

1 INTRODUCCIÓN	1
1.1 Lenguajes de programación web.	1
1.1.1 Tipos de páginas web. Estáticas y dinámicas.	1
1.1.2 Generación de páginas web.	2
1.1.3 Lenguajes de programación web.	3
1.2 ¿Qué es PHP?	5
1.2.1 Un poco de historia.....	5
1.2.2 Características de PHP.....	5
1.2.3 ¿Por qué PHP?	6
1.3 ¿Qué es Apache?	7
1.3.1 Otros servidores web.	8
1.4 Instalación.	8
1.4.1 Instalación paso a paso sobre <i>Windows</i>	9

1 INTRODUCCIÓN

1.1 Lenguajes de programación web.

1.1.1 Tipos de páginas web. Estáticas y dinámicas.

Las primeras páginas que aparecieron en Internet eran muy sencillas y poco interactivas. En ellas sólo se permitía mostrar información, ya se tratara de texto o imágenes, y seguir enlaces a nuevas páginas.

Para compensar esta falta de interactividad aparecieron los lenguajes de programación para el cliente. Es decir, para el navegador de Internet. Estos lenguajes aportaban una interactividad limitada. Dentro de esta categoría podemos destacar el lenguaje *Javascript*. Un par de ejemplos típicos de programación mediante *Javascript* podrían ser un menú jerárquico desplegable o una calculadora.

Pero hay multitud de funcionalidades que están fuera del alcance de los lenguajes de programación en el cliente. La interactividad que permiten se reduce prácticamente a mejorar la apariencia de la información en la página web, a añadir efectos visuales que aunque puedan ser curiosos no aportan un valor añadido, y a realizar operaciones matemáticas de mayor o menor dificultad con datos suministrados por el usuario final.

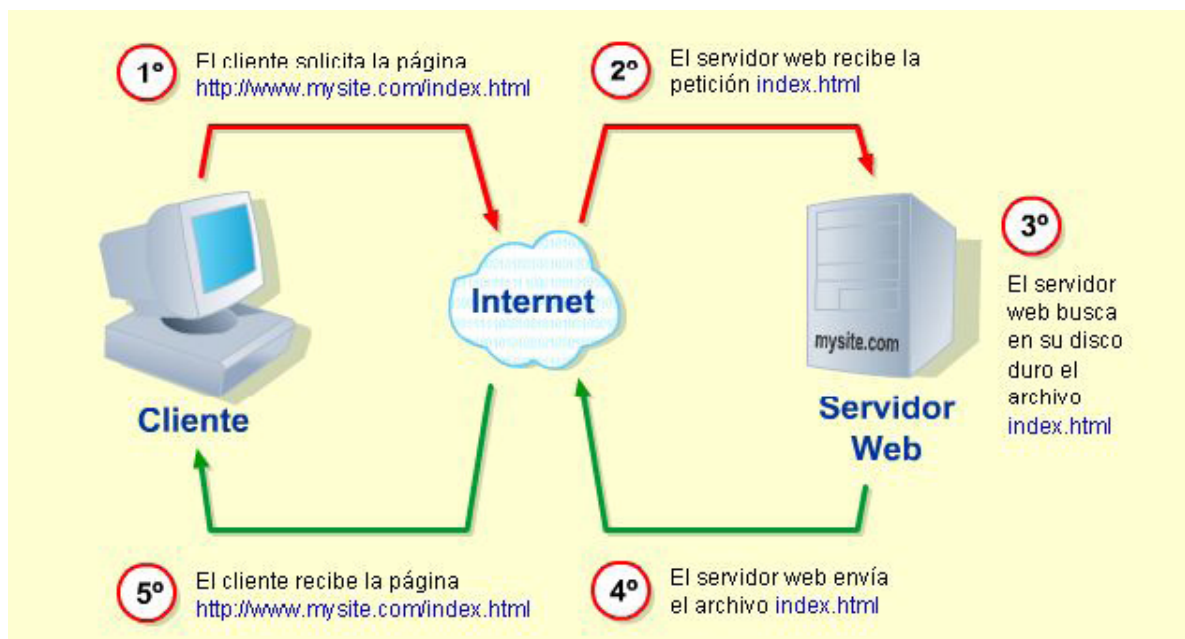
Con la llegada de los lenguajes de servidor aparecieron las primeras páginas dinámicas, cuyos principales avances consiste en la extracción de información de una base de datos, y el almacenamiento de información entre acciones. Hoy en día, la mayor parte de *Internet* está formada por este último tipo de páginas.

El buscador de *Internet* que usamos cada vez que queremos encontrar páginas de una temática y las tiendas virtuales que podemos encontrar en *Internet* son páginas dinámicas.

1.1.2 Generación de páginas web.

Para poder comprender la diferencia entre una página estática y una página dinámica, hay que tener presente el proceso que sigue una página web desde que la solicitamos, mediante un clic en un enlace del navegador, hasta que la recibimos y se muestra por pantalla.

