



estudios abiertos

SEAS

GRUPO SAN**VALERO**

1
UNIDAD
DIDACTICA

PHP

1. Instalando Apache, PHP y MySQL

ÍNDICE

OBJETIVOS	11
INTRODUCCIÓN	12
1.1. Programas necesarios	13
1.1.1. Apache	13
1.1.2. MySQL	17
1.1.3. PHP	19
1.2. Instalación paquete AMP	23
1.2.1. Instalación WAMP	23
1.2.2. Instalación LAMP	29
1.2.3. Instalación MAMP	32
RESUMEN	37

OBJETIVOS

- Comentar cuáles son las principales funciones de un servidor de páginas web.
- Conocer Apache y cuáles son sus principales ventajas frente a otras soluciones.
- Comentar qué es un sistema gestor de bases de datos y cuáles son sus objetivos.
- Conocer MySQL y cuáles son sus principales características.
- Conocer PHP como lenguaje de programación y sus principales características.
- Instalar y configurar paquetes de instalación completa Apache+MySQL+PHP para plataformas Windows, Linux y Mac OS.



- Servidor de Páginas web Apache.
- Intérprete PHP.
- MySQL como gestor de bases de datos.

Comentaremos sus principales características y ventajas, para a continuación pasar a comentar cómo se relacionan entre todas para conseguir que podamos programar páginas web dinámicas.

Realizaremos la instalación de aplicaciones AMP para las tres plataformas más utilizadas: Windows, Linux y Macintosh.

1.1. Programas necesarios

A lo largo de todo este libro, vamos a necesitar diferentes softwares, con diferentes objetivos cada uno de ellos. Aun así, entre los tres conseguiremos un objetivo común: diseño e implementación de páginas web dinámicas.

Las herramientas que vamos a emplear van a ser las siguientes:

- Apache. Como servidor de páginas web.
- MySQL. Como gestor de bases de datos.
- Intérprete PHP. Mediante este intérprete instalado en el servidor Apache y mediante consultas a las bases de datos MySQL podremos conseguir crear páginas web dinámicas, utilizando su lenguaje de programación.

1.1.1. Apache

Apache es un servidor HTTP, de código abierto y licenciamiento libre, que funciona en Linux, sistemas operativos derivados de Unix™, Windows, Novell Netware y otras plataformas. Ha desempeñado un papel muy importante en el crecimiento de la red mundial, y continúa siendo el servidor HTTP más utilizado, siendo además, el servidor de hecho contra el cual se realizan las pruebas comparativas y de desempeño para otros productos competidores. Apache es desarrollado y mantenido por una comunidad de desarrolladores auspiciada por *Apache Software Foundation* (<http://www.apache.org>).



Figura 1.1. Logo de Apache.

Antes de seguir introduciendo Apache, será conveniente explicar qué significa ser un servidor HTTP.

HTTP (*Hypertext Transfer Protocol*, o Protocolo de Tránsito de Hipertexto), es el protocolo utilizado para transferir o transportar información en Internet (WWW, *World Wide Web*). Su propósito original fue el proveer una forma de publicar y recuperar documentos HTML.

El desarrollo del protocolo fue coordinado por *World Wide Web Consortium* y la IETF (*Internet Engineering Task Force*, o Fuerza de Trabajo en Ingeniería de Internet), culminando con la publicación de varios RFC (*Request For Comments*), de entre los que destaca el RFC 2616.

HTTP es un protocolo de solicitud y respuesta a través de TCP, entre agentes de usuario (navegadores, motores de índice y otras herramientas) y servidores, regularmente utilizando el puerto 80. Entre la comunicación entre éstos pueden intervenir como servidores Intermediarios (Proxies), puertas de enlace y túneles.

Volviendo a Apache, en primer lugar, tenemos que destacar el gran auge que ha tenido en tan corto espacio de tiempo, y como hoy en día, según las estadísticas, es empleado en el 60% de soluciones de servidores http en el mundo, desbancando a sus principales rivales como es el caso de Microsoft, con su software Microsoft IIS (*Internet Information Server*).

Si observamos una de las principales empresas que realizan este tipo de encuestas, www.netcraft.com, y miramos los datos actualizados a Junio 2011, no hay lugar a dudas. Veamos primero un histórico de diferentes soluciones de servidores HTTP a lo largo de los años para comprobar su rápido crecimiento y posterior mantenimiento del liderazgo:

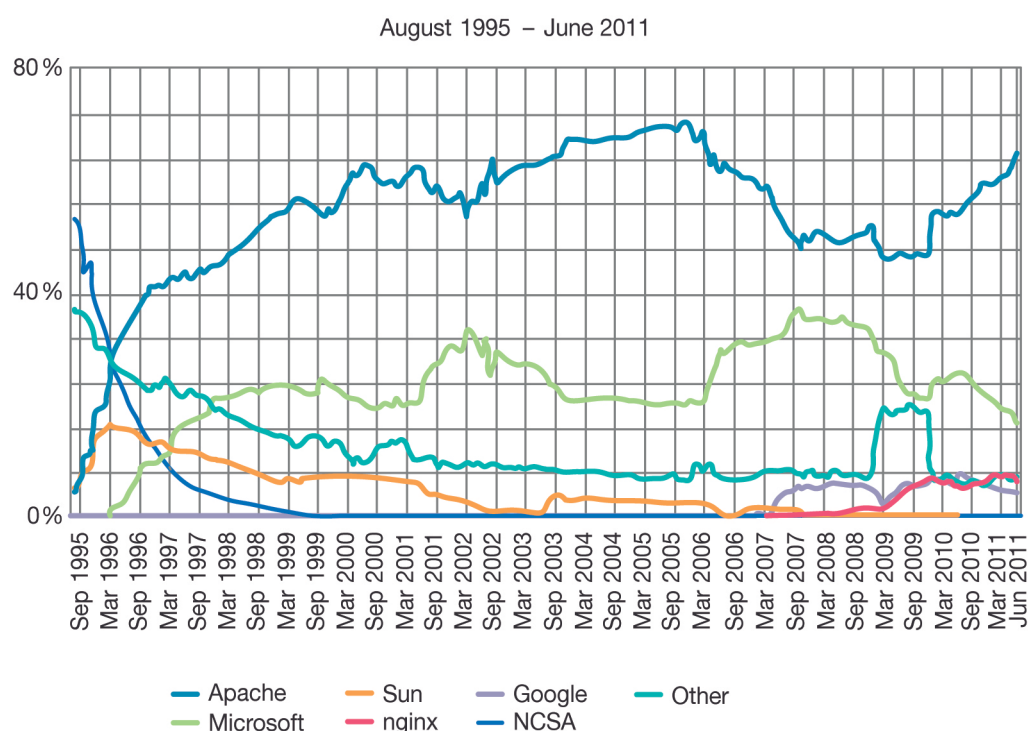


Figura 1.2. Evolución histórica de soluciones http.

