

## EJERCICIO 1

Se desea programar una aplicación web para gestionar una lista de direcciones de correo. En una primera aproximación, la aplicación se compondrá de tres páginas:

- Una página de presentación, donde se explica el cometido de la aplicación y su funcionamiento. Contiene un enlace que lleva a la siguiente página.
- Una página de introducción de datos, donde cualquiera se puede dar de alta introduciendo su nombre y su dirección de correo. Contiene un formulario que, una vez rellenado, envía los datos y redirige a la siguiente página.
- Una página de visualización de datos, en la que se muestra la lista, con nombres y direcciones de correo, de todos los que se han anotado en la aplicación.

Responde a las siguientes preguntas sobre la aplicación que se quiere desarrollar:

- A) ¿Qué tipo de páginas, estáticas o dinámicas, utilizarás para programar cada una de las páginas que componen la aplicación web? ¿Por qué?

La primera página estática, porque el contenido nunca cambia.

La segunda y tercera dinámicas, porque interactúan con el usuario y muestra información dinámica.

- B) Si en la página web de introducción de datos quieres comprobar, antes de enviar los datos, que el correo electrónico introducido cumple unas ciertas normas (por ejemplo, que tiene una @), ¿qué tecnología/lenguaje utilizarás?

Javascript.

- C) Si en esa misma página web, ahora quieres comprobar que el correo electrónico introducido no se haya introducido anteriormente y ya figure en la lista, ¿qué tecnología/lenguaje utilizarás?

PHP.

- D) ¿Qué arquitecturas puedes usar en el servidor para ejecutar la aplicación? ¿Cómo es el o los lenguajes que se usa en cada una de esas arquitecturas: de guiones, compilado a código nativo o compilado a código intermedio?

Java EE – Compilado.

AMP – Compilado.

CGI/Perl

ASP.Net

- E) ¿Qué parámetros debes tener en cuenta para decidirte por usar una arquitectura u otra?

Principalmente hay que tener en cuenta las capacidades que tenga el servidor web y los módulos o extensiones que tenga instalados.

- F) Si te decides por utilizar una arquitectura AMP para la aplicación, ¿qué componentes necesitas instalar en tu servidor para ejecutar la aplicación? Indica algún producto concreto para cada componente.

Servidor web: Apache

Servidor de bases de datos: MySQL

Lenguaje de programación: PHP

- G) ¿Qué necesitas instalar en tu ordenador para poder desarrollar la aplicación?

Un servidor web.

## EJERCICIO 2

La arquitectura de un servidor web varía en función de la plataforma sobre la que se vaya a utilizar y las capacidades que se requieren. Algunos de los servidores web más conocidos son Apache, de la *Apache Foundation*, o Internet Information Server de *Microsoft*. Busca la última versión de estos servidores y describe tanto su arquitectura básica como los mecanismos de extensión que ofrecen para el soporte de diferentes tecnologías de programación en entorno del servidor.

Apache:

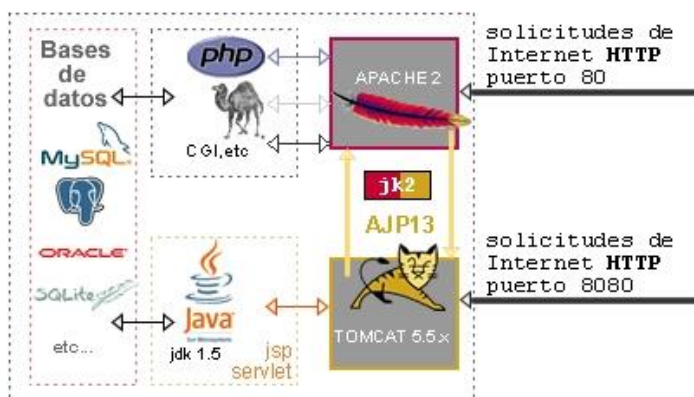
Última versión estable 2.4.27 - 11/07/2017

Los módulos del Apache se pueden clasificar en tres categorías:

Módulos Base: Módulo con las funciones básicas del Apache

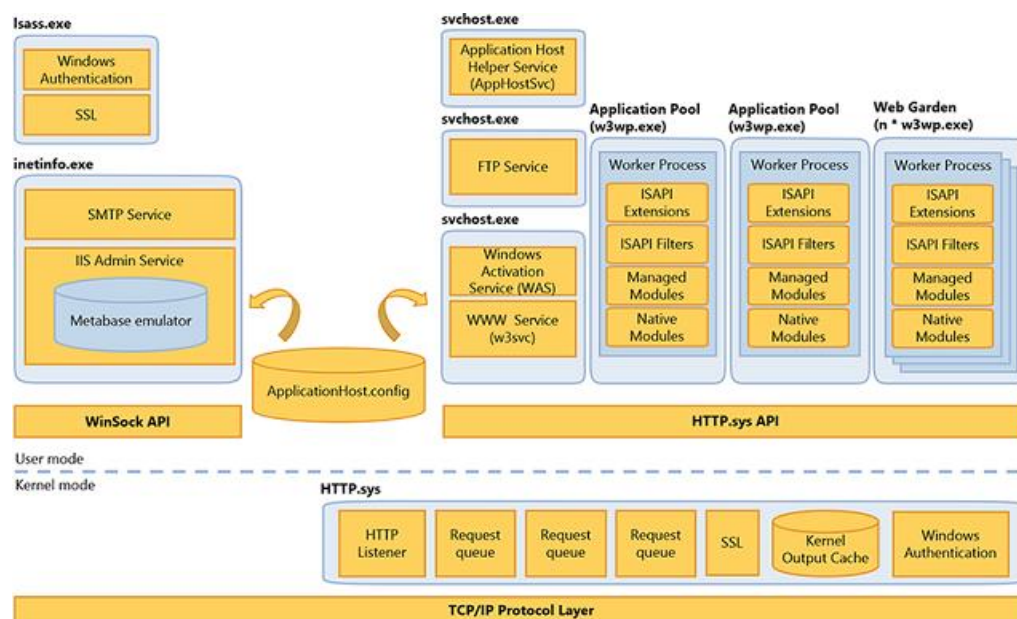
Módulos Multiproceso: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones

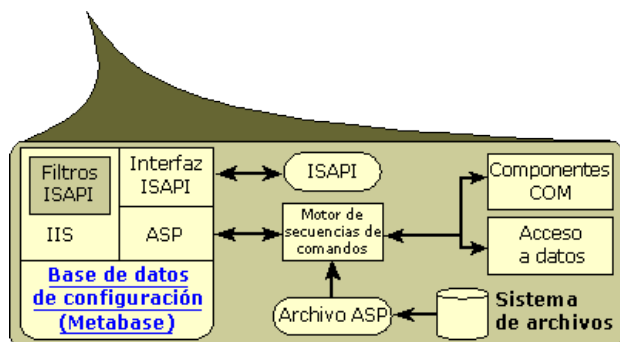
Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.



Internet Information Services:

Última versión estable IIS 10.0, Windows 10 y Windows Server 2016





### EJERCICIO 3

Los lenguajes del entorno del servidor presentados han ido evolucionando históricamente incluyendo cada vez más funcionalidades. Elige uno de esos lenguajes y realiza una descripción detallada de su evolución, indicando cuáles son las influencias recibidas de otros lenguajes y sobre qué otros lenguajes ha influido. **PHP:**

Versión	Fecha de lanzamiento	Fin de soporte	Notas
1.0	08/06/1995		Llamada oficialmente "Personal Home Page Tools (PHP Tools)". Es la primera versión en usar el nombre "PHP".
2.0	01/11/1997		Considerada por sus creadores la "herramienta más rápida y sencilla" para crear páginas web dinámicas.
3.0	06/06/1998	20/10/2000	Zeev Suraski y Andi Gutmans reescribieron la base para esta versión.
4.0	22/05/2000	23/01/2001	Se añadió un sistema de parsing de dos fases llamado motor Zend.
4.1	10/12/2001	12/03/2002	Se introducen 'superglobales' (\$_GET, \$_POST, \$_SESSION, etc.)
4.2	22/04/2002	06/09/2002	Se deshabilitan register_globals por defecto. Datos recibidos por la red no son insertados en el espacio de nombres global, cerrando posibles agujeros de seguridad en las aplicaciones.
4.3	27/12/2002	31/03/2005	Se introduce CLI, y CGI.
4.4	11/07/2005	07/08/2008	Se añaden páginas man para phpize y php-config.
5.0	13/07/2004	05/09/2005	Motor Zend II con un nuevo modelo de objetos.
5.1	24/11/2005	24/08/2006	Mejoras de rendimiento con la introducción de variables de compilador para el nuevo motor de PHP.
5.2	02/11/2006	06/01/2011	Activa extensión de filtro por defecto. Soporte JSON nativo.
5.3	30/06/2009	14/08/2014	Soporte para espacios de nombres, enlace estático en tiempo de ejecución, etiqueta de salto (GOTO limitada), Clausuras nativas, Soporte nativo para archivos PHP (phar), recolección de basura para referencias circulares, soporte mejorado para Windows, sqlite3, mysqlnd como reemplazo a libmysql como biblioteca para extensiones que funcionan con MySQL, fileinfo como reemplazo de mime_magic para mejor soporte MIME, extensión de Internacionalización, y etiqueta ereg obsoleta.
5.4	01/03/2012	15/09/2015	Soporte para Trait y sintaxis abreviada de array. Elementos eliminados: register_globals, safe_mode, allow_call_time_pass_reference, session_register(), session_unregister() y session_is_registered(). Servidor web incorporado. Varias mejoras a características existentes y al rendimiento, y requerimientos de memoria menores.
5.5	20/06/2013	20/06/2016	Nuevos generadores para bucles, empty() soporta expresiones. Se pierde el soporte para Windows XP y Windows Server 2003.
5.6	20/08/2014	31/12/2018	Constantes con expresiones escalares, listas de argumentos de longitud variable y exponencial mediante el operador **.
6.x			El desarrollo de PHP 6 ha sido retrasado porque los desarrolladores decidieron que el enfoque actual para tratar cadenas Unicode no es correcto, y están considerando formas alternas para la siguiente versión. Las mejoras planeadas para PHP 6 fueron añadidas en su lugar en PHP 5.3.0 (Soporte para espacios de nombre, enlace estático en tiempo de ejecución, funciones lambda, clausuras, goto) y 5.4.0 (traits, revinculación de clausura). Se ha abandonado la versión 6 de PHP siendo la siguiente versión la 7.
7.0	03/11/2015	03/11/2018	Mejoras de rendimiento, declaración de tipos de retorno en funciones.
7.1	01/12/2016	01/12/2019	Tipo de retorno vacío, Clase constante modificadores de visibilidad, Tipos nullable, iterable de tipo seudo, Captura múltiples tipos de excepciones.