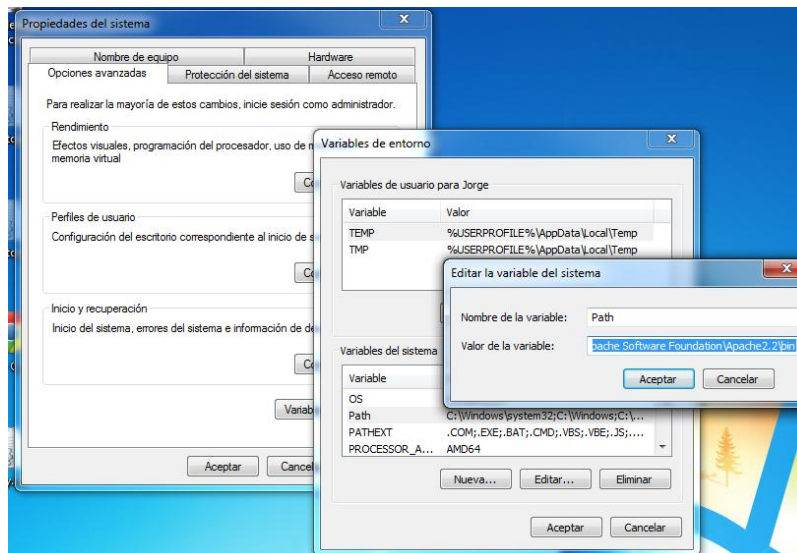
Apache se instala normalmente en una ruta como:

C:\Program Files (x86)\Apache Software Foundation\Apache2.2

Con lo cual la ruta a añadir al PATH para llegar a los binarios es:

C:\Program Files (x86)\Apache Software Foundation\Apache2.2\bin

El PATH del sistema se modifica desde las propiedades del Sistema. Habrá que añadir la ruta de Apache al final.



iniciar y parar el servicio

Apache se instala como servicio de Windows. El arranque y la parada por tanto se pueden realizar desde la pantalla de servicios o bien desde un icono que Apache instala en la barra de tareas al lado del reloj.

Desde la línea de comandos se pueden también hacer estas tareas (y otras):

- Apache se instala como servicio en Windows con **httpd -k install**
- Se inicia con **net start apache2.2** o **httpd -k start**
- Se para con **net stop apache2.2** o **httpd -k stop**
- Se para también con **httpd -k shutdown**
- Se reinicia con **httpd -k restart**
- Y se puede desinstalar como servicio con **httpd -k uninstall** (o con **sc delete apache**)

estructura de directorios de apache

En el directorio raíz de Apache en Windows, que estará en la carpeta en la que se almacenan las aplicaciones de 32 bits (normalmente **C:\Program Files (x86)\Apache Software Foundation\Apache2.2**) se pueden apreciar estas carpetas

Carpeta	Contenido
bin	Archivos ejecutables
cgi-bin	Scripts de servidor
config	Archivos de configuración de Apache

Carpeta	Contenido
error	Archivos que contienen los mensajes de error del servidor (en varios idiomas)
htdocs	Directorio en el que se guardan las páginas web
icons	Carpeta que contiene los iconos que usa el servidor para mostrar en algunos de sus mensajes
logs	Archivos de información sobre conexiones y errores acaecidos
manual	Contiene el manual de Apache
modules	Módulos y extensiones de Apache

(1.3.3) instalación de código fuente en Linux

En principio estos pasos valdrían para cualquier distribución Linux. La ventaja es que conociendo esta instalación, no importará bajo qué distribución Linux estemos; la desventaja es que es compleja y se puede fallar en alguno de los pasos y además no nos ayudará ninguno de los asistentes.

- (1) Se requiere que estén instaladas las herramientas de compilación C/C++ (gcc-c++ es el más popular) y de construcción. Lo habitual es que ya lo estén, sino necesitamos instalar los paquetes correspondientes.
- (2) Hay que ir a la dirección <http://httpd.apache.org/download.cgi> y descargar uno de los archivos comprimidos (normalmente se usa el de formato **gz**).
- (3) Hay que descomprimir el archivo con el comando:

```
tar -vzxf nombreadarchivo.tar.gz
```

o bien desde el entorno gráfico de Linux es fácil que pulsando el botón derecho en el archivo dispongamos de posibilidades para descomprimir

- (4) Configurar la instalación. Para ello dentro del directorio descomprimido de Apache basta con ejecutar el archivo **configure** (mediante el comando **./configure** si estamos en ese directorio en la línea de comandos). Usado sin parámetros, ese comando instala (compilando el código fuente) Apache en el directorio **/usr/local/apache2** con las opciones de instalación por defecto.

Pero el comando **configure**, admite parámetros que permiten configurar toda la instalación. La opción más habitual es utilizar el parámetro **--prefix** que permite indicar el directorio raíz en el que se instalará Apache, pero hay muchas más posibilidades de configuración (visibles a través del comando **configure --help**)

Un parámetro casi obligatorio de uso si se desea compilar con módulos de PHO es **--enable-so**

Ejemplo de uso de **--prefix** (estando en el directorio descomprimido de Apache):

```
./configure --prefix=/sw/web/apache2 --enable-so
```

Usado con más parámetros permite decidir todos los aspectos de la instalación. Para ver dichos parámetros basta con escribir **configure --help** en la línea de comandos

- (5) Compilar el código. Basta con ejecutar el comando **make**. Tardará unos cuantos minutos la compilación. Después se ejecuta el comando **make install** y así se produce ya la instalación en base a la configuración indicada.

modificar el PATH de Linux

Es interesante cambiar el PATH de Linux al igual que el de Windows para que la carpeta **bin** de Apache se encuentre en el PATH y así buscar los comandos de Apache allí. Para ello basta encadenar la ruta a la carpeta bin de Apache en el PATH, mediante:

```
export PATH=$PATH:/usr/local/apache2/bin
```

Suponiendo que **/usr/local/apache2** es la ruta a la raíz de instalación de Apache. Para que esos cambios sean permanentes y al iniciar sesión dispongamos siempre de ellos, esa línea debería estar en el archivo **.bash_profile** o en el **.bash_rc** de nuestro usuario.