Si nos preocupamos de los datos relativos a la actualidad, para convertir ese liderazgo en un tanto por ciento de mercado, pudiendo sacar comparaciones rápidas con sus principales competidores, en el siguiente gráfico podremos observarlo de una manera contundente:

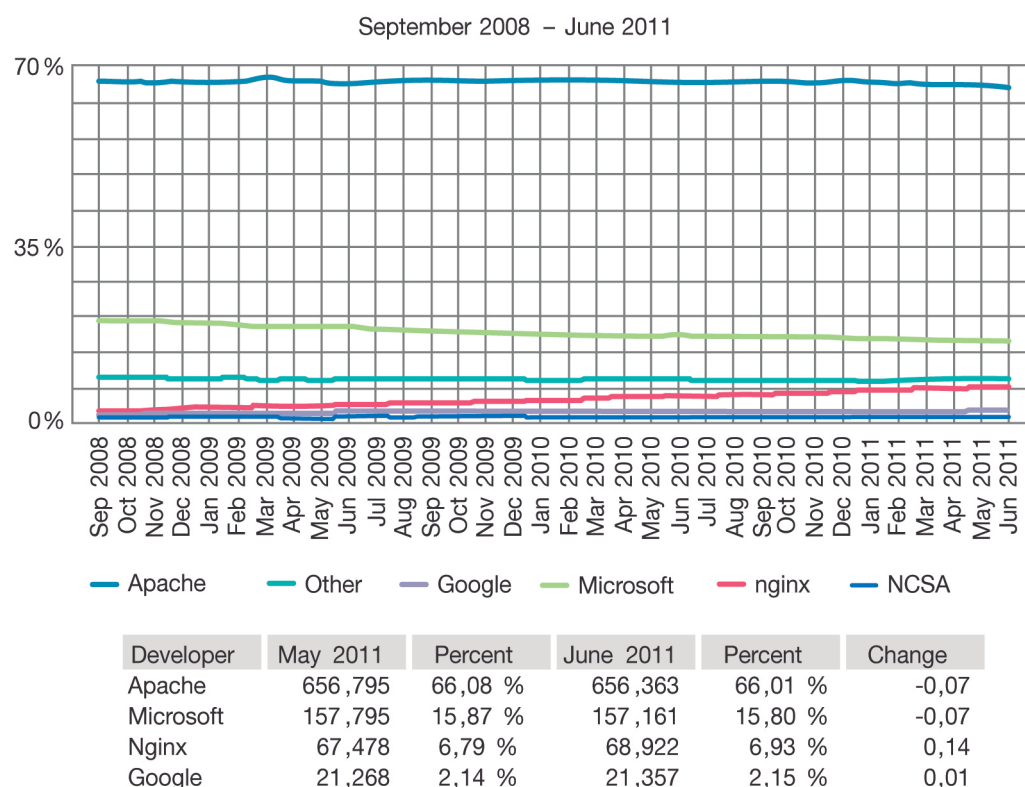


Figura 1.3. Comparativa actual en %.

Como podemos observar, lidera el mercado con un 66% de la totalidad, frente a un 15,80% de su rival más directo, Microsoft.

Las estadísticas son asombrosas, y obviamente tienen que ir acompañadas de una explicación de sus principales características diferenciadoras del resto de soluciones. Podríamos citar a modo general las siguientes:

- **Apache es un servidor altamente configurable de diseño modular.** Será sencillo poder ampliar las capacidades del servidor Web Apache. Con un poco de experiencia en lenguaje como C o Perl, podremos escribir nuestro propio módulo para poder realizar una función determinada. Como consecuencia, existen una gran cantidad de módulos diseñados para Apache y que están disponibles para su utilización.
- **Apache es una tecnología gratuita de código fuente abierto (*open source*).** El hecho realmente importante no es que sea gratuita, que también, sino sobre todo el hecho de que sea de código abierto, ya que vamos a poder implementar soluciones desarrolladas y mejoradas por una gran comunidad.
- **Apache puede trabajar con una gran cantidad de lenguajes de programación como PHP, Perl u otros lenguajes de script.**

- **Apache fue diseñado especialmente para sistemas Unix.** Linux, al ser un sistema basado en Unix, se ha aprovechado especialmente del auge que ha supuesto la implementación de soluciones Apache, y por lo tanto, forman el binomio habitual en la instalación y configuración de servidores de servicios de hosting. Otros sistemas de Unix como FreeBSD, Solaris o Mac OS, también pueden implementar Apache, formando también parte de la ampliación de usuarios de Apache.
- **Apache también trabaja bajo Windows.** A partir de la versión 2.0 de Apache, su integración en sistemas Windows mejoró ostensiblemente, y hoy en día, suele ser implementado en multitud de sistemas Windows, en detrimento de la propia solución de Microsoft, *Microsoft Internet Information Server*.

Estas características, diríamos que son principalmente ventajas que oferta Apache a un nivel general.

Pero Apache tiene otras características, desde el punto de vista técnico, que también lo diferencian de sus competidores. Entre ellas podemos destacar un elaborado índice de directorio, negociación de contenidos, un directorio de alias, informes de errores HTTP altamente configurable, ejecución SetUID de programas CGI, gestión de recursos para procesos hijos, reescritura de URLs, integración de imágenes del lado del servidor, comprobación de la ortografía de las URL, así como manuales on line.

Otras características técnicas especialmente importantes serán las siguientes:

- **Soporte del último protocolo HTTP 1.2.** Apache fue una de las primeras soluciones de servidor de páginas web en integrar dicho protocolo, por lo que no solo es compatible con esta versión, sino con las anteriores 1.1 y 1.0.
- **Configuración basada en el archivo *httpd.conf*.** El servidor Apache no trae una configuración gráfica a la que podamos acceder para realizar cambios en sus configuraciones originales. Tenemos que hacer el uso del archivo *httpd.conf*, mediante el uso de cualquier editor de texto. No obstante, es flexible en cuanto a la posibilidad de repartir la configuración de su host virtual en otros archivos para no sobrecargarlo con toda la gestión de las múltiples configuraciones que podremos realizar.
- **Soporte para CGI (*Common Gateway Interface*).** Mediante la inclusión de los módulos "*mod_cgi*" y "*mod_cgid*", Apache será capaz de soportar CGI, aportando también otras características extendidas como la personalización de las variables de entorno y soporte de reparación de errores o debugging, más complicadas de implementar en otras soluciones de servidores web.
- **Soporte para Perl.** El lenguaje Perl es el preferido para programar CGI. A través del módulo "*mod_perl*", podremos descargar un script CGI hecho en Perl a la memoria, y reutilizarlo tantas veces como necesitemos.
- **Soporte para host virtuales.** Apache fue uno de los primeros servidores web en poder soportar tanto host que estuvieran basados en direcciones IP como en host virtuales.

- **Soporte de autenticación HTTP.** Apache soporta autenticación básica basada en la web. De esta forma, será capaz de implementar la autenticación básica a través de archivos estándar de contraseña como los DBM, llamadas a SQL o llamadas a programas externos de autenticación.
- **Soporte para PHP.** Mediante la inclusión del módulo “*mod_php*”, podremos hacer uso de este lenguaje de programación, que vamos a conocer en profundidad en este libro.
- **Estado del servidor y adaptación de registros.** Con Apache tendremos una gran flexibilidad en las tareas de registro y monitorización del estado del servidor, que además podremos realizar mediante un navegador web. Podremos personalizar estos registros según nuestras necesidades.
- **Soporte para SSI (*Server Side Includes*).** Apache ofrece un conjunto de SS que añadirán una gran cantidad de flexibilidad para los desarrolladores web.
- **Soporte para SSL (*Secured Socket Layer*).** Podremos crear de forma sencilla un sitio web SSL mediante la implementación de **OpenSSL** y el módulo “*mod_ssl*” de Apache.