1.1.2.1 Página estática.

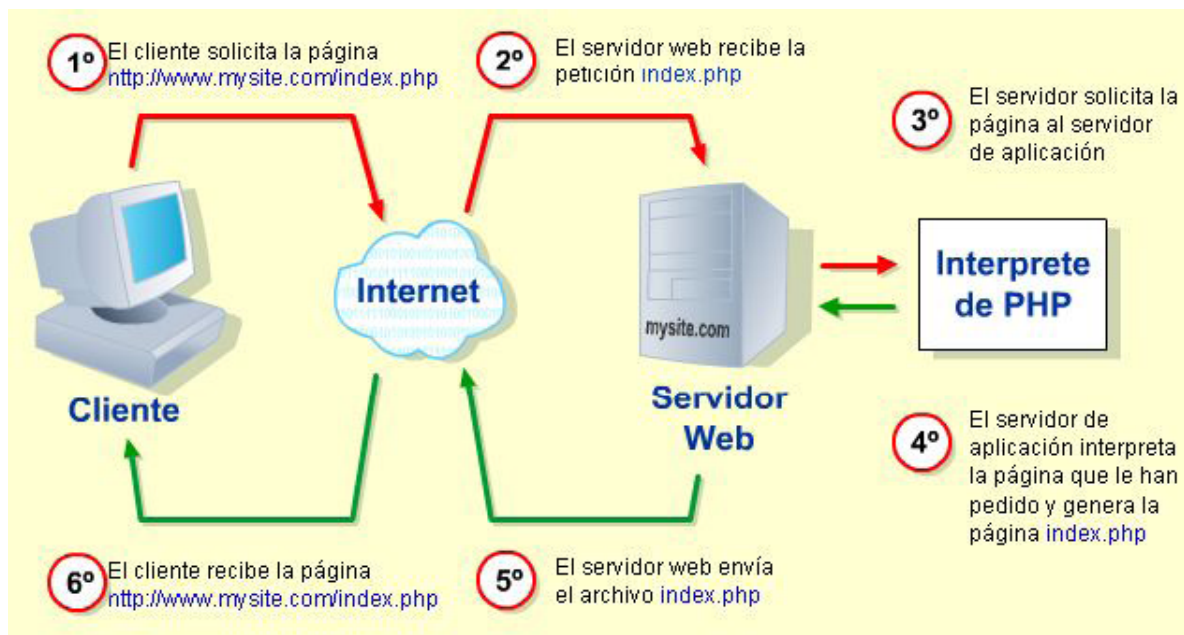


Elementos del proceso:

- a) Navegador: aplicación que permite al usuario recibir y visualizar las páginas web.
- b) Cliente: en el ámbito de Internet, es el ordenador que está conectado a la red. Su función se limita a realizar las peticiones de páginas web y mostrar los resultados en el navegador.
- c) Internet: es el medio físico por el que se transmiten las peticiones web, y sus respuestas.
- d) Servidor web¹ (hardware): ordenador conectado a Internet que aloja páginas web.
- e) Servidor web (software): programa que transfiere páginas web a los clientes mediante el protocolo *HTTP*.

¹ El término servidor web se usa indistintamente para denominar tanto al ordenador conectado a Internet que aloja páginas web como a la aplicación software que gestiona la transferencia de páginas web.

1.1.2.2 Página dinámica.



En este caso disponemos de un nuevo elemento, el servidor de aplicación.

- f) Servidor de aplicación: es el programa que colabora con el servidor web para generar páginas dinámicas.

A diferencia de las páginas estáticas, en las que se guardaba en el disco duro el archivo *html* correspondiente, las páginas dinámicas no se envían tal cual están almacenadas en el disco duro. En su lugar hay un programa, que se ejecuta en el servidor de aplicaciones, encargado de interpretar y ejecutar determinados ficheros con el fin de crear el archivo que se transferirá al cliente.

1.1.3 Lenguajes de programación web.

Para generar páginas web se han usado a lo largo del tiempo un gran número de tecnologías. En los comienzos no había ningún lenguaje de programación específico, por lo que se utilizaban lenguajes de uso más general como *C*, *C++*, *Basic* o *Pascal*.

Para comunicar estos programas con el servidor web se hacía uso de la especificación *CGI*, siglas de *Common Gateway Interface*, o en castellano *Pasarela de Interfaz Común*. Hoy en día, aún hay aplicaciones que siguen este enfoque, y comúnmente se las conoce como *CGI*'s.

Pero la utilización de lenguajes de uso general hacía complicado el proceso de construir páginas web dinámicas. El código se escribía de forma casi artesanal, y era difícil de mantener y actualizar. Además, el código que funcionaba en un servidor web podía no funcionar en otro

o requerir de un gran esfuerzo para su instalación. Además, los *CGI* suelen ser poco eficientes.