Si el cambio es para todos los usuarios, se modifica el archivo **/etc/profile**

lanzar el servidor Apache

Una vez instalado podremos ejecutar el servidor Apache siempre que queramos gracias al programa **apachectl** que se encuentra en la carpeta **bin** de la instalación de Apache. Tiene estas posibilidades (se requieren permisos de administrador):

- **httpd -k start** ó **apachectl start**. Inicia el proceso httpd; es decir ejecuta el servidor Apache.
- **httpd -k stop** ó **apachectl stop**. Para el proceso; detiene el servidor.
- **httpd -k restart** ó **apachectl restart**. Reinicia el servidor.
- **apachectl help**. Ayuda para ver todas las opciones del programa.
- **httpd** sin parámetros, lanza el servidor
- **cat rutaACarpetaLogsDeApache/httpd.pid | xargs kill**. Elimina el proceso cuyo identificador (**pid**) se encuentra en el archivo **httpd.pid** (dentro de la carpeta log de Apache). Es decir elimina el proceso de Apache. El archivo **httpd.pid** se crea cada vez que se ejecuta Apache precisamente para grabar el identificador de proceso.

(1.3.4) instalación del paquete desde el sistema Linux

Otra posibilidad de instalación en Linux es instalar el paquete creado para esa distribución. En las dos distribuciones más populares las de tipo **Red Hat** (como **Fedora**) o **apt-get** y las **Debian** (como **Ubuntu**) se instala de forma distinta.

instalar paquete en sistemas Red Hat

Actualmente la utilidad más habitual para instalar un paquete es **yum** mediante:

```
yum install httpd
```

También podemos instalar el paquete desde el entorno gráfico del sistema.

En cualquier caso, tras la instalación todo estará configurado con las opciones habituales. La instalación se realiza en **/usr/local/apache2** y se puede desde ese momento manejar como en el caso del código fuente.

instalar paquete en sistemas Debian

En el caso de Ubuntu y el resto de sistemas Debian, se usa actualmente el comando **apt-get** desde la línea de comandos. Sería:

```
apt-get install apache2
```

Eso mismo se puede hacer desde el gestor gráfico de paquetes (por ejemplo **Synaptic**).

El problema es que la instalación es diferente:

- **/etc/apache2** es el directorio de instalación de Apache
- **apache2.conf** es el archivo de configuración (que hará referencia a httpd.conf)
- **/usr/sbin** contiene los ejecutables
- No hay archivo **httpd**, todo se maneja con **apachectl**

Aunque la gestión es sencilla (quizá más que en las otras instalaciones, sus tremendas diferencias son una desventaja para el aprendizaje.

(1.4) configuración de Apache

(1.4.1) opciones de httpd

El programa que lanza apache es **httpd** (también se usa **apachectl** en Linux, pro no tiene las mismas opciones, además apachectl acaba invocando a httpd). Al ejecutar httpd se pueden utilizar estas opciones:

opción	significado
-k start	Lanza el servidor Apache
-k stop	Para el servidor Apache
-k restart	Reinicia el servidor Apache
-D nombre	Define un nombre para las directivas <IfDefine name>
-d directorio	Permite indicar un directorio raíz alternativo para Apache
-f rutaArchivo	Permite indicar un archivo alternativo de configuración
-C "directiva"	Procesa la directiva indicada antes de leer la configuración
-c "directiva"	Procesa la directiva indicada después de leer la configuración
-v	Muestra la versión de Apache
-V	Muestra las opciones de compilación
-h	Ayuda para conocer las opciones de httpd .

opción	significado
-l	Lista de módulos compilados
-L	Lista de directivas
-t	Ejecuta el analizador de sintaxis para los archivos de configuración de Apache
-T	Igual pero no comprueba la sintaxis

La ubicación de **httpd** depende de la instalación (como se ha comentado anteriormente); lo más habitual es que se encuentre en la carpeta **bin** dentro de la raíz de instalación de Apache.

(1.4.2) instalar Apache como servicio al arranque

Windows

Si la instalación se hace con el asistente, esa situación ya ocurre. Si no fuera así por la forma de instalar o por cualquier otra razón. Bastaría con:

httpd -k install -n nombreDeServicio

Si no deseamos indicar un nombre de servicio, entonces no se usa el parámetro **-n** y el nombre del servicio será **Apache**.

Linux

En ese caso hay que:

- (1) Copiar **httpd** al directorio **/etc/init.d**. Por ejemplo:

cp /usr/local/apache2/bin/httpd /etc/init.d/apache

apache es el nombre que se da al servicio

- (2) Ejecutar una de estas dos opciones:

- Versión Debian:

update-rc.d apache defaults

- Versión Red Hat

chkconfig apache on

- En otros casos habrá que manualmente copiar un enlaces simbólico con el nombre adecuado a las carpetas **rcX.d**
En ese caso conviene conocer perfectamente cómo se configurar manualmente las opciones de inicio del sistema.

(1.4.3) archivos de configuración

Para modificar el funcionamiento de Apache, se utilizan sus archivos de configuración. El principal es **httpd.conf** (en algunas instalaciones es **apache2.conf**) que se encuentra en el directorio **conf** de la instalación de **Apache**. Puede haber otros, pero lo habitual es usar ese como principal. En cualquier caso un archivo de configuración consta de:

- **Directivas**. Se trata de una palabra clave a la que se sigue un valor. Por ejemplo:

listener 80

Que usa el puerto 80 de comunicaciones. La lista de directivas se puede mostrar con -L

- **Secciones.** Se trata de las partes del documento, para así hacer que las directivas sólo se apliquen a la sección a la que se refiere la sección. Usan etiquetas estilo XML. Por ejemplo:

```
<Directory /usr/local/apache/htdocs/dir2>  
    Deny from all  
    Allow from 192.168.4.21  
</Directory>
```

- **Comentarios.** Comienzan por el signo #. Sirven para documentar el archivo.

ejemplos de directivas básicas

- **Listen.** Permite modificar el puerto por el que se comunica el servidor
- **ServerName.** Nombre del servidor (para probar se suele utilizar **localhost**)
- **ErrorLog.** Ruta al archivo LOG de errores que permite examinar los problemas acaecidos en el servidor Apache.

```
ErrorLog "logs/errorlog"
```

- **LogLevel.** Permite indicar qué eventos se almacenan en el archivo LOG de errores. Hay estas posibilidades (en orden decreciente de significación crítica, mas alto menos evento se almacenan, sólo los):

- **emerg.** Sólo almacena eventos que impiden el desarrollo del servidor
- **alert.** Si hay que tomar acciones inmediatamente.
- **crit.** Si hay fallos críticos
- **error.** Si ocurren condiciones de error
- **warn.** Avisos, no significan errores, sólo advertencias de posibles errores.
- **notice.** Cualquier evento que tenga cierta significancia. Si se elige este nivel se almacena cualquier evento de los anteriores.