1.1.2. MySQL

MySQL es un sistema gestor de bases de datos relacionales (RDBMS, *Rapid Database Management System*) rápido y sólido. Con la utilización de bases de datos, podremos realizar operaciones de búsqueda, ordenación y consulta de datos de una forma óptima. Instalando un servidor MySQL, podremos controlar el acceso a estos datos para, de esta forma, poder asegurar el uso simultáneo de varios usuarios, para que puedan acceder a sus datos, garantizando que sólo podrán acceder a aquellos quienes tengan autorización para realizar dichas tareas.

MySQL es un servidor multiusuario y de subprocesamiento múltiple que utiliza el lenguaje SQL (*Structured Query Language*, Lenguaje de consulta estructurado), que es el lenguaje estándar para la consulta de bases de datos más utilizado en todo el mundo.

Una base de datos relacional va a guardar sus datos en tablas separadas en lugar de poner todos sus datos en un gran almacén. Con esto conseguimos aumentar la velocidad y flexibilidad. La parte SQL de MySQL se refiere a “*Structured Query Language*”.

SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. El estándar SQL ha evolucionado desde 1986 y existen varias versiones. En este manual, “SQL-92” se refiere al estándar del 1992, “SQL:1999” se refiere a la versión del 1999, y “SQL:2003” se refiere a la versión actual del estándar. Usamos la frase “el estándar SQL” para referirnos a la versión actual de SQL.

MySQL se distribuye bajo un sistema de licencias dual. Podremos utilizarlo con una licencia de código abierto (GPL), que será gratuita mientras cumplamos sus condiciones. Pero si deseamos distribuir una aplicación que no sea GPL y que incluya MySQL, tendremos que comprar una licencia adicional.

MySQL™ es propiedad y patrocinio de MySQL AB, compañía fundada por David Axmark, Allan Larsson y Michael Widenius, con base de operaciones en Suecia, la cual posee los derechos de autor de casi todo el código que lo integra. MySQL AB desarrolla y mantiene el sistema vendiendo servicios de soporte y otros valores agregados, así como el licenciamiento propietario para los desarrollos de equipamiento lógico que requieren mantener cerrado su código.



Figura 1.4. Logo MySQL.

Entre los competidores oficiales de MySQL se encuentran otros gestores de bases de datos como, por ejemplo, PostgreSQL, Microsoft SQL Server u Oracle entre otros.

En cuanto a las razones por las que nos descantamos por MySQL, podemos citar las siguientes:

- **Escalabilidad y flexibilidad.** MySQL puede trabajar con bases de datos sencillas de 1Mb de tamaño hasta otras con varios Tb de información. Como ya hemos comentado, su flexibilidad nos permitirá poder instalarlo en sistemas Windows, Linux o Mac OS. Al ser open source, podremos personalizar características de MySQL en el servidor.
- **Alto rendimiento.** MySQL puede trabajar en cualquier sistema, desde un sistema de procesamiento transaccional de alta velocidad, hasta una página web que tenga millones de consultas diarias. Esto lo puede realizar gracias a la implementación de herramientas de carga de alta velocidad, índices full-text y otros mecanismos.
- **Alta disponibilidad.** MySQL tiene a la disposición de sus clientes herramientas de alta disponibilidad, tales como replicación a servidores de cluster especiales.
- **Soporte transaccional.** MySQL tiene uno de los motores de bases de datos transaccionales mejores del mercado. Esto se traduce en la inclusión de un sistema ACID (atómica, consistente, aislada, duradera), realización de transacciones distribuidas, soporte de transacciones múltiples donde un lector no podrá bloquear a un escritor, o al revés. Ofrece integridad completa de los datos a través de integridad referencial o niveles de aislamiento de transacciones, por ejemplo.
- **Protección de datos.** Suministra potentes mecanismos para autorizar el acceso al servidor solo a usuarios autenticados, soporte SSH y SSL para conexiones seguras, funciones de cifrado y descifrado, utilidades de backup y recuperación o una fuerte estructura de privilegios para usuarios.

- **Posibilidad de desarrollo con diferentes lenguajes de programación.** No importará si un programador va a utilizar PHP, Perl, Java, Visual Basic o .NET, por ejemplo, ya que MySQL ofrece procedimientos almacenados, triggers, vistas, funciones, cursores, SQL estándar y otras librerías que facilitarán la vida del programador.
- **Facilidad de instalación.** El proceso de instalación de MySQL es extremadamente sencillo. Desde su descarga hasta su primer funcionamiento no deberían de pasar más de 15 minutos.

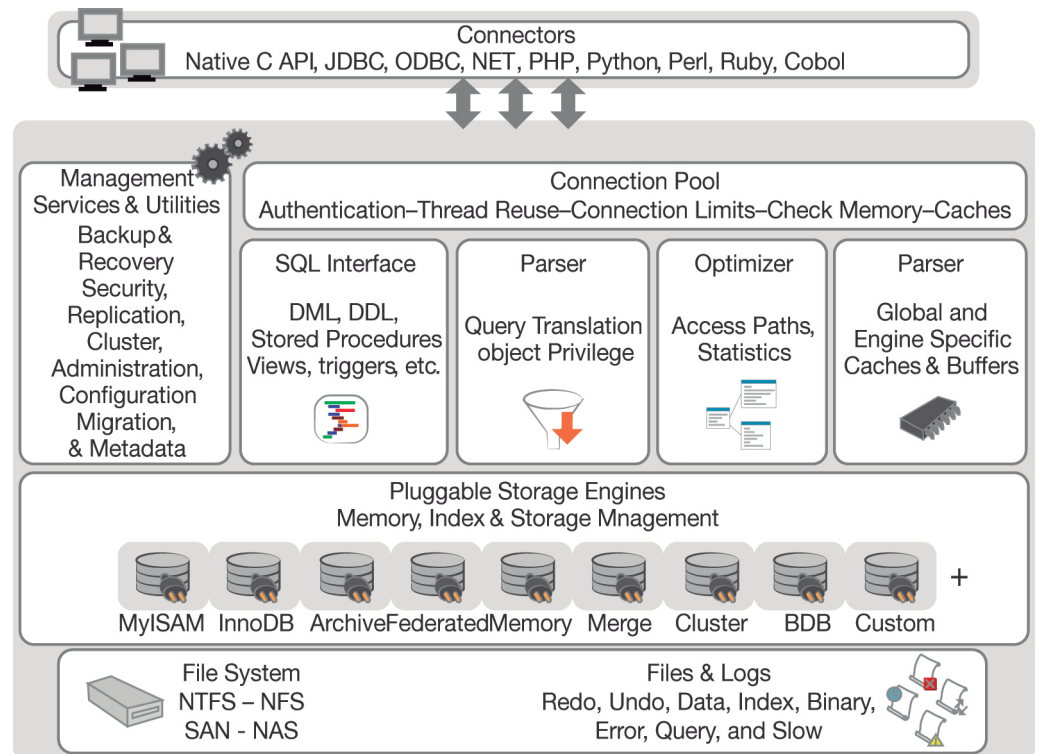


Figura 1.5. Esquema de funcionamiento de MySQL.

1.1.3. PHP

PHP (*PHP Hypertext Preprocessor*) es un lenguaje de secuencia de comandos de servidor que fue desarrollado de forma específica con el objetivo de construir sitios web. Utilizaremos el lenguaje PHP en el diseño de una web, programando un código fuente que será ejecutado cada vez que un usuario visite la página web. El código que hayamos generado se ejecutará siempre en el lado del servidor, y será transformado a código HTML, así como a otro tipo de contenido.