Afortunadamente, hoy en día el panorama ha cambiado radicalmente. El éxito de *Internet* ha ido acompañado de la aparición de lenguajes de programación web que facilitan todo el proceso. Para especializarse en la creación de páginas web dinámicas, estos lenguajes intentan cumplir los siguientes objetivos:

- Ser de fácil uso, para permitir una curva de aprendizaje sencilla.
- Adaptarse eficientemente a la rápida evolución de *Internet*, ofreciendo de manera continua nuevas características y actualizaciones de seguridad.
- Tener una gran comunidad de usuarios que garanticen el éxito del producto.

Estos lenguajes han proliferado de tal modo, que en la actualidad hay una gran cantidad de tecnologías disponibles, con la intención de acaparar la mayor cuota de mercado posible.

De entre estos lenguajes podemos destacar tres tecnologías mediante las cuales se desarrollan la mayoría de los sitios web: *ASP*, *JSP* y *PHP*.

1.1.3.1 ASP: Active Server Pages.

Esta es una tecnología propietaria de *Microsoft*. Actualmente, ésta engloba en su estrategia *.NET*. Su principal ventaja es su sencillez, además de una muy buena integración con otros productos de *Microsoft* como su entorno de desarrollo *.NET*, que permite crear aplicaciones en un tiempo récord. Se integra muy bien con el servidor web de *Microsoft*, *IIS* (*Internet Information Center*).

1.1.3.2 JSP: Java Server Pages.

Esta es una tecnología basada en uno de los lenguajes de uso general más populares, *Java*. Es quizás el más complejo de los 3 pero a su vez el más potente. La gran comunidad de usuarios garantiza la existencia de numerosas herramientas, librerías de código abierto y bases de conocimiento. Además se integra muy bien en el servidor web mayoritario, *Apache*, gracias a la extensión *Tomcat*.

1.1.3.3 PHP: PHP Hypertext Preprocessor.

A su favor tiene el ser una iniciativa de código abierto, por lo que es mantenido e impulsado por una gran comunidad de programadores. Además, es el más sencillo de los tres, por lo que atrae a usuarios de muy diferentes procedencias: programadores en lenguajes de

programación más generalistas, o diseñadores que desean completar sus conocimientos.

1.2 ¿Qué es PHP?

1.2.1 Un poco de historia.

Ya hemos visto que *PHP* responde a las siglas *PHP Hypertext Preprocessor*. Pero, ¿Cómo surgió?

En el año 1994 el programador danés-canadiense *Rasmus Lerdorf* lo creó para poder programar su página web personal, que consistía básicamente en un curriculum vitae y un registro del tráfico que recibía. Un año más tarde lo publicó con el nombre de *Personal Home Page Tools*. En 1997 atrajo el interés de dos programadores israelíes del *Technion* que reescribieron el intérprete gramatical de este lenguaje creando la base de la primera versión realmente popular de *PHP*, *PHP 3*.

En 1999 reescribieron el lenguaje y lo dotaron de mayor eficiencia mediante el motor *Zend*, dando lugar a la versión estable más utilizada actualmente, *PHP 4*.

Existe ya una versión 5 de este lenguaje, que incluye un sistema novedoso, el *PDO* (Objetos de Información de *PHP* o *PHP Data Objects*) y mejoras utilizando las ventajas que provee el nuevo *Zend Engine 2*. Los cambios que incorpora la versión son notables, y posicionan a *PHP* como un lenguaje dinámico muy potente.

Durante este curso se recomienda el uso de *Xampp*, que viene con la versión 5.2.6 de *PHP*. No obstante, como el curso es de iniciación, no se hará uso de las características avanzadas, por lo que no nos afectará la versión que estemos utilizando. De todas formas siempre es conveniente conocer la versión en la que ejecutaremos nuestras aplicaciones.

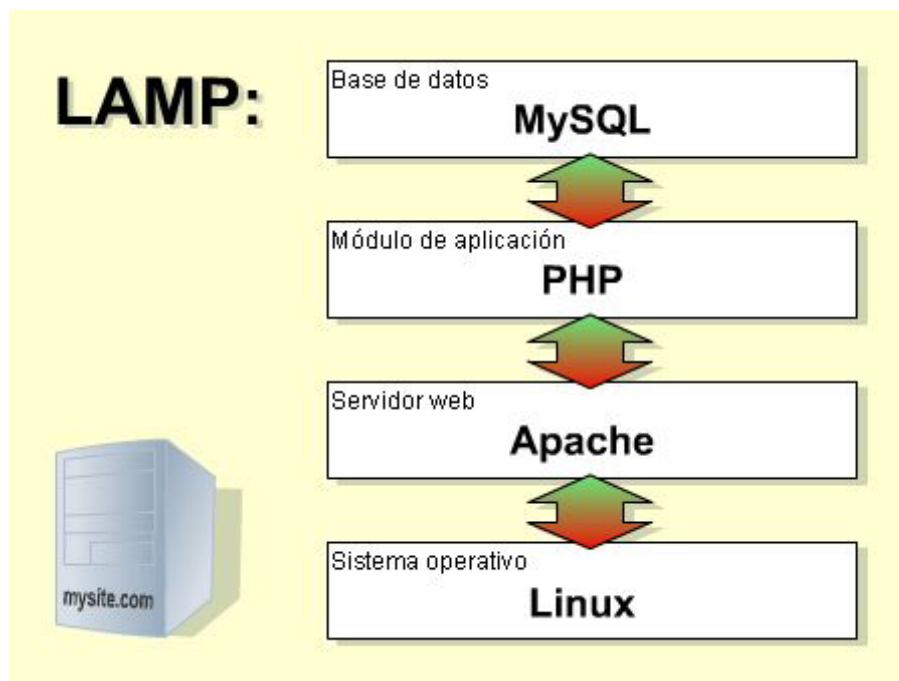
1.2.2 Características de PHP.