- **DocumentRoot.** Indica una ruta que se entiende que será la raíz de los documentos de Apache

- **DirectoryIndex.** Indica el nombre del archivo que se mostrará cuando se acceda a un directorio del servidor. Por ejemplo con **localhost/imágenes/**. En ese caso se mostrará el contenido del directorio si no existe el archivo índice del directorio (normalmente index.html). Ejemplo:

```
DirectoryIndex index.html index.txt index.pl
```

archivos .htaccess

Tienen las mismas posibilidades que el archivo **httpd.conf**, pero sólo se aplica su configuración al directorio en el que están colocados

(1.5) Instalación de MySQL

(1.5.1) pasos previos

Antes de instalar **MySQL** (al igual que otro software cuya instalación sea crítica) se deben tomar estas decisiones:

- Comprobar hardware mínimo necesario
- Decidir la distribución. MySQL está disponible para numerosas plataformas, hay que decidir para cuál nos interesa, en base al precio del sistema, fiabilidad, buena integración, etc.
- Decidir el formato de la distribución. Hay dos posibilidades:
 - **Distribución binaria.** Se trata de una forma más fácil y rápida de instalar. Puede ser a través de un instalador preparado o bien a través de binarios genéricos comprimidos. En el último caso, simplemente con descomprimir y realizar unos cuantos ajustes, tenemos la instalación finalizada
 - **Código fuente.** Se trata de que debemos compilar el código fuente para que funcione el SGBD. Es más complicada de realizar pero permite un mayor control de todos los componentes a instalar, así como preparar un ejecutable óptimo para nuestro sistema o bien incluso modificar el código (que es C y C++).
- Obtener los archivos. De la dirección: <http://dev.mysql.com/downloads>.

(1.5.2) documentación

MySQL dispone de manuales tanto en línea como descargables (por ejemplo en formato PDF) a través de la página <http://dev.mysql.com/doc/>

(1.5.3) instalación en Windows

Es la versión más descargada de MySQL. Hay dos opciones:

- Distribución binaria, que permite iniciar el servidor inmediatamente. Es un archivo **msi**, con las ventajas que eso tiene en un entorno de trabajo Windows. Opciones:
 - Instalador **msi**
 - Instalación comprimida en formato ZIP
- Distribución de código fuente para ser compilada con el compilador **VC++ 6.0**.

instalación con un paquete **msi**

instalación con el asistente

El asistente de MySQL es el programa que permite facilitar la instalación del software. Los pasos a realizar con él son:

- (1) Aceptar el cuadro con la información inicial
- (2) Escoger el tipo de instalación. La opción personalizada (**Custom**) permite elegir más a fondo los componentes a instalar.

(3) Confirmar la instalación

Tras finalizar estos pasos, automáticamente se modifica el registro de Windows, se añade un grupo para MySQL en el menú de Inicio y una carpeta (por ejemplo **MySQL 5.5**) dentro de la carpeta de los programas en el sistema de archivos de Windows (por ejemplo con la ruta **C:\Program Files\MySQL\MySQL 5.5**).

instalar la instancia de MySQL

El sistema MySQL en sí funcionará en cuanto lancemos el asistente para gestionar la instancia. La instancia es el proceso de base de datos que permite el acceso a la base de datos. Eso puede hacerse en el último paso del asistente anterior o bien lanzando el programa en la carpeta **bin** del directorio de MySQL, **MySQLInstanceConfig**.

Una vez lanzado la primera pantalla es esta:

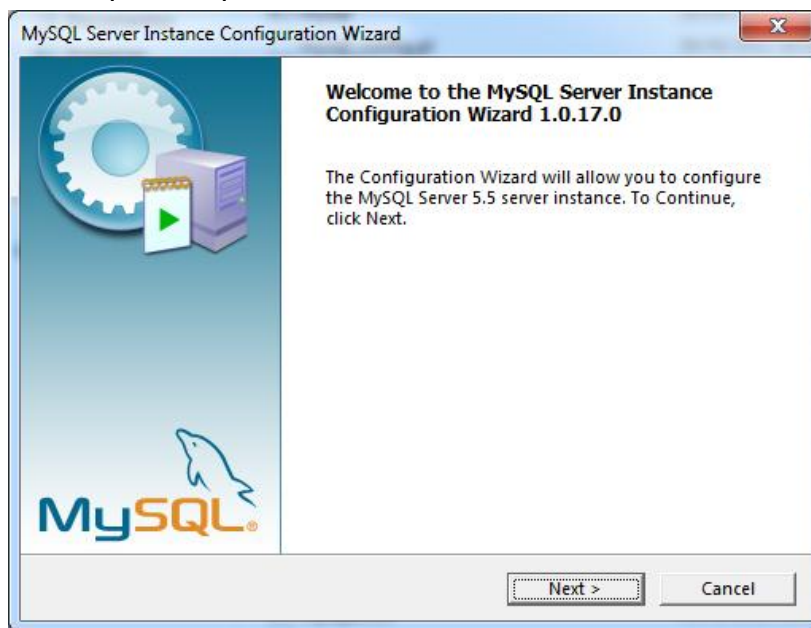


Ilustración 2, Imagen inicial del asistente de configuración de la instancia del servidor MySQL

Con él los pasos a realizar son:

- Elegir el tipo de instalación. La detallada permite más opciones de configuración.
- Elegir el tipo de servidor que deseamos:
 - **Developer Machine**. Máquina de Desarrollo. Para uso personal solamente, consumirá una cantidad mínima de recursos.
 - **Server machine**. Servidor. Se entiende que comparte espacio con otros servidores (web, de correo, ftp). Consume una cantidad moderada de recursos.
 - **Dedicated Machine**. Servidor dedicado. Se elige si sólo se utiliza como servidor a MySQL.
- Uso de la base de datos. En realidad gestiona qué motores de la base de datos se usarán. MySQL dispone de dos opciones **InnoDB** (con buena capacidad de manejo de transacciones, pero más lenta en añadir datos) y **MyISAM**, más

rápida pero con menos control de la integridad de los datos. Las opciones del cuadro son:

- **Multifuncional.** Instala ambos motores y les da la misma potencia.
- **Transaccional.** Instala ambos, pero da preferencia a InnoDB en los recursos a fin de manejar correctamente las transacciones.
- **No transaccional.** Desactiva InnoDB y activa sola MyISAM.

