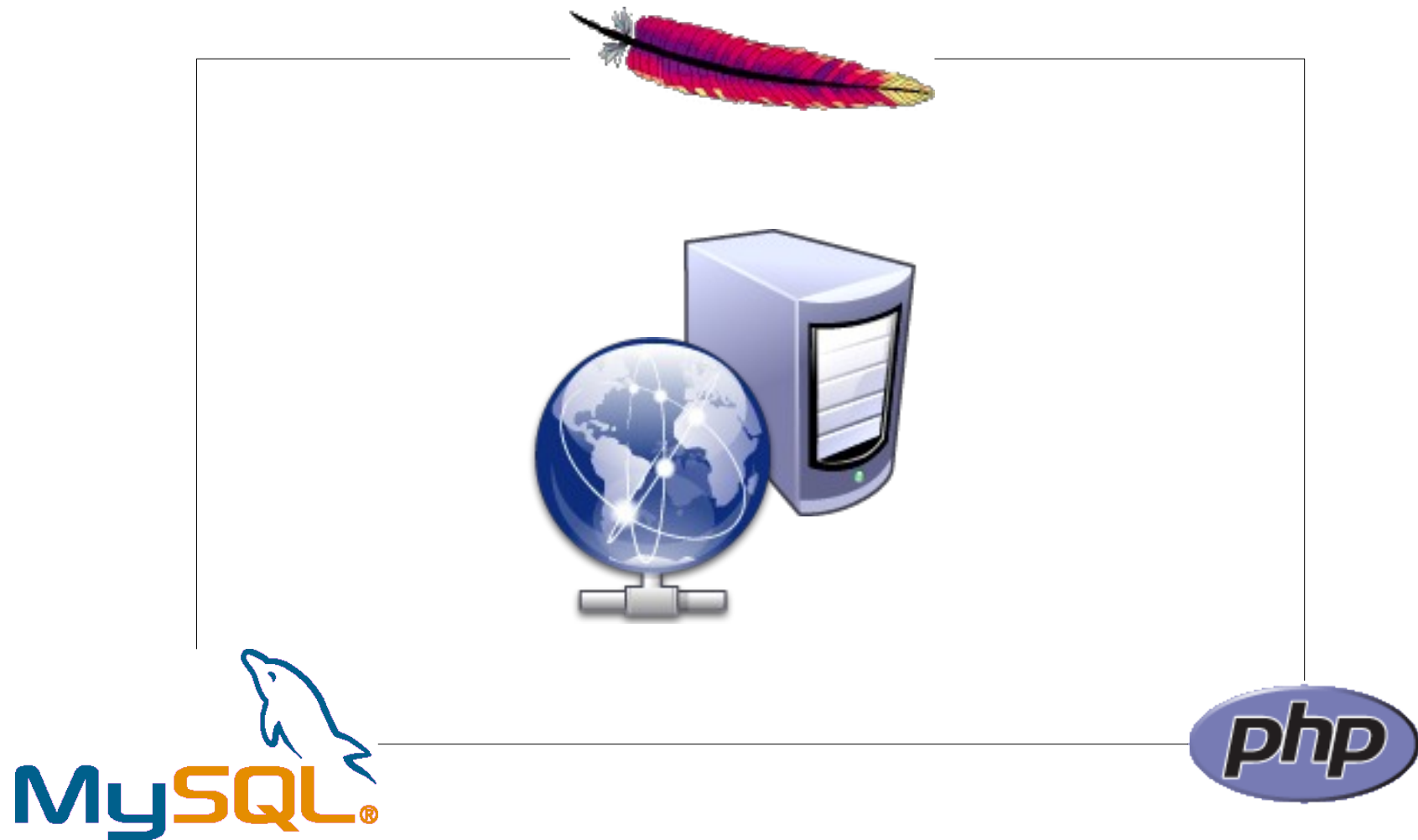


Configuración de servidores LAMP



Tomás Vilariño Fidalgo



ÍNDICE PONENCIA

LAMP

APACHE

HTTP

DIFERENCIAS ENTRE APACHE 1 / APACHE 2

PHP

MYSQL

INSTALACIÓN

MEDIANTE COMPILACIÓN

CON APT

CONFIGURACIÓN

APACHE

CONTROL DE ACCESO A DIRECTORIOS

VIRTUAL HOST

AUTENTICACIÓN

PHP

CMS

INTALACIÓN XOOPS

INTALACIÓN JOOMLA

INSTALACIÓN WORDPRESS



LAMP

LAMP hace referencia a un conjunto de programas de software libre usados de forma conjunta para ejecutar páginas web dinámicas