A nivel técnico se pueden destacar los siguientes aspectos:

- Es un lenguaje interpretado, o lo que es lo mismo, un lenguaje de tipo script. Por tanto, no requiere de compilación.
- La sintaxis es similar a los lenguajes de programación estructurada más populares, como *C*, por lo que resulta muy sencilla la transición desde otros lenguajes.
- En el código fuente se mezclan habitualmente fragmentos de código *PHP* con el cuerpo de la página *HTML*, generando dinámicamente el contenido de la misma.

- Dispone de una gran cantidad de módulos, también llamados extensiones, que amplían su funcionalidad.
- Es un lenguaje portable. Es decir, puede ser ejecutado en la mayoría de los sistemas operativos. También existen módulos intérprete para casi todos los servidores web, y en muchos de ellos viene ya instalado en el paquete básico.
- Soporta algunas características de la programación orientada a objetos.
- Se integra bien con otras tecnologías, como *XML* o *Javascript*.
- Ofrece conexión a numerosas bases de datos en modo nativo: *MySQL*, *Postgres*, *Oracle*, *SQL Server*.

Un uso común es dotar a una aplicación web de un servidor de base de datos, en el mundo *PHP* y *Apache* esta es *MySQL*, lo que en conjunto se ha denominado como la solución *LAMP*, al componerse de las siguiente herramientas: *Linux*, *Apache*, *MySQL* y *PHP*.



1.2.3 ¿Por qué PHP?

Hay muchas razones para aprender a usar *PHP*:

- Como ya se ha dicho con anterioridad, el aprendizaje de este lenguaje es muy sencillo.
- No obstante *PHP* ofrece muchas características avanzadas al alcance del programador experto.

- Es seguro y robusto, cumpliendo con las exigencias de cualquier entorno profesional.
- Es gratuito (*open source*), por lo que ha atraído a una gran comunidad de usuarios. Todos contribuyen de una u otra forma. Los desarrolladores expertos arreglando errores y aportando mejoras mientras que los usuarios testean el código e intercambian ideas que podrían mejorar el lenguaje.
- Existe una gran cantidad de aplicaciones de código abierto que están desarrolladas en *PHP*. Conociendo *PHP* se tiene siempre la posibilidad de ampliarlas, personalizarlas y proporcionarles el mantenimiento adecuado.
- La práctica totalidad de los proveedores de alojamiento de páginas web ofrece soporte para este lenguaje y acceso a una base de datos *MySQL*.

1.3 ¿Qué es Apache?

Apache es un producto software desarrollado por una asociación de carácter no lucrativo, la *Apache Software Foundation*, en adelante *ASF*. La *ASF* no sólo desarrolla el servidor *HTTP Apache*, también engloba múltiples proyectos, algunos como extensiones del servidor web y otros independientes.

Dentro de toda la gama de servidores web, *Apache* es sin duda el que mejor considerado está por parte de los usuarios. La prueba es que la mayoría de sitios web están alojados en servidores de este tipo (según la fuente http://news.netcraft.com/archives/web_server_survey.html).

Apache se ha ganado la confianza de los usuarios principalmente debido a las siguientes razones:

- Es muy eficiente, robusto y seguro.
- Dispone de características avanzadas como mensajes de error altamente configurables, contenido negociable, autenticación...
- Está disponible para la mayoría de las plataformas, incluyendo *Windows* y *Linux*.
- Es software libre, con las ventajas que esto supone. No sólo es gratuito, sino que además dispone de una numerosa comunidad de usuarios que contribuyen a su continua mejora.

La mayor objeción que se le ha puesto a *Apache* es que no dispone de un modo gráfico amigable para su configuración, por lo que su aprendizaje puede suponer más esfuerzo que el de otros servidores.

Durante este curso se propone la utilización de *Apache* como servidor al que se le añade como extensión el intérprete de *PHP*.

1.3.1 Otros servidores web.

La principal alternativa a *Apache* es el servidor *HTTP* de *Microsoft*. La ventaja de este servidor es que dispone de un interfaz gráfico amigable que hace más sencilla su instalación, configuración y despliegue de sitios web.

Existen otros servidores que, o bien son pequeños productos con pocas características, o bien son desarrollos de grandes empresas que no son accesibles al público en general.

1.4 Instalación.

Las páginas con código *PHP* son archivos que usan la extensión *".php"*, Para poder visualizarlas, deben estar alojadas en un servidor web remoto capaz de interpretarlas.