- Elección del directorio donde instalar los datos. Lo que se conoce como la ubicación del **tablespace** de datos. Sólo se usa con InnoDB y permite colocar los datos en otra ubicación respecto a los del programa MySQL o incluso en otra unidad de disco.
- Elección del número de conexiones concurrentes. La primera opción es para bases de datos con escaso número de conexiones concurrentes, la segunda pensada para transacciones en línea (OLTP) y la tercera permite editar a mano el número de conexiones concurrentes.
- Selección del número de puerto (normalmente el 3306) y elegir si deseamos el modo estricto de trabajo (es lo recomendable).
- Elección del juego de caracteres que se utilizará para codificar el texto de la base de datos.
- Elección del modo de trabajo del servicio. Normalmente MySQL se instala como un servicio de Windows más, pero podríamos no desearlo. Se puede cambiar el nombre al servicio.

En este mismo paso se puede modificar el **PATH** de Windows para incluir el directorio **bin** en el **path** y así desde la línea de comandos utilizar los comandos mysql sin tener que modificar nuestra ruta.

- El siguiente paso es muy importante. Se DEBE de cambiar la contraseña del usuario **root**, del superadministrador, por seguridad. Además se puede elegir si permitimos los accesos de este usuario de forma remota (desde otro ordenador) y si activamos la cuenta anónima (acceso sin contraseña), que no es recomendable.
- Tras esos pasos comienza la instalación en sí. Si todo va bien aparece un mensaje como éste:

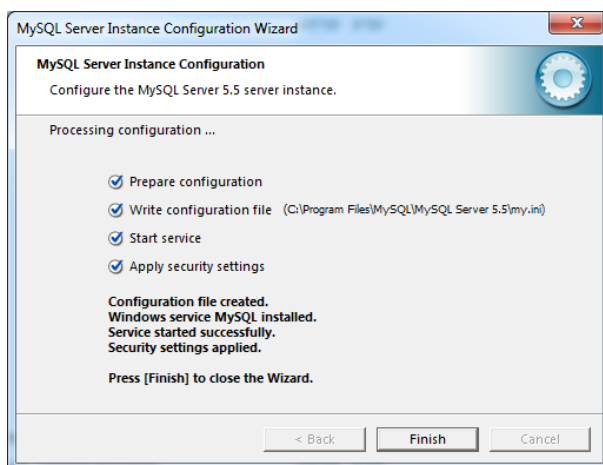


Ilustración 3, Cuadro de diálogo que aparece cuando la instalación ha concluido

En el cuadro último, se indica la ruta del archivo de configuración en el que se han guardado los parámetros de instalación. Es el archivo **my.ini** situado en la carpeta de instalación de MySQL. Examinar ese archivo es interesante y modificarle permitiría cambiar el funcionamiento de MySQL independientemente de lo que se haya especificado en la instalación.

comprobar funcionamiento de MySQL

Necesitamos probar la instalación. En Windows, MySQL crea las tablas de privilegios y usuarios sin tener que configurar más. Además creará tres bases de datos y diversas tablas en ellas como ejemplo.

MySQL será un servicio que se iniciará automáticamente en el arranque. De no desearlo así debemos ir a la pantalla de **Servicios** de Windows (dentro de la carpeta de herramientas administrativas) y parar el servicio además de indicar que no deseamos su inicio automático (en las propiedades del servicio **MySQL**).

Para detener el servicio, basta ir a la línea de comandos e indicar el comando:

```
net stop MySQL
```

Si deseamos iniciar de nuevo el servicio:

```
net start MySQL
```

Si el servicio está funcionando, para probar el funcionamiento basta con conectar con MySQL desde la línea de comandos, por ejemplo con:

```
mysql -u root -p
```

Después se nos solicitará la contraseña del administrador. Y entonces estaremos viendo el Shell de MySQL (la pantalla que interpreta comandos MySQL). Por ejemplo ejecutando el comando **show databases**, se nos mostrarán las bases de datos ejemplo de MySQL.

En el caso de que el servidor no arranque con el comando net, podemos lanzar MySQL invocando al programa (demonio) **mysqld**

desinstalar MySQL en el caso de instalación con el paquete msi

Para ello basta con ir al panel de control, localizar MySQL y desinstalar. Pero eso elimina sólo los archivos de MySQL que se crearon en la instalación, los archivos de configuración y datos permanecerán donde se crearon y eso puede causar problemas en otra instalación. Para desinstalar completamente:

- (1) Desinstalar desde el panel de control
- (2) Eliminar el contenido de **C:\Documents and Settings\All Users\Application Data\MySQL**, o bien en Windows 7 o Windows Server 2008 **C:\ProgramData\MySQL** (**ProgramData** es una carpeta oculta que deberemos mostrar con la opción de mostrar carpetas y archivos ocultos de Windows)
- (3) Revisar la carpeta de instalación de MySQL y comprobar que no ha quedado ningún archivo sin borrar

instalar MySQL desde un fichero ZIP

En ese caso tenemos un archivo comprimido que contiene la instalación de MySQL, simplemente habrá que configurar la instalación. Los pasos son:

- (1) Extraer el fichero ZIP en la ubicación deseada
- (2) Crear un fichero de opciones con al menos las líneas donde colocar la ubicación de MySQL y los datos. Aunque hay varias posibilidades de colocar el archivo de configuración (como se describirá más adelante), lo habitual es llamarle **my.ini** y colocarlo en la carpeta de instalación de Windows (por ejemplo **C:\Windows**).

Conviene copiar uno de los archivos de configuración que se encuentran en la carpeta raíz de MySQL (por ejemplo **my-medium.ini**), cambiarle el nombre a **my.ini** y modificar las líneas deseadas.

Ejemplo de ello es por ejemplo indicar cuál es el directorio raíz de MySQL y la carpeta donde se almacenará la base de datos. Para ello se añadirían las siguientes líneas a la sección **mysqld**:

```
[mysqld]
..... otros parámetros de configuración
# coloca en basedir el directorio de instalación
basedir=E:/mysql
# coloca en datadir el directorio de datos
datadir=E:/mydata/data
```

Como se ve en el ejemplo MySQL usa las barras de los directorios al estilo Unix (/ en lugar de \). Para que la ruta cuadre con el archivo de opciones es necesario asegurar que la raíz de los archivos de MySQL se encuentra donde se ha marcado en el archivo de opciones.

Nota: Normalmente el directorio de datos (**data**), se encuentra debajo de la raíz de instalación de MySQL, **si deseamos utilizar la nueva ubicación deberemos copiar el contenido de data a esa nueva ubicación.**

- (3) Colocar la ruta al directorio **bin** de MySQL dentro de la variable PATH de Windows.
- (4) Lanzar la instancia de la base de datos desde la línea de comandos escribiendo:
mysqld --console.