Linux, sistema operativo

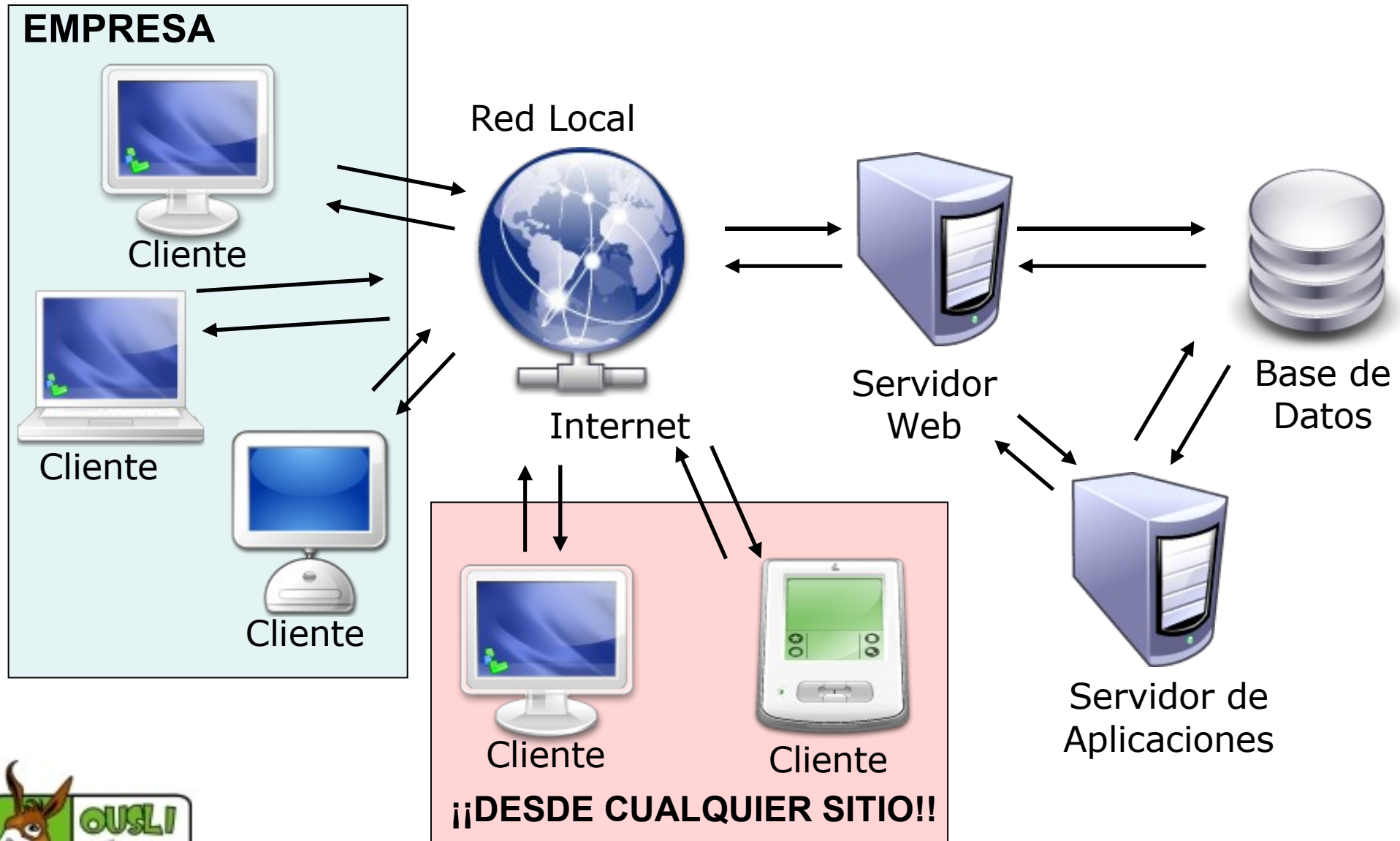
Apache, servidor web

MySQL, sistema gestor de bases de datos

Perl, PHP o Python, lenguajes de programación/scripts



Aplicaciones cliente/servidor



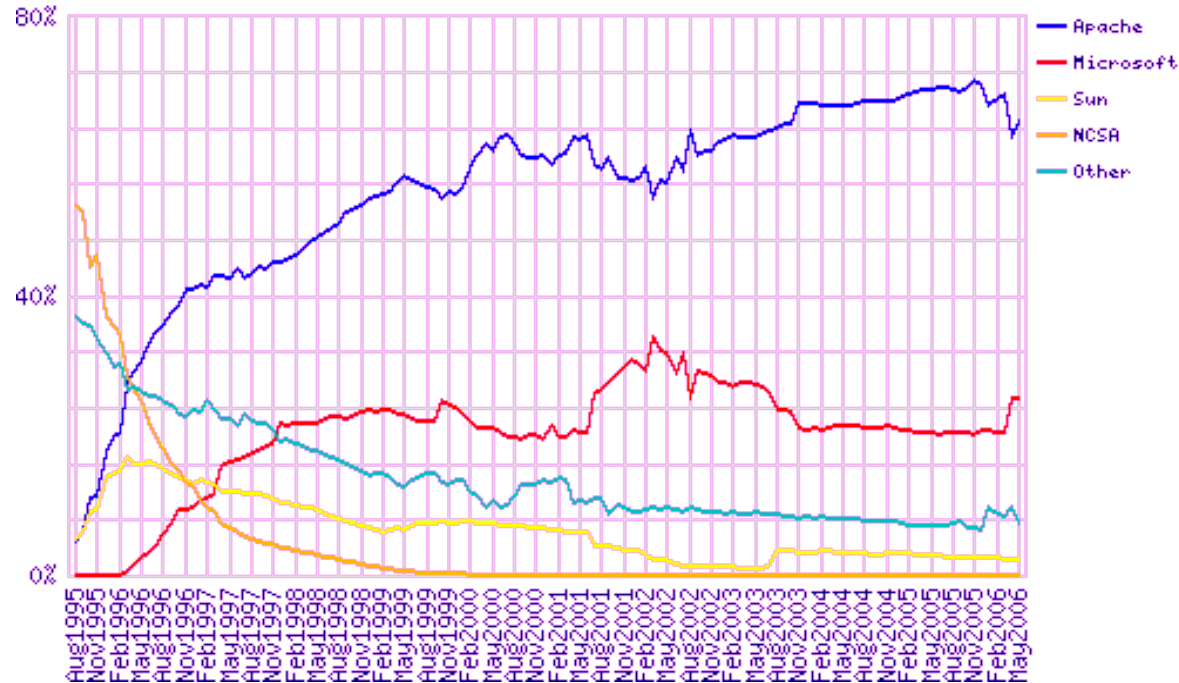
Apache

Apache es el servidor web por excelencia.

Gratuidad: Totalmente gratuito, distribuido bajo la licencia Apache Software License, que permite la modificación y adaptación de código.

Fiabilidad: Alrededor del 90% de los servidores con más alta disponibilidad funcionan con Apache.