PHP es una de las soluciones adoptadas por la mayoría de programadores y empresas para llevar a cabo sus proyectos web en Internet. Tiene todas las ventajas para que así sea, ya que se trata de un software de código abierto (*Open Source*), y por lo tanto, con la ventaja que supone también que exista una gran comunidad dando soporte a la mejora de la tecnología y compartiendo sus proyectos para que puedan ser mejorados por el resto.

Está desarrollado para que pueda trabajar con las plataformas más utilizadas (Windows, Linux, Macintosh, Solaris, etc.) e incluye soporte para un gran número de gestores de bases de datos y otras tecnologías como Java o XML.

Otro de sus puntos fuertes es su fácil aprendizaje. Como podremos comprobar a lo largo de este libro, crear páginas PHP no resultará nada difícil; y si además, ya poseemos algún conocimiento sobre todos lenguajes de programación, aún será más fácil asimilarlo y llevar a la práctica esta nueva forma de crear páginas dinámicas.

En el diseño de páginas web se pueden utilizar lenguajes que se ejecutan en cliente o en el servidor. Los lenguajes de cliente (HTML, Javascript o VBscript...) se caracterizan porque su código es ejecutado en los navegadores de los usuarios que visitan estas páginas web, de tal forma que será el navegador el encargado de interpretar el código y mostrarlo en pantalla. Esto nos supone el principal inconveniente, que a la hora de desarrollar el código, tendremos que prestar especialmente atención a los diferentes navegadores que existan en el mercado, para que pueda ser compatible con todos.

Por el contrario, los lenguajes como PHP que se ejecutan en el lado del servidor, será éste quien se encarga de interpretar el código y llevar hasta el usuario el resultado obtenido. El proceso será el siguiente: el navegador realiza una llamada al servidor web, que pasará la petición a través de la interfaz PHP del servidor web. Esta interfaz realizará una llamada al intérprete PHP, que accederá a la zona del disco donde estén almacenados los scripts PHP, de tal forma que el servidor recuperará el código PHP y lo enviará a un compilador. Este compilador convertirá el código en HTML, que podrá ser interpretado por el navegador del usuario.

Para hacer un resumen de sus principales características, vamos a dividirlo en dos partes. En primer lugar, haremos un resumen de sus principales características más generales, y a continuación, haremos hincapié en características más técnicas.

En cuanto a las características generales, el resumen sería el siguiente:

- **Lenguaje de programación multiplataforma.** Podremos trabajar con PHP en entornos diferentes como Linux, Windows o Macintosh.
- **Accesibilidad a un gran número de gestores de bases de datos.** Con PHP podremos utilizar una gran diversidad de SGBD (Sistemas Gestores de Bases de Datos). La lista sería inacabable, pero podríamos citar de entre las más conocidas: Adabas D, dbm, dBase, filePro, Hyperwave, Informix, InterBase, LDAP, Microsoft SQL Server, MsqI, MySQL, ODBC, Oracle, PostgreSQL o Solid, entre otras.
- **Gran cantidad de librerías.** PHP tiene definidas muchas funciones ya implementadas que nos permitirán realizar acciones como, por ejemplo, las siguientes:
 - Acceder a servidores IMAP.
 - Envío de correo mediante SMTP.
 - Acceso a servidores FTP.

- Generación de gráficos dinámica y documentos PDF.
- Análisis de documentos XML.
- **Gran variedad de aplicaciones para facilitar la programación con PHP.** Existen una gran cantidad de editores gratuitos diseñados para trabajar exclusivamente con PHP. También existen muchos frameworks (entornos de trabajo) y CMS (Gestores de Contenidos) basados en PHP como Joomla, Drupal, etc.
- **Simplicidad y facilidad de aprendizaje.** PHP dispone de una gran documentación, tanto en manuales como ejemplos, que podemos conseguir a través de Internet. Con este libro y los recursos comentados, el aprendizaje será más sencillo.

En cuanto a sus características más técnicas, tendremos que empezar comentando que está a punto de salir a luz la versión PHP 6.0, que nos aportará las siguientes características como principales novedades, si bien es cierto que algunas de ellas ya se desarrollaron desde la versión 5.3:

- **Soporte para Unicode.** PHP6 soportará Unicode en la mayoría de sus funciones principales. Esta característica tendrá un fuerte impacto, ya que permitirá una mayor capacidad de internacionalización al soportar un conjunto de caracteres mucho mayor mediante sus funciones nativas. Actualmente podemos activar esta característica bajo petición, con lo cual PHP consume más recursos al tener que guardar las clases, funciones y el resto de elementos necesarios.
- **PHP6 soporta *namespaces*.** Una manera de evitar colisiones entre los nombres que asignemos a las diferentes clases sin tener que estar elaborando notaciones específicas para evitar este problema. De esta manera al crear nuestro propio espacio de nombres no tendremos que preocuparnos por si PHP tiene una clase con el mismo nombre o si una librería externa podría fallar debido a una colisión en los nombres de clase. No es obligatorio el uso de namespaces, lo que permite que el código de la versión 5 sea compatible con esta nueva característica. Los namespaces aparecen ya soportados en la versión PHP 5.3
- **SOAP.** SOAP es uno de los protocolos empleados por los servicios web soportado en otros lenguajes como JAVA. SOAP, y que permite la interoperatividad entre diferentes plataformas. PHP6 tendrá una extensión SOAP que nos permitirá una manera más sencilla de implementar tanto clientes como servidores SOAP.
- **XML.** Desde PHP 5.1, *XMLReader* y *XMLWriter* forman parte del núcleo de PHP, lo que facilita el trabajo con documentos XML en las aplicaciones PHP. Con la llegada de la nueva versión, el funcionamiento mejorará, ya que *XMLReader* y *XMLWriter* estarán activadas por defecto.

Para finalizar esta parte de introducción al lenguaje PHP, vamos a comentar cuáles son las principales utilidades en las que podemos emplear los scripts que creemos con PHP:

- **Scripts para ejecutar en un servidor.** Esta sería la principal finalidad de todo desarrollador en PHP: generar un código que almacenar en un servidor web como Apache, que finalmente se convierta en una aplicación web que pueda consultar un usuario a través de su navegador.

- **Scripts en línea de comandos.** También podremos crear scripts en PHP cuya finalidad no sea la de ser ejecutados en un servidor web o un navegador. Podemos tener instalado simplemente un intérprete PHP en nuestro sistema, y por ejemplo, destinar nuestro código a la creación de scripts que sean ejecutados como tareas cron en sistemas Unix o en el planificador de tareas de Windows.
- **Diseño de aplicaciones gráficas.** Seguramente PHP no será el lenguaje más optimizado para el desarrollo de aplicaciones gráficas, pero si somos capaces de conocer a fondo este lenguaje, podremos hacer uso de librerías como PHP-GTK, para desarrollar este tipo de soluciones.

1.2. Instalación paquete AMP

Afortunadamente, hoy en día podemos encontrar paquetes de software que en menos de 10 minutos se encargarán de realizar automáticamente la instalación conjunta de Apache+PHP+MySQL, con una breve participación por nuestra parte.

A este tipo de paquetes que integran las tres aplicaciones, se les denomina AMP (Apache MySQL PHP), y en función de las plataformas en las que vayan a ser instalados, se denominarán de la siguiente manera:

- **WAMP.** Para plataformas Windows.
- **LAMP.** Para plataformas Linux.
- **MAMP.** Para plataformas MacOX.