Tras este comando (si la configuración está bien), se quedará la consola con el mensaje: **Version: '5.5.16' socket: " port: 3306 MySQL Community Server (GPL)** (suponiendo que disponemos de MySQL Community Server versión 5.5.16). El servidor MySQL está lanzado y escuchando por el puerto 3306

- (5) Para el resto de veces bastará con lanzar el demonio de MySQL (mysqld), simplemente escribiendo **mysqld** en la línea de comandos.

arrancar MySQL como un servicio

También es posible en este tipo de instalación arrancar MySQL como un servicio. Basta con:

```
mysqld --install
```

Si no funcionara es posible que el servidor MySQL esté en funcionamiento. Detener implica usar el comando:

```
mysqladmin -u root shutdown
```

Desde este instante tendremos MySQL colocado como servicio de Windows. Disponemos de las opciones de comandos:

■ **mysql --install-manual**. Para instalar MySQL como servicio que debe iniciarse manualmente.

■ **mysql --remove**. Para eliminar MySQL como servicio.

desinstalar si la instalación es desde un ZIP

- (1) Desinstalar el servicio de MySQL si es tal. En todo caso detener la instancia definitivamente de MySQL con **sc delete mysql**
- (2) Si lo anterior no funciona (por ejemplo sino se configuró como servicio), entonces: **mysqladmin -u root shutdown**
- (3) Borrar el archivo de configuración **my.ini**
- (4) Borrar los archivos de MySQL

En realidad en este tipo de instalación, todo es más sencillo porque no se instala MySQL al estilo de las aplicaciones de Windows

El usuario administrador en este tipo de instalación no posee contraseña; habría que cambiarla más tarde.

(1.5.4) instalación en Linux/Unix

instalación binaria genérica

La instalación binaria genérica en los sistemas de tipo Unix es similar a la de Windows mediante archivo ZIP. Este tipo de instalación valdría para cualquier versión de Linux. La ventaja es la comodidad y el hecho de que se maneje igual en todo tipo de Unix/Linux. La desventaja es que esta instalación no está optimizada para la versión de Linux concreta en la que instalamos MySQL.

La instalación creará una carpeta raíz desde la que colgarán todos los directorios de MySQL, concretamente:

directorio	uso
bin	Ubicación de los programas MySQL
data	Ubicación de los datos de las bases de datos
docs	Manual en formato info de Linux
man	Manual en formato del comando man de Unix
include	Cabeceras de código fuente
lib	Archivos de librería
scripts	Contiene el script mysql_install_db
share	Mensajes de error, juegos de caracteres, ejemplos,...
sql-bench	<i>Benchmarks</i>

Los pasos son:

- (1) Descargar el archivo comprimido con la instalación desde <http://dev.mysql.com/downloads/> Será algo así como **Linux Generix xxxx.tar.gz TAR Compressed** donde las equis indican la versión del sistema (por ejemplo versión 2.6 de 32 o 64 bits)

- (2) Abrir la consola de comandos (el **Shell**) con privilegios administrativos y ejecutar la orden:

```
groupadd mysql.
```

Esto añade un grupo para el usuario relacionado con MySQL

- (3) Ejecutar la orden:

```
user add -g mysql mysql.
```

Esto crea un usuario llamado mysql que pertenece al grupo anterior.

- (4) Colocarnos en el directorio que contendrá la instalación de MySQL. Lo habitual es el comando: `cd /usr/local`.

- (5) Descomprimir el archivo con:

```
tar zxvf rutaCompletaAlArchivoTAR.tar.gz.
```

Necesitamos indicar la ruta al archivo **tar** para descomprimirlo en la carpeta en la que nos posicionamos en el punto 4. Es decir si hemos descargado MySQL en la carpeta de descargas, podría ser: `tar zxvf $HOME/Downloads/MySQL-5.5.16-1.Linux.2.6.x86_64.tar.gz`. El resultado es MySQL descomprimido en `/usr/local` (o en el directorio que hayamos indicado en el punto 4).

- (6) Crear un enlace simbólico al directorio de instalación MYSQL para tener un acceso más sencillo. Por ejemplo:

```
cd $HOME
```

```
ln -s /usr/local/mysql-5.5.16-linux.2.6-x86_64 mysql
```

Desde ese momento el comando `cd mysql`, ejecutado desde la carpeta local de usuario, entra en el directorio de instalación de MySQL.

- (7) Ir al directorio raíz de **MySQL** (con `cd mysql` usando el enlace creado en el punto anterior) desde ahí dar propiedad de los archivos de MySQL al usuario root y grupo **mysql** creados anteriormente. Al usuario **mysql** se le da la propiedad de la carpeta data donde se encuentran los datos:

```
chown -R root .
```

```
chown mysql data
```

```
chgrp -R mysql .
```

- (8) Para el paso siguiente necesitamos la librería **libaio1**, si no disponemos de ella hay que instalarla. En los Linux tipo Debian (como Ubuntu) con:

```
apt-get install libaio1 libaio-dev
```

En los Linux tipo Red Hat (como Fedora) con:

```
yum install libaio1
```

- (9) Instalar las tablas iniciales mediante (suponiendo que seguimos en la raíz de MySQL):

```
scripts/mysql_install_db --user=mysql
```

- (10) Iniciar MySQL, pero teniendo cuidado de hacerlo sin ser el usuario **root**. La forma de hacerlo (situados desde la raíz de MySQL):

```
bin/mysqld_safe --user=mysql &
```

Si sale bien el servidor estará funcionando correctamente. En la pantalla de consola, se queda sin salir el **prompt** mientras el servidor siga funcionando.

- (11) Comprobar que realmente MySQL está funcionando con (situados desde la raíz de MySQL):

```
bin/mysqladmin version
```

Si todo ha ido bien, el resultado sería algo como:

```
bin/mysqladmin Ver 8.42 Distrib 5.5.16, for linux2.6 on x86_64
Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Server version          5.5.16
Protocol version 10
Connection              Localhost via UNIX socket
UNIX socket             /tmp/mysql.sock
Uptime:                 1 min 27 sec
```

```
Threads: 1 Questions: 1 Slow queries: 0 Opens: 33 Flush tables: 1 Open tables:
26 Queries per second avg: 0.011
```

- (12) Comprobar que podemos cortar MySQL (situados desde la raíz de MySQL):

```
bin/mysqladmin -u root shutdown
```

Si volvemos a comprobar si MySQL está en pie mediante el comando **mysqladmin versión**, debería salir algo parecido a:

```
bin/mysqladmin: connect to server at 'localhost' failed
error: 'Can't connect to local MySQL server through socket '/tmp/mysql.sock' (2)'
Check that mysqld is running and that the socket: '/tmp/mysql.sock' exists!
```

- (13) Lanzar de nuevo el servidor (situados desde la raíz de MySQL):

```
bin/mysqld_safe --user=mysql
```