Extensibilidad: se pueden añadir módulos para ampliar las ya de por si amplias capacidades de Apache.



HTTP

Apache es un servidor **HTTP**

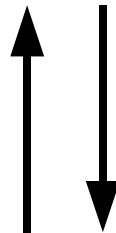
HTTP (Protocolo de Transferencia de HiperTexto): protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una web y la respuesta de esa web.

http://dirección[:puerto][path] → <http://www.ousli.org:80/path/page.php>

1. Se abre un socket con el host.

2. Se envía un mensaje request

```
GET /index.html HTTP/1.1
Host=es.wikipedia.org
User-Agent=Firefox/1.5.0.4
Accept=text/xml,
        application/xml,
        application/xhtml
Accept-Language=es-es
Accept-Charset=ISO-8859-1,utf-8
```



3. Mensaje response del servidor

```
HTTP/1.1 OK - 200
Date=Mon, 26 Jun 2006 18:32:32 GMT
Server=Apache
X-Powered-By=PHP/5.1.2
Content-Type=text/html; charset=utf-8
```

```
<html>
<body>
<h1>wikipedia</h1>
(Contenido)
</body>
</html>
```



Mensajes de estado HTTP

RANGO DE CÓDIGO	TIPO	CÓDIGO	MENSAJE	SIGNIFICADO
100-199	Informativo	100	Continue	Continúa con petición parcial (nuevo en HTTP 1.1)
		101	Switching Protocols	El servidor cumplirá con la cabecera Upgrade y cambiará a un protocolo diferente. (Nuevo en HTTP 1.1)
200 - 299	Exitoso	200	OK	Todo está bien.
		201	Created	El servidor creo un documento; la cabecera Location indica la URL.
		202	Accepted	La petición se está realizando, el proceso no se ha
		203	Non-Authoritative Information	El documento está siendo devuelto normalmente, pero algunas cabeceras de respuesta podrían ser incorrectas porque se está usando una copia del documento (Nuevo en
		204	No Content	No hay un documento nuevo; el navegador continúa mostrando el documento anterior.
		205	Reset Content	No hay documento nuevo, pero el navegador debería resetear el documento. Usado para forzar al navegador a borrar los contenidos de los campos de un formulario CGI (Nuevo en HTTP 1.1)
		206	Partial Content	El cliente envía una petición parcial con una cabecera Range, y el servidor la ha completado. (Nuevo en HTTP 1.1)



Mensajes de estado HTTP

RANGO DE CÓDIGO	TIPO	CÓDIGO	MENSAJE	SIGNIFICADO
300 - 399	Redirección	300	Multiple Choices	El documento pedido se puede encontrar en varios sitios; serán listados en el documento devuelto. Si el servidor tiene una opción preferida, debería listarse en la cabecera de respuesta Location .
		301	Moved Permanently	El documento pedido está en algún lugar, y la URL se da en la cabecera de respuesta Location.
		302	Found	Similar a 301, excepto que la nueva URL debería ser interpretada como reemplazada temporalmente.
		303	See Other	Igual que 301/302, excepto que si la petición original era POST, el documento redirigido (dado en la cabecera Location) debería ser recuperado mediante GET. (Nuevo en HTTP 1.1)
		304	Not Modified	El cliente tiene un documento en el caché y realiza una petición condicional (normalmente suministrando una cabecera If-Modified-Since indicando que sólo quiere documentos más nuevos que la fecha especificada). El servidor quiere decirle al cliente que el viejo documento del caché todavía está en uso.
		305	Use Proxy	El documento pedido debería recuperarse mediante el proxy listado en la cabecera Location. (Nuevo en HTTP 1.1)
		307	Temporary Redirect	Es idéntica a 302 ("Found" o "Temporarily Moved"). Fue añadido a HTTP 1.1 ya que muchos navegadores siguen erróneamente la redirección de una respuesta 302 incluso si el mensaje original fue un POST, y sólo se debe seguir la redirección de una petición POST en respuestas 303. Esta respuesta es algo ambigua: sigue el redireccionamiento para peticiones GET y POST en el caso de respuestas 303, y en el caso de respuesta 307 sólo sigue la redirección de peticiones GET.