Practicar con un servidor web profesional no es factible, ya que tiene las siguientes desventajas:

- Hay que contratar un servicio de alojamiento, lo que puede resultar costoso.
- Hay dependencia de la conexión a *Internet*.
- Gestionar los archivos que subimos al servidor a través de *FTP* es algo engorroso y lento, por lo que no se pueden hacer pruebas de código con agilidad.

La solución es sencilla: haremos que nuestro ordenador actúe como servidor web. A esta forma de trabajo se le llama coloquialmente, desarrollo en local.

Para desarrollar las habilidades necesarias para el manejo e instalación de una aplicación como un servidor *Apache* y su módulo para *PHP*, se necesitaría un curso completo. Debido a ello, *Apache* ha sido criticado en este aspecto, por no contar con una instalación y/o configuración gráfica/amigable al usuario, ya sea novel o experto.

Afortunadamente existen soluciones en el mercado que integran todo lo que necesitamos para trabajar, y además disponen de los llamados

instaladores automáticos, que nos dejan preparado y, lo más importante, configurado de manera homogénea el entorno de desarrollo que vamos a emplear en todo este curso.

De entre los instaladores disponibles, hemos elegido una solución probada y que cubre varios sistemas operativos (*Windows*, *GNU/Linux* y *Mac*). Esta se llama *Xampp* en su versión reducida (*lite*).

En el curso vamos a usar la versión para *Windows*, que será la referente por facilidad de uso para los usuarios menos avanzados, aunque veremos algunas pantallas de *Linux* como alternativa (99% idénticas).

En *Linux* y otros sistemas podemos usar *Xampp* o instalar *Apache+PHP+MySQL* por nuestra cuenta. Aunque esto último sólo es recomendable para usuarios avanzados, dado que hay que configurar el sistema.



Xampp en la versión elegida instala las siguientes versiones de *Apache*, *Php* y *MySQL*:

- *Apache*: v2.2.9
- *Php*: 5.2.6
- *MySQL*: v5.0.51b

Requerimientos:

- 64 Mb de memoria *RAM*.
- 58 Mb de disco.
- Windows 98, ME, XP Home, XP Professional, Windows NT, 2000, GNU/Linux, MacOS.

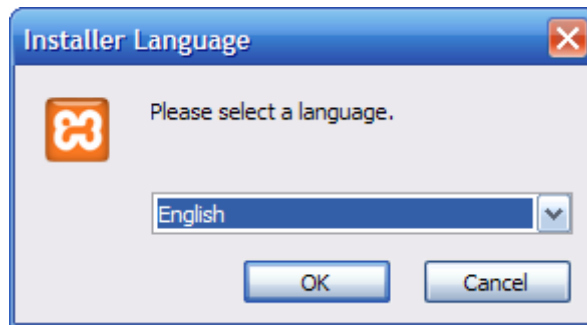
Hay que tener en cuenta que *Xampp* evoluciona muy rápidamente y que en el momento de realizar el curso puede encontrarse en una versión posterior. De todas formas es de esperar que la instalación siga siendo igual o muy similar.

1.4.1 Instalación paso a paso sobre *Windows*.

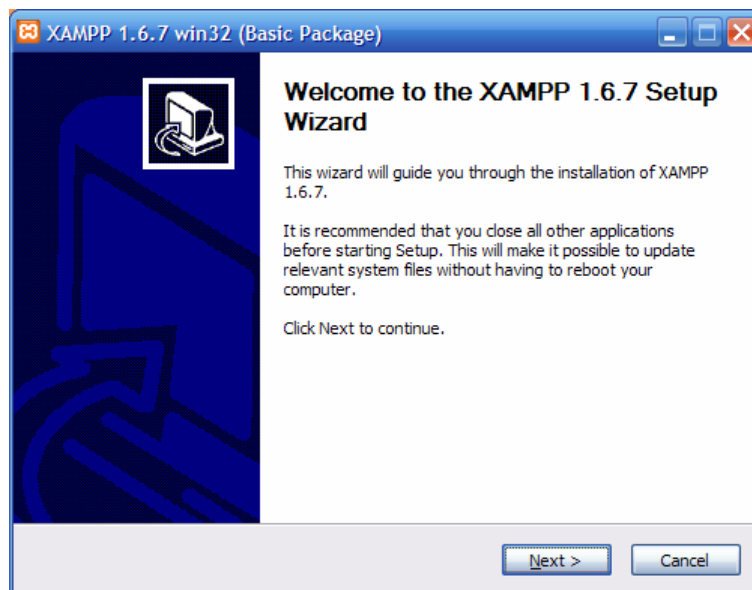
1. Descargamos la versión deseada según el sistema operativo que tengamos desde las direcciones siguientes:

- *Windows*:
<http://prdownloads.sourceforge.net/xampp/xampp-win32-1.6.7-installer.exe?download>.
- *Linux*: <http://www.apachefriends.org/en/xampp-linux.html>.