En los siguientes capítulos vamos a realizar el proceso de instalación de herramientas AMP para cada una de las plataformas comentadas.

1.2.1. Instalación WAMP

Para plataformas Windows, existen numerosas aplicaciones que ofertan las mismas soluciones para integración de AMP. Podríamos destacar las siguientes:

- Appserv 2.6.0.
- EasyPHP 3.5.3.3.
- Xampp 1.7.4.
- WAMP Server 2.0.

El proceso de instalación será muy parecido en todas. Nosotros para realizar esta demostración hemos escogido la aplicación **Xampp**, ya que la tendremos disponible para las tres plataformas.

Si accedemos a su página web, podemos ver lo que nos ofrece la instalación de su solución para la versión Xampp 1.7.4:

- Apache 2.2.17.
- MySQL 5.5.8.
- PHP 5.3.5.
- phpMyAdmin 3.3.9. Aplicación web para gestión de bases de datos MySQL que veremos en la unidad 2.
- FileZilla FTP Server 0.9.37. Servidor FTP para subir nuestros archivos a un servidor de hosting.

Para comenzar con la instalación primero tendremos que descargar la última versión desde el siguiente link:

<http://www.apachefriends.org/download.php?xampp-win32-1.7.4-VC6-installer.exe>

Una vez hayamos descargado el programa y lo ejecutemos, tendríamos que ver una pantalla como la siguiente:

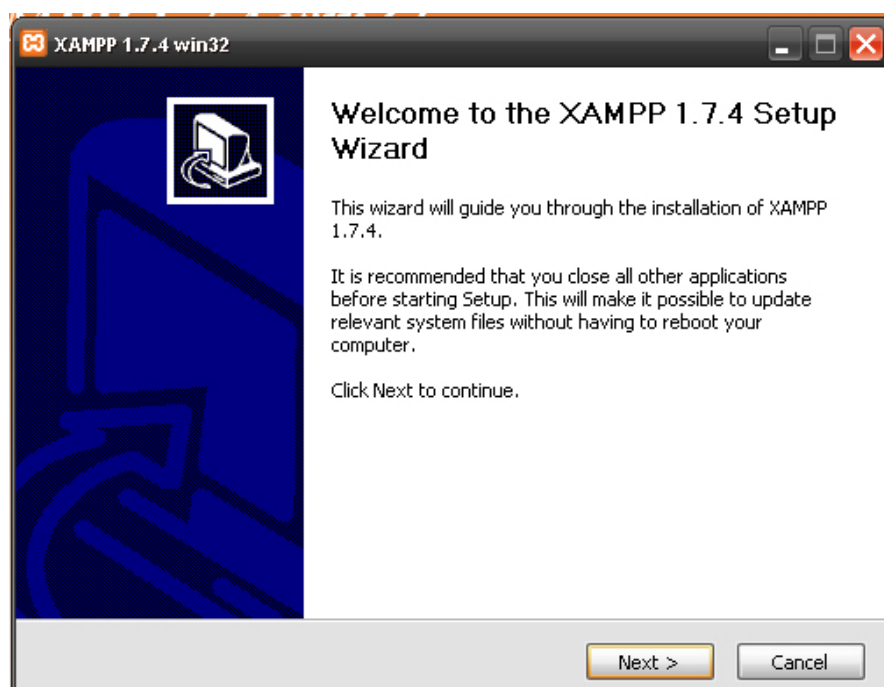


Figura 1.6. Comienzo de instalación de XAMPP 1.7.4.

En el proceso de instalación, como usuarios del mismo solo tendremos que intervenir en dos ocasiones. En la primera, tendremos que elegir el directorio donde queremos que se instale el conjunto de aplicaciones.

Nosotros dejamos la ruta por defecto (C:\XAMPP), pero a través del botón “**Browse**”, podríamos escoger la ruta que nos pareciera oportuna:

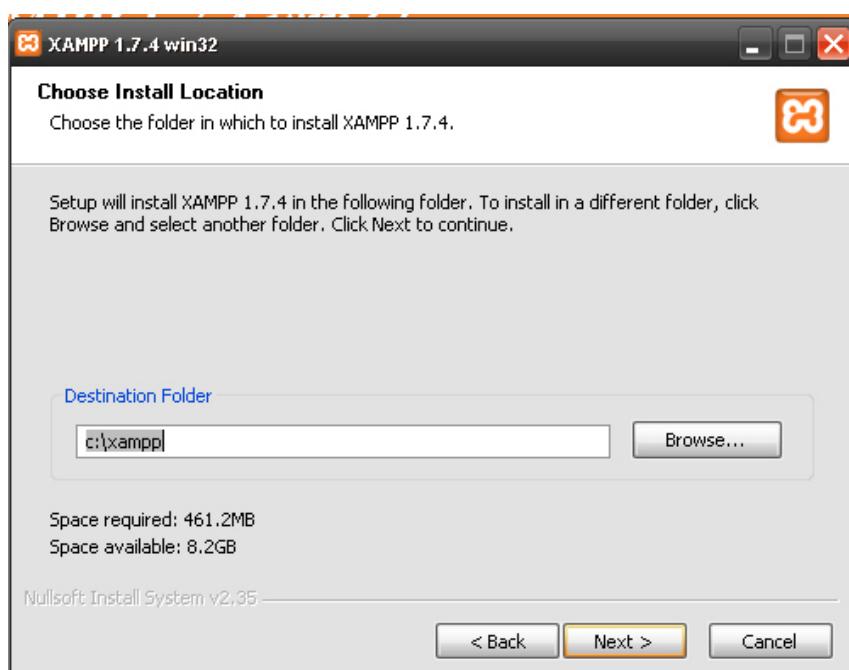


Figura 1.7. Directorio de instalación.

En la segunda intervención, podremos decidir si queremos crear un icono de acceso directo en el escritorio y que nos cree una entrada en el menú de Inicio junto con el resto de programas que tengamos instalados.

Además, podremos activar las opciones para que tanto Apache, como MySQL y el servidor FTP Filezilla, se instalen como servicio de Windows, lo cual nos garantizará que en cada arranque del equipo, se iniciarán los servicios necesarios para que se ejecuten las aplicaciones comentadas.

La activación de estos servicios será recomendable en aquellos casos en los que en la mayoría de las ocasiones, vayamos a utilizar estos programas. Si solo los utilizamos de vez en cuando, no se recomienda su activación, ya que ralentizaría el inicio del sistema operativo, y además, se estaría ejecutando servicios en memoria que no se están utilizando, sobrecargando la memoria y el procesador del equipo.