Mensajes de estado HTTP

RANGO DE CÓDIGO	TIPO	CÓDIGO	MENSAJE	SIGNIFICADO
400 – 499	Error del Cliente	400	Bad Request	Mala Sintaxis de la petición.
		401	Unauthorized	El cliente intenta acceder a una página protegida por password sin las autorización apropiada.
		403	Forbidden	Falta permisos de fichero o directorios en el
		404	Not Found	No se pudo encontrar el recurso en esa dirección.
		405	Method Not Allowed	El método de la petición (GET, POST, HEAD, DELETE, PUT, TRACE, etc.) no estaba permitido para este recurso particular. (Nuevo en HTTP 1.1)
		406	Not Acceptable	El recurso indicado genera un tipo MIME incompatible con el especificado por el cliente mediante su cabecera Accept. (Nuevo en HTTP 1.1)
		407	Proxy Authentication Required	Similar a 401, pero el servidor proxy debería devolver una cabecera Proxy-Authenticate. (Nuevo en HTTP 1.1)
		408	Request Timeout	El cliente tarda demasiado en enviar la petición. (Nuevo en HTTP 1.1)
		411	Length Required	El servidor no puede procesar la petición a menos que el cliente envíe una cabecera Content-Length. (Nuevo en HTTP 1.1)
		412	Precondition Failed	Alguna condición previa especificada en la petición era falsa (Nuevo en HTTP 1.1)
		415	Unsupported Media Type	La petición está en un formato desconocido. (Nuevo en HTTP 1.1)



Mensajes de estado HTTP

RANGO DE CÓDIGO	TIPO	CÓDIGO	MENSAJE	SIGNIFICADO
500 – 599	Error Servidor	500	Internal Server Error	Mensaje genérico "server is confused".
		501	Not Implemented	El servidor no soporta la funcionalidad de rellenar peticiones. Usado, por ejemplo, cuando el cliente envía comandos como PUT que el cliente no soporta.
		502	Bad Gateway	Usado por servidores que actúan como proxies o gateways; indica que el servidor inicial obtuvo una mala respuesta desde el servidor remoto.
		503	Service Unavailable	El servidor no puede responder debido a mantenimiento o sobrecarga.
		504	Gateway Timeout	Usado por servidores que actúan como proxies o gateways; indica que el servidor inicial no obtuvo una respuesta a tiempo del servidor remoto. (Nuevo en HTTP 1.1)
		505	HTTP Version Not Supported	El servidor no soporta la versión de HTTP indicada en la línea de petición. (Nuevo en HTTP 1.1)



MEJORAS APACHE2

Procesamiento en Unix

Nuevo sistema de configuración y compilación

Soporte Multiprotocolo

Soporte mejorado para las plataformas que no son tipo Unix

Nueva interfaz de programación (API) de Apache

Soporte de IPv6

Filtros

Mensajes de error en diferentes idiomas

Configuración simplificada

Soporte de Unicode Nativo para Windows NT

Actulización de la librería de expresiones regulares (regular expressions)



PHP

PHP (acrónimo de "PHP: Hypertext Preprocessor", hipertexto preprocesado) es un lenguaje **interpretado**, de **alto nivel**, **embebido** en páginas HTML y **ejecutado en el servidor**.

Interpretado: Son procesados mediante un intérprete, que realiza la traducción del programa a código fuente, a medida que es necesario y normalmente, no guardan el resultado de dicha traducción.