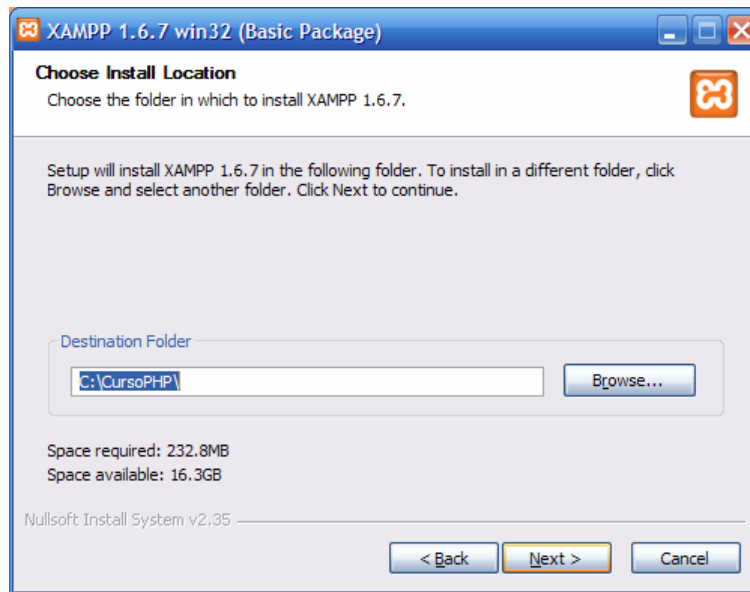
2. Ejecutamos el fichero. Al hacerlo nos aparecerá la siguiente ventana preguntando en que idioma queremos instalarlo. Dado que no disponemos del idioma español, seleccionaremos "English":



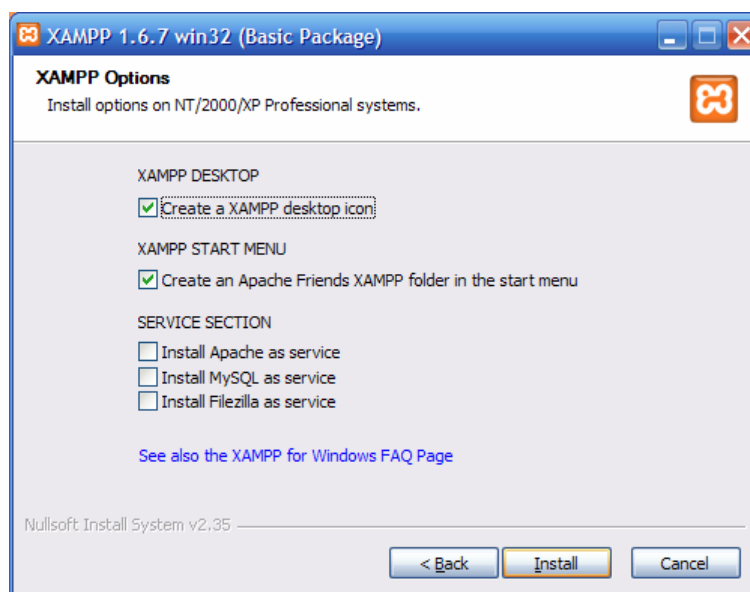
3. A continuación, el instalador nos informará de la aplicación que estamos instalando. Simplemente, pasaremos a la siguiente ventana mediante el botón "Next".



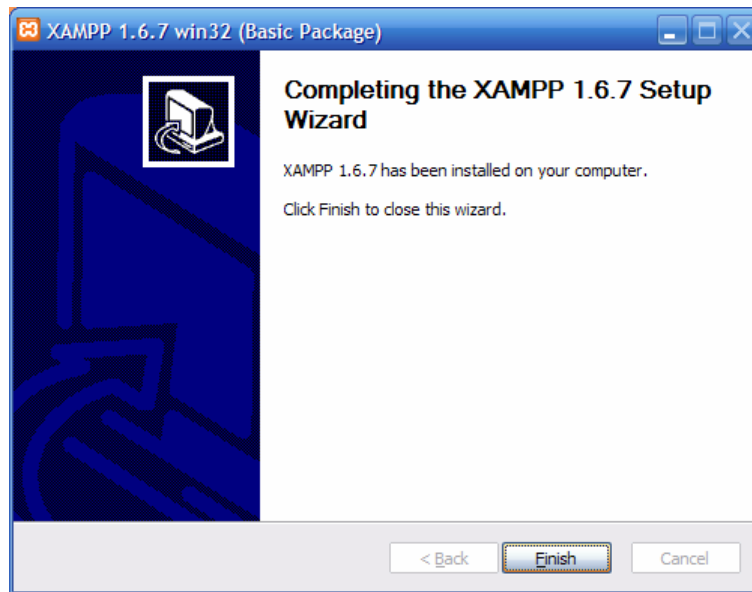
4. A continuación deberemos seleccionar la ruta de instalación, tal y como se muestra en la siguiente pantalla:



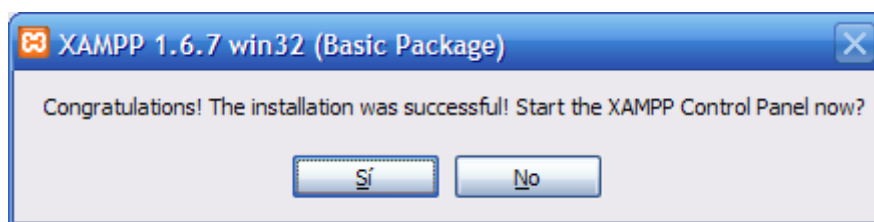
5. La siguiente ventana de la instalación nos consultará sobre las opciones de instalación, que completaremos tal y como se ve en la pantalla, asegurándonos de no instalar ninguna de las aplicaciones como servicio.