En nuestro caso, como vamos a hacer uso frecuente tanto de Apache como de MySQL, los activamos para que se inicien al arranque de Windows:

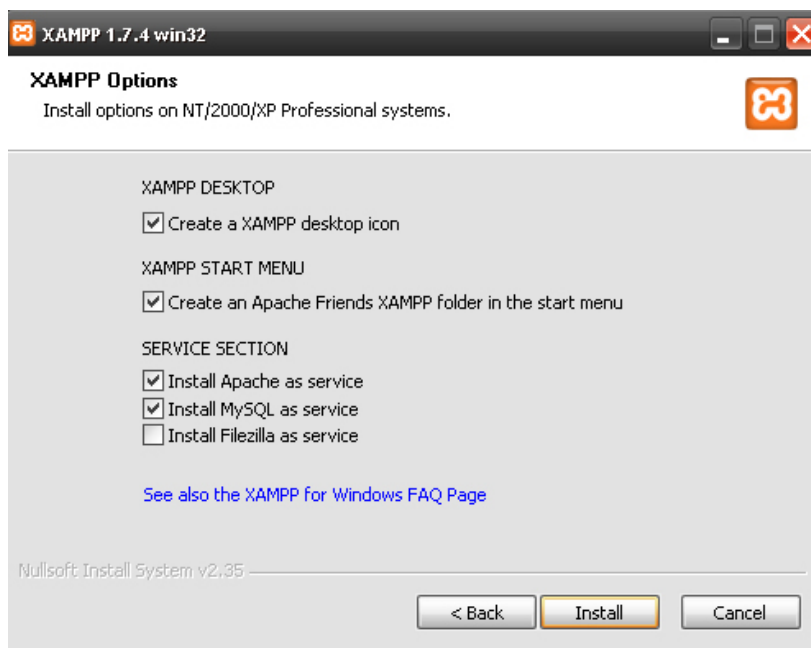


Figura 1.8. Opciones de instalación.

A continuación, comenzará el proceso extracción e instalación de los archivos necesarios para cada una de las aplicaciones, y al terminar, nos tendríamos que encontrar con una pantalla como la siguiente:

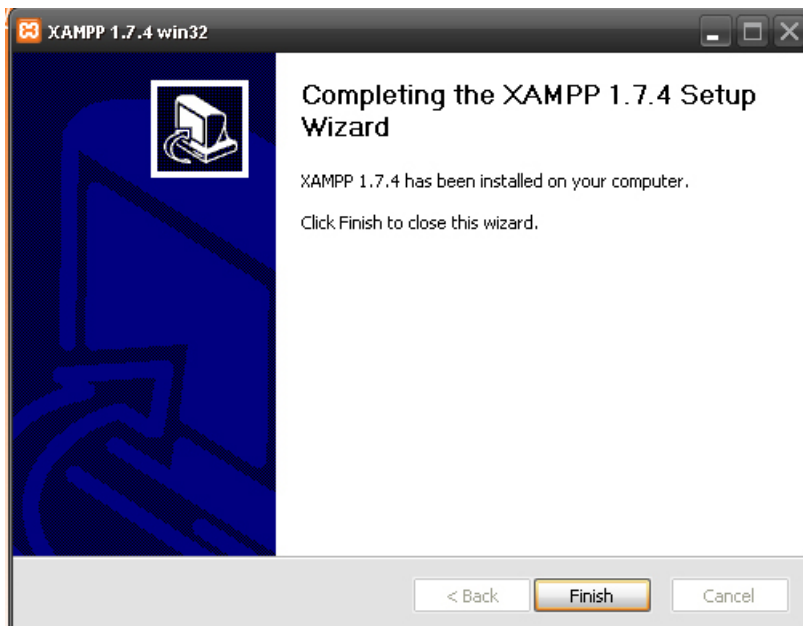


Figura 1.9. Fin de instalación.

Una vez terminada la instalación, a continuación, XAMPP lanzará los procesos de inicialización de los servicios de Apache y MySQL por primera vez, de tal forma, que una vez que hayan sido iniciados, nos preguntará si deseamos ejecutar el panel de control, a través del cual se administran todas las aplicaciones:

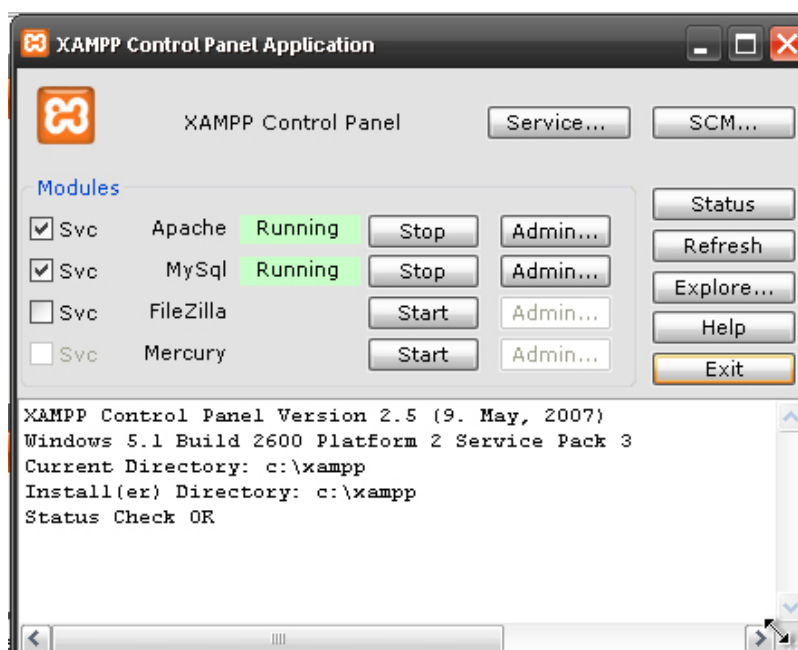


Figura 1.10. Panel de control de XAMPP.

Tal y como podemos ver en la imagen anterior, podremos iniciar o detener los servicios de Apache, MySQL y FileZilla con los botones **Stop** y **Start**.

Como podemos observar, tanto Apache como MySQL aparecen marcados por el checkbox en la columna denominada “Svc” que indica que están instalados como servicios. Si quisiéramos que esto dejara de ser así, valdría con desactivar el checkbox.

Para comprobar que todo funciona correctamente, hay que escribir en el navegador la dirección “<http://localhost>”. Al abrir la página por primera vez, XAMPP pedirá seleccionar el idioma:



Figura 1.11. Comprobación de correcto funcionamiento.

Una vez elegido el idioma, se mostrará la página de configuración de XAMPP:



Figura 1.12. Página de configuración de XAMPP.

Una última consideración antes de dar por terminado el proceso de instalación: para que esta página web por defecto que nos muestra XAMPP en la anterior imagen pueda ser cargada, necesita estar almacenada en la ruta que tiene configurada el servidor Apache, y que será en la que siempre vaya a buscar los proyectos web almacenemos en su servidor.

Esta ruta es C:\XAMPP\HTDOCS:

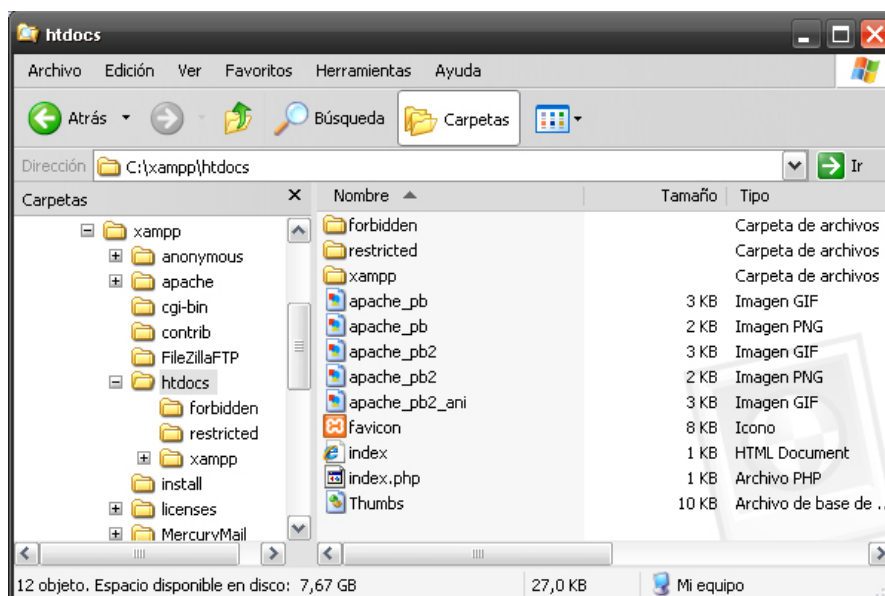


Figura 1.13. Carpeta de proyectos.