- (14) Entrar en MySQL (situados desde la raíz de MySQL):

```
bin/mysql
```

No hay contraseñas para ningún usuario, por lo que es conveniente hacerlo (se explica más adelante)

preparar MySQL para el inicio automático

Si se desea que MySQL se inicie automáticamente en cada encendido del ordenador, bastará con copiar el script **mysql.server** que se encuentra en el directorio **support-files** que, a su vez, está en la raíz de MySQL. La copia se realiza al directorio que contiene los scripts de inicio del sistema, normalmente **/etc/init.d**

El comando a ejecutar sería (estando en la raíz de MySQL):

```
cp support-files/mysql.server /etc/init.d/mysql
chmod +x /etc/init.d/mysql
```

Eso le coloca en los scripts de inicio con el nombre **mysql** y le da permiso de ejecución. Desde ese momento el servicio mysql ya existe. Para lanzarle en cada reinicio existen comandos en los diversos Linux. El más popular:

```
chkconfig --level 345 mysql on
```

Que iniciará el servidor en los niveles 3,4 y 5 del sistema, con el código de secuencia 50.

En el caso de Ubuntu (y de otros Linux tipo Debian), ese comando no existe y actualmente se puede utilizar:

```
update-rc.d mysql defaults
```

que coloca el servicio mysql en los niveles 2,3,4 y 5 y le asigna el código de secuencia 20. Este otro comando es equivalente al **chkconfig** anterior:

```
update-rc.d mysql start 50 3 4 5 . stop 50 0 1 2 6
```

Ambos comandos tienen más opciones que serán fáciles de utilizar para quienes tengan conocimientos sobre cómo funciona el arranque en Linux.

Para que estas opciones funcionen, el archivo de configuración **my.cnf** situado normalmente en **/etc/my.cnf** debería contener estas líneas al menos:

```
[mysqld]
datadir=/usr/local/mysql/data
basedir=/usr/local/mysql
```

La ruta **/usr/local/mysql** en el ejemplo es la que apunta a la raíz de MySQL.

desinstalar MySQL si la instalación es binaria ZIP

Si deseamos desinstalar MySQL del sistema; en este caso basta con parar el servidor, eliminar los servicios relacionados con MySQL (si es necesario) que se inician con el sistema y eliminar la carpeta raíz y todos los subdirectorios (comando **rm -R**)

instalación mediante gestores de paquetes

Casi todos los Linux tienen predefinidos paquetes para instalar MySQL. Es el caso de los paquetes rpm que se pueden instalar fácilmente en cualquier Linux de tipo **Red Hat** (como **Fedora**).

Es la instalación recomendada desde el fabricante, la cuestión es que la localización de los programas varía respecto a la instalación mediante binarios genéricos. Por paquetes MySQL se organiza de la siguiente manera:

directorio	uso
<code>/usr/bin</code>	Ubicación de los programas MySQL y de sus scripts
<code>/usr/sbin</code>	Ubicación del servidor mysqld
<code>/var/lib/mysql</code>	Archivos log y bases de datos
<code>/usr/share/info</code>	Manual en formato info de Linux
<code>/usr/share/man</code>	Manual en formato del comando man de Unix
<code>/usr/include/mysql</code>	Cabeceras de código fuente
<code>/usr/lib/mysql</code>	Archivos de librería
<code>/usr/share/mysql</code>	Raíz de MySQL. Mensajes de error, juegos de caracteres, ejemplos,...
<code>/usr/share/sql-bench</code>	<i>Benchmarks</i>
<code>/etc/apache2</code>	Archivos de configuración y soporte

instalación en Linux tipo RedHat (como Fedora)

Se suele usar la utilidad **yum** (o bien la utilidad gráfica de instalación de paquetes):

```
yum install mysql-server
```

Esto descarga los archivos necesarios y cambia la contraseña del administrador de la base de datos (la pide por teclado).

Una vez instalado se usaría el comando:

```
service mysqld start
```

Lanza el servidor MySQL y lo coloca como servicio del sistema. Si deseamos que sea un servicio que funcione desde el arranque del sistema, se haría:

```
chkconfig --levels 235 mysqld on
```

De esa forma en los niveles de ejecución 2,3 y 5; el demonio de MySQL (el proceso servidor) se arranca desde el inicio. El código:

```
chkconfig --levels 235 mysqld off
```

Elimina del arranque al servicio.

instalación en Linux tipo Debian (como Ubuntu)

En los Linux derivados de **Debian** (como **Ubuntu**) pueden instalar el servidor MySQL con:

```
sudo apt-get install mysql-server
```

También se puede utilizar la herramienta gráfica **Synaptic** para instalar el paquete de servidor de MySQL.

Se pedirá una contraseña administrativa y se habrá arreglado el PATH, se habrá colocado el archivo de configuración en `/etc/mysql/my.conf` y se colocará como servicio de arranque el script `/etc/init.d/mysql` que permite que el servidor puede lanzarse automáticamente durante el arranque.

En todo momento podemos parar el servidor con:

```
service mysqld stop
```


y lanzar con:

```
service mysqld start
```

Además podremos colocar el servicio mysql para que se inicie automáticamente mediante:

```
update-rc.d mysqld defaults
```

(1.5.5) asegurando la instalación

contraseñas de usuarios

Hay que intentar desde el primer momento que la instalación de MySQL no permita que los usuarios puedan sobrepasar sus privilegios y dañar las bases de datos. Por ello hay que asegurar que todos los usuarios (y en especial los administrativos) tienen contraseña.

En las instalaciones en Windows ya existen las cuentas de usuario y se las pone contraseña durante la instalación. Sin embargo algunas instalaciones en Windows (las genéricas) no tienen los usuarios preconfigurados. Para configurar inicialmente los usuarios se lanza el script (comentado anteriormente) `mysql_install_db`. Con ello en la tabla `mysql.user` dispondremos de los usuarios iniciales.

En Windows hay un usuario `root` (superusuario) que tienen privilegios totales pero que solo puede acceder a la máquina local. Para permitir el acceso con esos privilegios desde otra máquina, se crea otro usuario `root` (durante la instalación de MySQL se pregunta esa posibilidad). En Linux los usuarios `root` permiten el acceso local.

Hay cuentas anónimas (no tienen nombre de usuario) algunas para acceder de forma local y otras no. Para asegurar el acceso se deben poner contraseñas a las cuentas anónimas o bien eliminarlas.