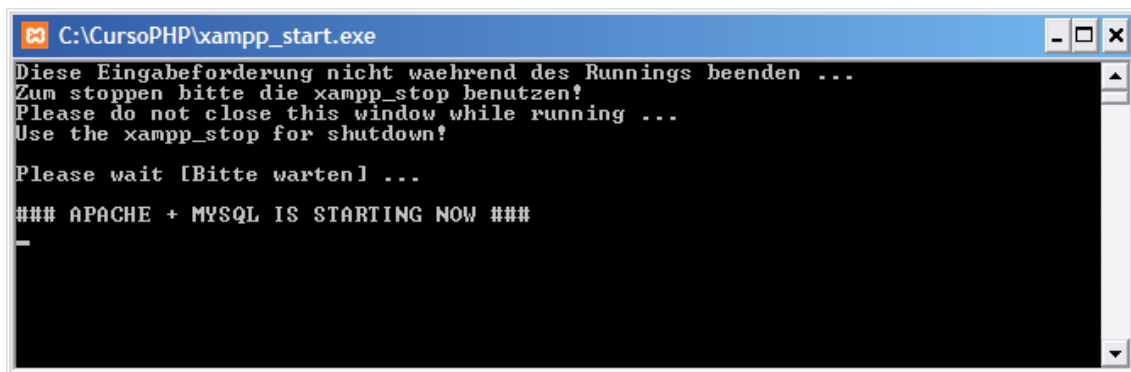
6. Como último paso de la instalación, si todo ha funcionado correctamente, se nos informará de la corrección del mismo.



7. Tras instalar la aplicación, se nos ofrecerá la posibilidad de acceder al panel de control. Sin embargo, esta acción no será necesaria.