Como podemos observar, una de las carpetas que tenemos almacenadas en C:\XAMPP\HTDOCS, es la carpeta XAMPP, que va a almacenar en su interior los ficheros necesarios para mostrarnos la pantalla de bienvenida que hemos podido ver en la figura 1.12.

Por lo tanto, a partir de ahora, todos los proyectos que vayamos a crear con nuestros scripts en PHP, tendrán que ser almacenados de forma organizada en esta carpeta para que puedan ser localizados por el servidor web Apache.

1.2.2. Instalación LAMP

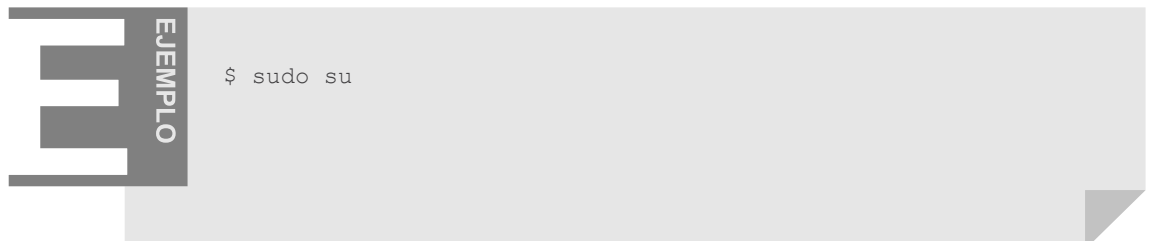
El proceso de instalación para una distribución Linux es algo más elaborado. En primer lugar, nos descargaremos el paquete desde la siguiente dirección:

<http://www.apachefriends.org/download.php?xampp-linux-1.7.4.tar.gz>

La instalación de XAMPP nos va instalar las herramientas principales que necesitamos (Apache, MySQL y PHP), pero también contiene otras herramientas: Pearl, phpMyAdmin, openssl y proFTPD.

A continuación, tendremos que acudir a la consola de comandos y realizar los siguientes pasos:

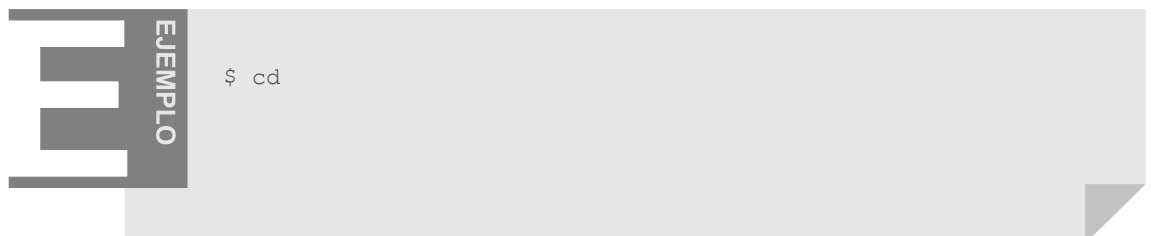
1. En primer lugar, nos cambiamos al superusuario para tener permisos de instalación mediante el siguiente comando:



EJEMPLO

```
$ sudo su
```

2. A continuación, tendremos que desplazarnos hasta el directorio donde guardamos el paquete XAMPP. En nuestro caso, el directorio personal. Con la siguiente instrucción, estemos donde estemos, nos situará en nuestro directorio personal:



EJEMPLO

```
$ cd
```

3. La siguiente tarea será desempaquetar el archivo bajado y colocar su contenido en el directorio **/opt**:

EJEMPLO

```
$ tar zxvf xampp-linux-1.7.4.tar.gz -C /opt
```

Con el anterior comando, se habrá instalado XAMPP en el directorio `/opt/lampp/`.

Como la versión XAMPP que hemos descargado es para Linux, podremos ver que recibe el nombre de LAMPP.

4. A continuación, iniciaremos XAMPP:

RECUERDA

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

Tendríamos que ver algo como lo siguiente:

RECUERDA

```
Starting XAMPP 1.7.4...
LAMPP: Starting Apache...
LAMPP: Starting MySQL...
LAMPP started.
```

5. Para verificar que se hayan iniciado los servidores Apache y MySQL, abrimos nuestro navegador de internet y en la barra de direcciones volvemos a escribir `http://localhost`:

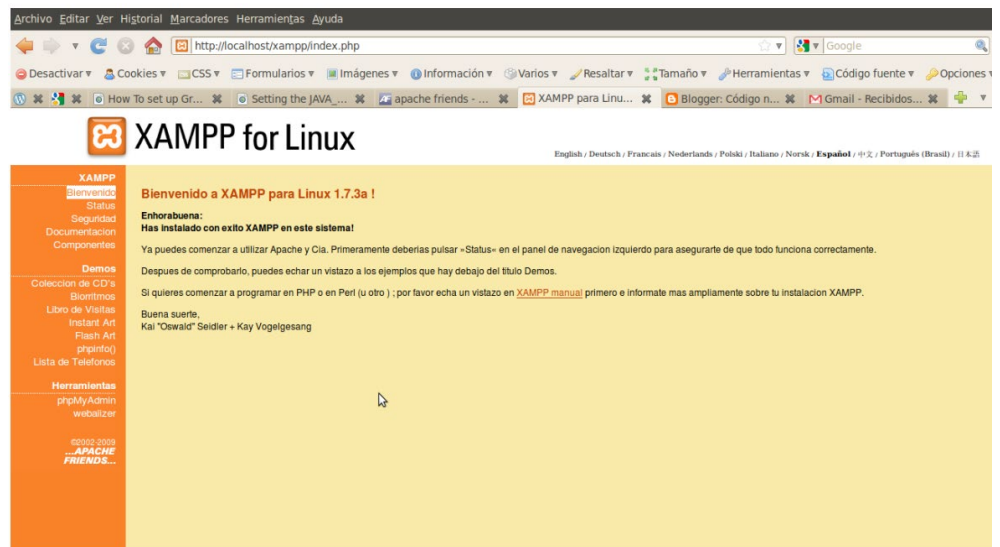
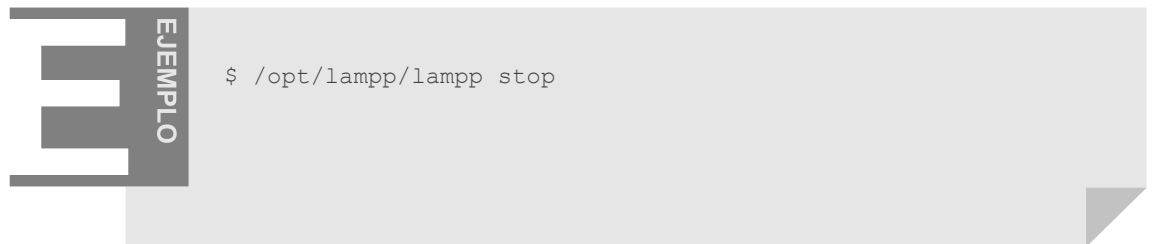
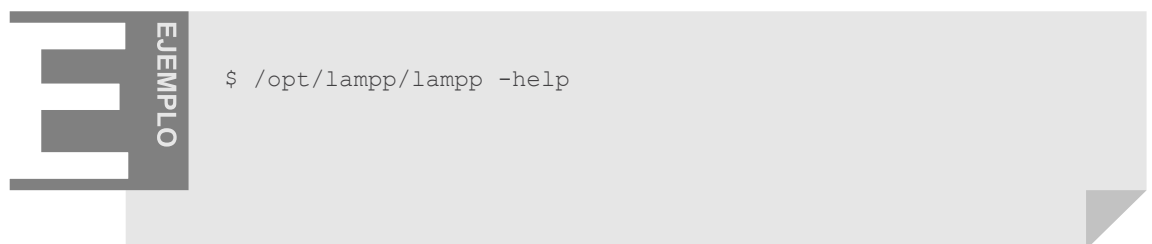


Figura 1.14. XAMPP funcionando.