Para examinar la lista de usuarios se usa:

```
SELECT user, host, password FROM mysql.user;
```

Para cambiar la contraseña, por ejemplo:

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('Atrezzo');
```

En el ejemplo se cambia la contraseña para el superadministrador `root` cuando accede de forma local.

Otra forma (mejor puesto que modifica de golpe la contraseña de todos los root) es:

```
UPDATE mysql.user SET password=PASSWORD('Atrezzo')  
WHERE user='root';  
FLUSH PRIVILEGES;
```

La última instrucción es la que hace que el cambio se lleve a cabo.

eliminación de la base de datos ejemplo

En muchas instalaciones de MySQL se dispone de una base de datos llamada **test** que, en realidad, es para pruebas. Dejarla puede proporcionar un agujero de seguridad, por lo que si no deseamos usarla para aprender se debe de eliminar con:

```
DELETE FROM mysql.db WHERE Db LIKE 'test%';  
FLUSH PRIVILEGES;
```

La última orden (**FLUSH PRIVILEGES**) confirma los cambios.

(1.6) PHP

(1.6.1) ¿qué es PHP?

Actualmente significa **Hipertext Pre Processor** y se trata del **lenguaje de scripts de servidor** más popular.

Un lenguaje de scripts, en general, es un lenguaje cuyo código se incrusta dentro de otro. Es el caso de **JavaScript** que es un lenguaje que va incrustado dentro del código HTML de una página web. Pero, en el caso de JavaScript está en el lado del cliente; es decir es el navegador de Internet el que tiene que tener la capacidad de interpretar el código del lenguaje script, además del HTML.

Eso provoca una desventaja: los navegadores tienen que tener capacidades añadidas y un nuevo lenguaje de ese tipo implica nuevos plugins para los navegadores, con el riesgo de que algunos usuarios naveguen por las páginas web sin poder verlas adecuadamente porque su navegador no tiene instalado el plugin.

Por ello los lenguajes de scripts de servidor han tenido mucho éxito. En ese caso, es el servidor el que interpreta el lenguaje script y devuelve al navegador el resultado de interpretar dicho lenguaje, que siempre es HTML.

PHP es gratuito y software de código abierto que tiene una relación excelente con Apache, MySQL y Linux; aunque actualmente en Windows también se instala muchísimo.

PHP se puede instalar también en servidores **IIS** de Microsoft y en otros muchos; además puede utilizar sistemas de bases de datos como Oracle, Informix, DB2,...

Los archivos de instalación de PHP están disponibles en www.php.net/download.php. En cualquier caso las posibilidades de instalación son las habituales:

- Mediante archivos binarios. Que tienen estas dos posibilidades
 - Un instalador msi (**Windows**) o instalación por paquetes (**Linux**)
 - Binarios comprimidos (**ZIP** o **TAR**)
- Instalar por código fuente

(1.6.2) instalación en Windows

Desde el sitio <http://windows.php.net/download/> se descarga el archivo msi (o los archivos binarios ZIP) deseados. Es **muy importante leer la columna izquierda para saber qué versión de PHP bajarse en función de nuestro servidor web**.

Para Apache desde esa página (actualmente) habría que bajarse las versiones VC6 que están optimizadas para él. Se comenta la instalación mediante **binario ZIP**:

- (1) Descomprimir los binarios en la ubicación deseada (por ejemplo [C:\PHP](#))
- (2) Añadir la ruta a la carpeta raíz de PHP y a la carpeta Ext (dentro de la anterior, normalmente) como parte de la variable PATH del sistema.
- (3) En la carpeta de instalación de PHP, copiar **php.ini-recommended.ini** a **php.ini**. Eso usa un archivo de configuración preconstruido, que después se puede modificar. **php.ini** es el archivo de configuración principal.
- (4) Buscar y modificar (o añadir si no se encuentran) estas líneas al archivo **php.ini**:

```
; Muestra errores
display_errors = On
;Especificar la ruta de las extensiones de PHP
extension_dir = "./ext"
;Habilitar el soporte de MySQL
extension=php_mysql.dll
extension=php_mysqli.dll
```

- (5) Editar el archivo **httpd.conf** de Apache y añadir estas líneas (suponiendo que PHP está instalado en [c:\PHP](#)):

```
PHPIniDir "C:/php"
LoadModule php5_module "c:/php/php5apache2_2.dll"
AddType application/x-httpd-php .php
```

- (6) Probamos el servicio (puede ser bueno incluso reiniciar el ordenador)
- (7) Crear este archivo en la carpeta **htdocs** de Apache (si esa sigue siendo la carpeta la que se almacenan las webs que sirve Apache) y llamarle **prueba.php**:

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

- (8) Arrancar Apache y navegar a <http://localhost/prueba.php>

(1.6.3) instalar en Linux

instalar con código fuente

- (1) Descargar la instalación en forma de archivo **tar.gz**
- (2) Descomprimir desde el sistema gráfico de Linux (normalmente usando la opción **Extraer aquí**) o desde la línea de comandos usando el comando:

```
tar xzvf archivoPHP.gz.zip
```

- (3) Crear el directorio en el que se instalará PHP (por ejemplo [/usr/local/php](#))
- (4) Puede ser necesario instalar el paquete **libxml2-dev**. En Linux tipo Debian (como Ubuntu) se hace:

```
apt-get install libxml2-dev
```

En Linux

```
yum install libxml2-devel)
```

Para compatibilidad con MySQL habría que instalar los paquetes relacionados, en especial **mysql-client**

- (5) Entrar en la carpeta descomprimida anteriormente y ejecutar el comando **configure**. Este comando sirve para configurar la instalación, en él hay que indicar dónde está el directorio **apxs**. Normalmente está dentro del directorio **bin** en la raíz de Apache (por ejemplo **/usr/local/apache2/bin/apxs**).

La opción **--with-mysql** da soporte para MySQL y la opción **--prefix** permite elegir dónde se realizará la instalación (si no se usa **prefix**, se instala en **/usr/local/php**):

```
./configure --with-apxs2=raízApache/bin/apxs --with-mysql=raízMySQL  
--prefix=raízPHPDeseada
```

- (6) Compilar (desde dentro de la carpeta anterior):

```
make  
make install
```

- (7) Copiar el archivo **php-development** (que está en el directorio descomprimido de instalación) a **/usr/local/lib/php.ini** para convertirlo en el archivo de configuración **php.ini**. Dicho archivo podría estar en otra ubicación si en el paso 5 (uso del comando **./config9** se hubiera hecho uso de la opción:

```
./config .... --with-config-file-path=rutaDeseadaParaPHP.ini
```

La copia se haría con (suponiendo que seguimos dentro de la carpeta descomprimida con la instalación de PHP):

```
cp php.ini-development /usr/local/lib/php.ini
```

- (8) Añadir estas líneas al archivo general de configuración de Apache **httpd.conf**:

```
LoadModule php5_module modules/libphp5.so  
AddType application/x-httpd-php .php  
AddType application/x-httpd-php .phtml
```

La más importante es la primera; se supone que la orden **make install**, coloca el módulo de PHP en el directorio modules de Apache, pero conviene comprobarlo para asegurarse que funcionan esas configuraciones.

- (9) Reiniciar el servidor Apache

- (10) Crear un documento php en el directorio de documentos de Apache

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

(11) El resultado deberá ser algo como:

PHP Version 5.3.8	
System	Linux ubuntu 2.6.32-33-generic #70-Ubuntu SMP Thu Jul 7 21:09:46 UTC 2011 i686
Build Date	Oct 13 2011 19:53:46
Configure Command	'./configure' '--with-apxs2=/sw/servidores/apache2/bin/apxs' '--prefix=/sw/servidores/php'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/sw/servidores/php/lib
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,NTS
PHP Extension Build	API20090626,NTS
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
Registered PHP Streams	php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.3.0, Copyright (c) 1998-2011 Zend Technologies

Powered By
Zend Engine 2

instalar con paquetes

Como en el caso de Apache y MySQL, es posible instalar PHP mediante gestores de paquetes. Los nombres varían en cada sistema, por ejemplo en Debian sería:

```
apt-get install libapache2-mod-php5
```

(1.7) paquetes completos Apache/MySQL/PHP

Hay varias posibilidades de instalar paquetes con Apache, MySQL y PHP ya perfectamente configurado. La ventaja es la facilidad de instalación, la desventaja es que suelen facilitar la instalación a cambio de hacerla más insegura.

(1.7.1) AppServ

Es quizá la opción más potente instala Apache, PHP, MySQL y el gestor phpMyAdmin para administrar MySQL con miras a PHP. Se descarga de:

<http://www.appservnetwork.com/>

Desde ahí se descarga un ejecutable que, al lanzarse, realiza paso a paso la instalación.



Ilustración 4, Pantalla inicial de AppServ

Tras breves preguntas sobre, el directorio de instalación (normalmente **C:\AppServ**) la configuración de Apache (nombre del servidor y puerto) y la configuración de MySQL (contraseña de root).

Todos los comandos de parada y encendido, así como la configuración de **httpd.conf**, **my.ini** y **php.ini** (los archivos de configuración de las tres aplicaciones) funcionan igual que en cualquier instalación; pero además disponemos de un grupo de programas en el menú inicio que nos facilita casi todas las tareas con los servidores.

directorios de AppServ

El problema de estas instalaciones en paquete a veces es entender la nueva configuración de directorios. Pero en este caso es fácil, de la raíz de instalación cuelgan:

- **Apache 2.2.** Raíz de Apache
- **MySQL.** Raíz de MySQL.
- **php5.** Raíz del módulo de PHP
- **www.** Todas las carpetas anteriores tienen la configuración habitual de cada aplicación. Pero **www** es la carpeta (en lugar de la habitual **htdocs** de Apache) que aloja las páginas a publicar. Es decir, es el **DocumentRoot** de Apache.

(1.7.2) XAMPP

Se trata de la instalación más popular de estos servicios, es multiplataforma; pero es sobre todo utilizada en Linux. Instala Apache, MySQL, PHP, **Perl** e incluso un servidor **FTPD** además de otras muchas herramientas (por ejemplo Open SSL), lo que la hace muy versátil. Se descarga de

<http://www.apachefriends.org/es/xampp.html>

Una vez descargada los pasos para instalarla son:

- (1) Extraer el archivo descargado al directorio **/opt** (con permisos de **root**). Por ejemplo:

```
tar xvfz xampp-linux-1.7.7.tar.gz -C /opt
```

- (2) Lanzar los servidores con:

```
/opt/lampp/lampp start
```

- (3) Comprobar la instalación usando un navegador y escribiendo <http://localhost> La página que aparece permite ver la documentación y configurar la aplicación.



seguridad

La seguridad de la instalación es baja, de hecho ocurren estos detalles:

- El administrador de MySQL no tiene contraseña
- El servidor de MySQL es accesible por la red
- El servidor FTP que viene con el paquete (ProFTP) tiene un usuario y contraseña conocido: usuario **nobody** contraseña **lampp**
- El gestor visual **PhpMyAdmin** es accesible por la red
- Las páginas de ejemplo son accesibles.
- Tanto MySQL como Apache usan el mismo usuario al ejecutarse (usuario **nobody**)

El propio XAMPP tiene una opción que reduce el riesgo antes de que configuremos adecuadamente la seguridad, es lanzarse de esta forma:

```
/opt/lampp/lampp security
```

Que protege más la instalación (usando contraseñas aleatorias y otros usuarios creados para la ocasión).

otras opciones

El comando **lampp** dispone de estas opciones:

Opción del comando	Significado
stop	Detiene todo XAMPP
restart	Detiene todo XAMPP y lo vuelve a lanzar
startapache	Lanza sólo Apache
startssl	Activa el protocolo seguro SSL para todas las ejecuciones futuras de Apache
startmysql	Lanza sólo MySQL
startftp	Lanza el servidor FTPD que permite transmitir ficheros con el usuario inicial nobody y la contraseña lampp
stopapache	Detiene Apache
stopssl	Desactiva el protocolo seguro SSL para todas las ejecuciones posteriores del servidor Apache
stopmysql	Detiene MySQL
stopftp	Detiene FTPD. Si se lanza todo XAMPP con start , FTPD no se ejecutará
security	Programa para probar el sitio web XAMPP con total seguridad

directorios

La organización de directorios es:

Directorio	Uso
/opt/lampp/bin/	Raíz de los programas ejecutables de las aplicaciones. Así: <code>/opt/lampp/bin/mysql</code> lanza el cliente de MySQL
/opt/lampp/htdocs/	Raíz de las páginas web alojadas en Apache. Es decir DocumentRoot de Apache.
/opt/lampp/etc/httpd.conf	Configuración de Apache
/opt/lampp/etc/my.cnf	Configuración de MySQL
/opt/lampp/etc/php.ini	Configuración de PHP
/opt/lampp/etc/proftpd.conf	Configuración de ProFTPD
/opt/lampp/phpmyadmin/config.inc.php	Configuración de PhpMyAdmin

(1.7.3) Wampserver

Descargable desde <http://wampserver.com> es una de las más populares, sólo disponible para Windows. Simplemente se descarga, se siguen las instrucciones y se instala con parámetros por defecto que, después, podríamos modificar. Es decir al estilo de las dos anteriores.