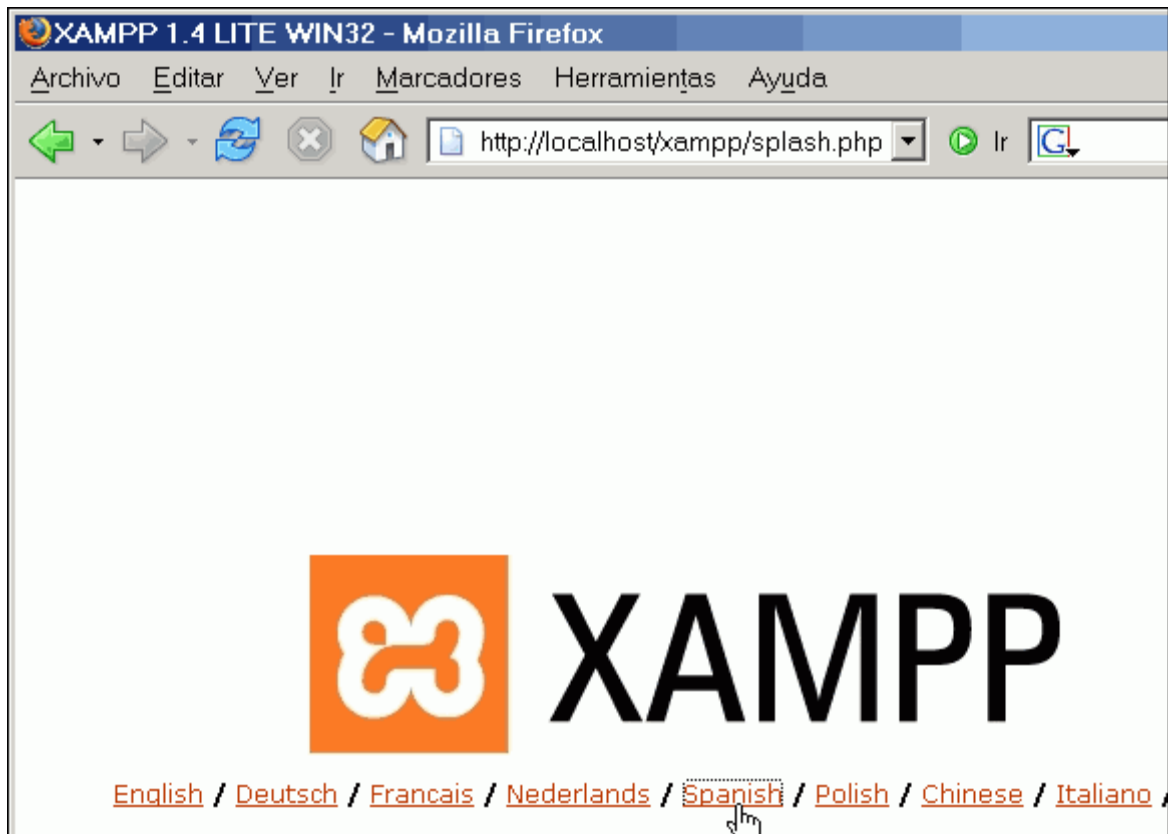


8. En este momento lo tenemos instalado, y listo para arrancar los servidores *Apache* y *MySQL*. Para ello hay que lanzar el ejecutable *xampp_start.exe* ubicado dentro de la carpeta *C:\CursoPhp*:



Deberemos mantener abierta esta ventana mientras utilicemos *PHP*. Una vez que terminemos de utilizarlo, y queramos finalizar la aplicación deberemos hacerlo mediante el ejecutable *xampp_stop.exe*, que se encargará de cerrar de forma ordenada la aplicación.

9. Ahora comprobaremos que *Apache* está funcionando correctamente. Para ello, vamos a ir a la dirección <http://localhost> con nuestro navegador favorito y deberíamos visualizar la siguiente página:

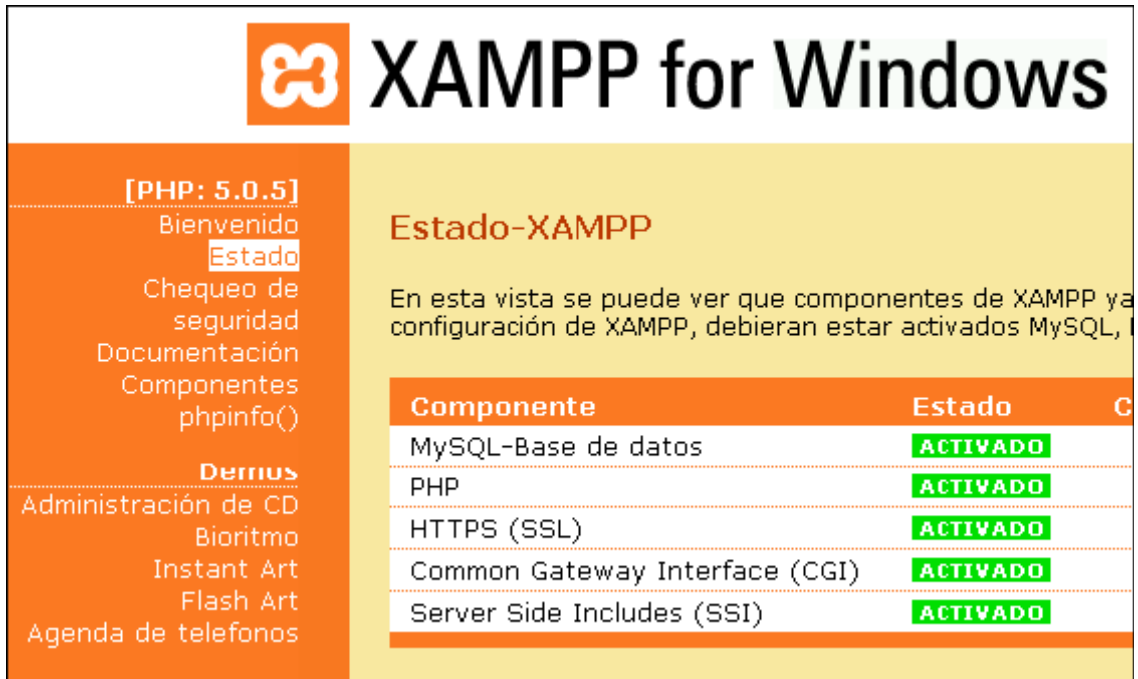


Esta página web está siendo servida por *Apache* y *PHP*. Ahora pulsaremos en el idioma español (*Spanish*) para continuar.

10. Accederemos a la página principal de *Xampp* en español, similar a la que se muestra a continuación. Desde ella podremos navegar por las demostraciones de *PHP*.



11. Pulsaremos en el enlace *"Estado"* para probar toda la instalación definitivamente:



XAMPP for Windows

[PHP: 5.0.5]
 Bienvenido
Estado
 Chequeo de seguridad
 Documentación
 Componentes
 phpinfo()

Demos
 Administración de CD
 Bioritmo
 Instant Art
 Flash Art
 Agenda de telefonos

Estado-XAMPP

En esta vista se puede ver que componentes de XAMPP ya configuración de XAMPP, debieran estar activados MySQL, I

Componente	Estado	C
MySQL-Base de datos	ACTIVADO	
PHP	ACTIVADO	
HTTPS (SSL)	ACTIVADO	
Common Gateway Interface (CGI)	ACTIVADO	
Server Side Includes (SSI)	ACTIVADO	

Para concluir, si todo ha funcionado correctamente deberemos tener activos todos los componentes que se muestran en la anterior captura.