Hasta aquí, el proceso de instalación. Para detener los servicios de XAMPP, tendremos que ejecutar el siguiente comando en el Terminal:



Para obtener más ayuda sobre iniciar, detener, reiniciar, etc, los servicios de XAMPP, tendremos que ejecutar el siguiente comando:



La ruta donde tendremos que almacenar nuestros proyectos web será `/opt/lampp/htdocs/`:

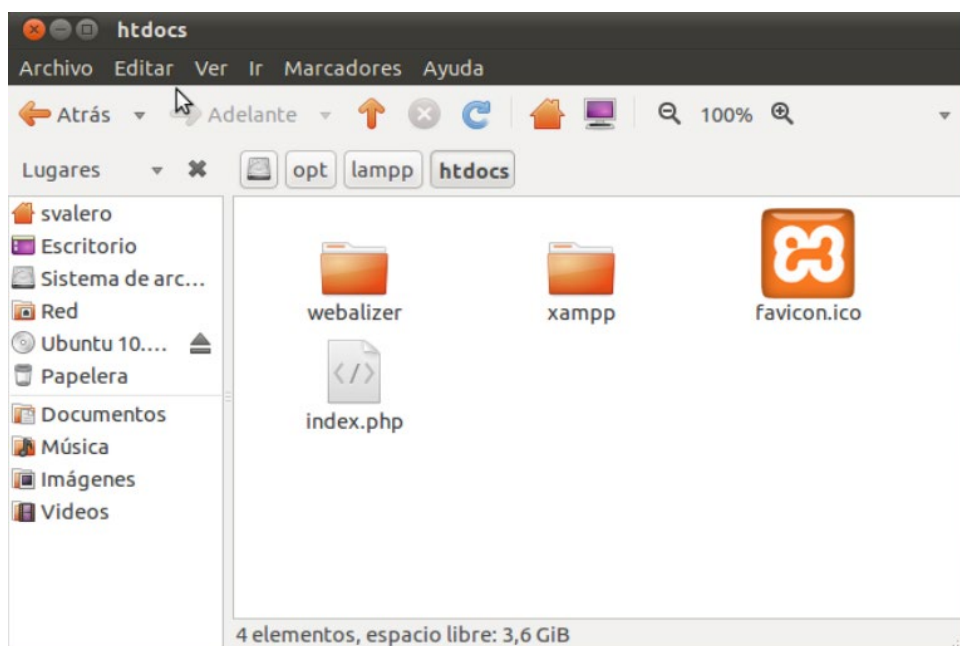


Figura 1.15. Ruta para archivos PHP.

Por último, una serie de recomendaciones a tener en cuenta relativas a la seguridad, o mejor dicho, a cuestiones que tienen que ver con la seguridad, pero que en este proceso de instalación son muy débiles:

- La contraseña del administrador de MySQL (**root**), no tiene password.
- El demonio MySQL es accesible vía red.
- ProFTPD, que es la aplicación FTP, utiliza el password “**lampp**” para el usuario “**nobody**”.
- PhpMyAdmin también es accesible por red.
- MySQL y Apache se están ejecutando con el mismo usuario (**nobody**).

1.2.3. Instalación MAMP

El proceso de instalación de XAMPP en equipo Mac OS es tan sencillo como en un equipo Windows.

Para empezar deberemos descargarnos el binario universal del programa del siguiente enlace:

<http://www.apachefriends.org/download.php?xampp-macosx-1.7.3.dmg>

Una vez se haya descargado y montado de forma automática la imagen (si no nos apareciera un icono similar al de un USB con el nombre de “**XAMPP for Mac OS X 1.7.3**”, tendremos que desplazarnos hasta la carpeta “descargas” y ejecutar el archivo que hemos descargado), nos tendría que aparecer la siguiente imagen:



Figura 1.16. Imagen DMG descargada.

Como muchas de las aplicaciones para este sistema operativo, lo instalaremos arrastrando el archivo a la carpeta aplicaciones.

Una vez que se haya copiado, al abrir la carpeta, podemos ver que tiene varios alias. Nosotros ejecutaremos el icono llamado “XAMPP Control”.



Figura 1.17. Icono XAMPP.

Al ejecutarlo, nos aparecerán dos ventanas: una ayuda rápida y el panel de control.



Figura 1.18. Ventana de primeros pasos.

La pantalla “**Getting started**”, nos proporciona información básica sobre cómo acceder al “**localhost**” y que, por defecto, el usuario **root** de MySQL no tiene contraseña, como ya hemos explicado anteriormente.

A continuación, vamos a arrancar el servidor Apache mediante el botón “**Start**”, y haremos lo mismo con el servidor MySQL. Es posible que nos pida la contraseña de nuestro usuario de Mac OS. Esto es así porque ejecuta unos scripts para poder hacer funcionar todo el sistema de BBDD. Si todo ha ido bien, las luces pertenecientes a los grupos que nos interesan estarán en verde:

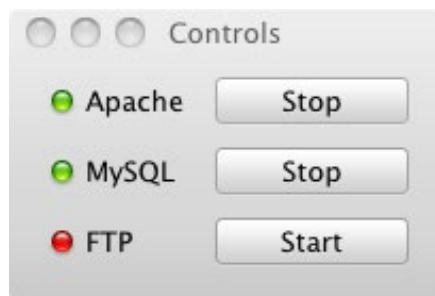


Figura 1.19. Panel de Control.

Ahora, ya estamos en condiciones de acudir a nuestro navegador, en introducir **http://localhost**. Como ya hemos explicado, en primer lugar nos preguntará por nuestro idioma:



Figura 1.20. XAMPP sobre Mac OS X.

RESUMEN

- Se han explicado las principales funciones de un servidor de páginas web.
- Hemos introducido el servidor de páginas web Apache, como referencia en soluciones http.
- Se han explicado las principales funciones de un gestor de bases de datos.
- Hemos introducir MySQL y se han explicado sus principales ventajas y características.
- Se ha explicado las diferencias entre un lenguaje de programación orientado a cliente y orientado a trabajo en servidor.
- Hemos introducido PHP como referencia en lenguajes de programación y hemos explicado sus principales características.
- Se ha explicado qué son las herramientas AMP, las ventajas que nos supone una instalación conjunta.
- Se han instalado y configurado soluciones AMP como XAMPP en plataformas Windows, Linux y Mac OS.