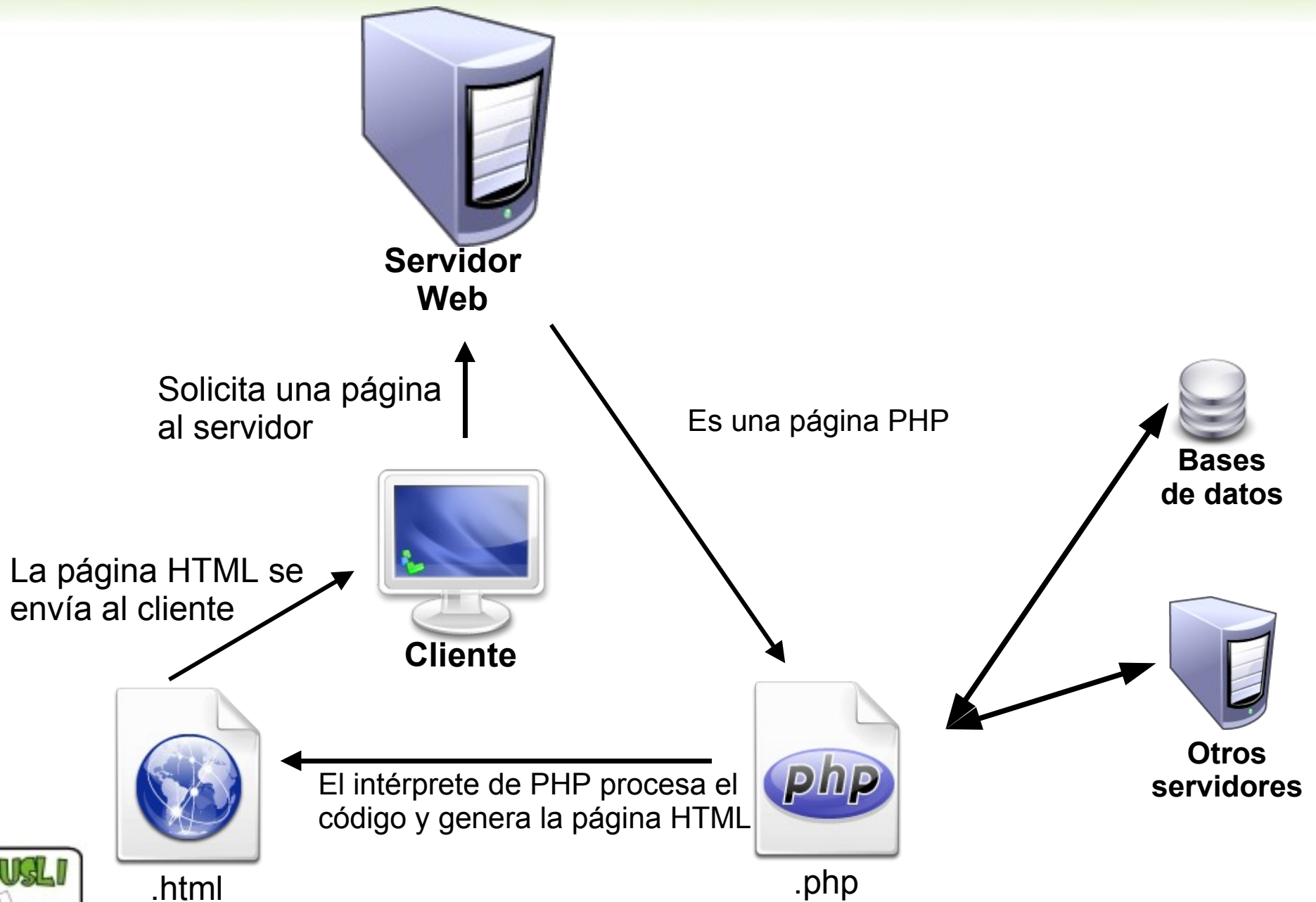
Alto nivel: Lenguaje de programación, cuyas instrucciones son similares al lenguaje humano, que necesita ser traducido a un lenguaje de bajo nivel que el sistema pueda entender.

Embebido: "incrustado" en el código HTML, lo que permite añadir el código dinámico entre las etiquetas `<?php ?>`

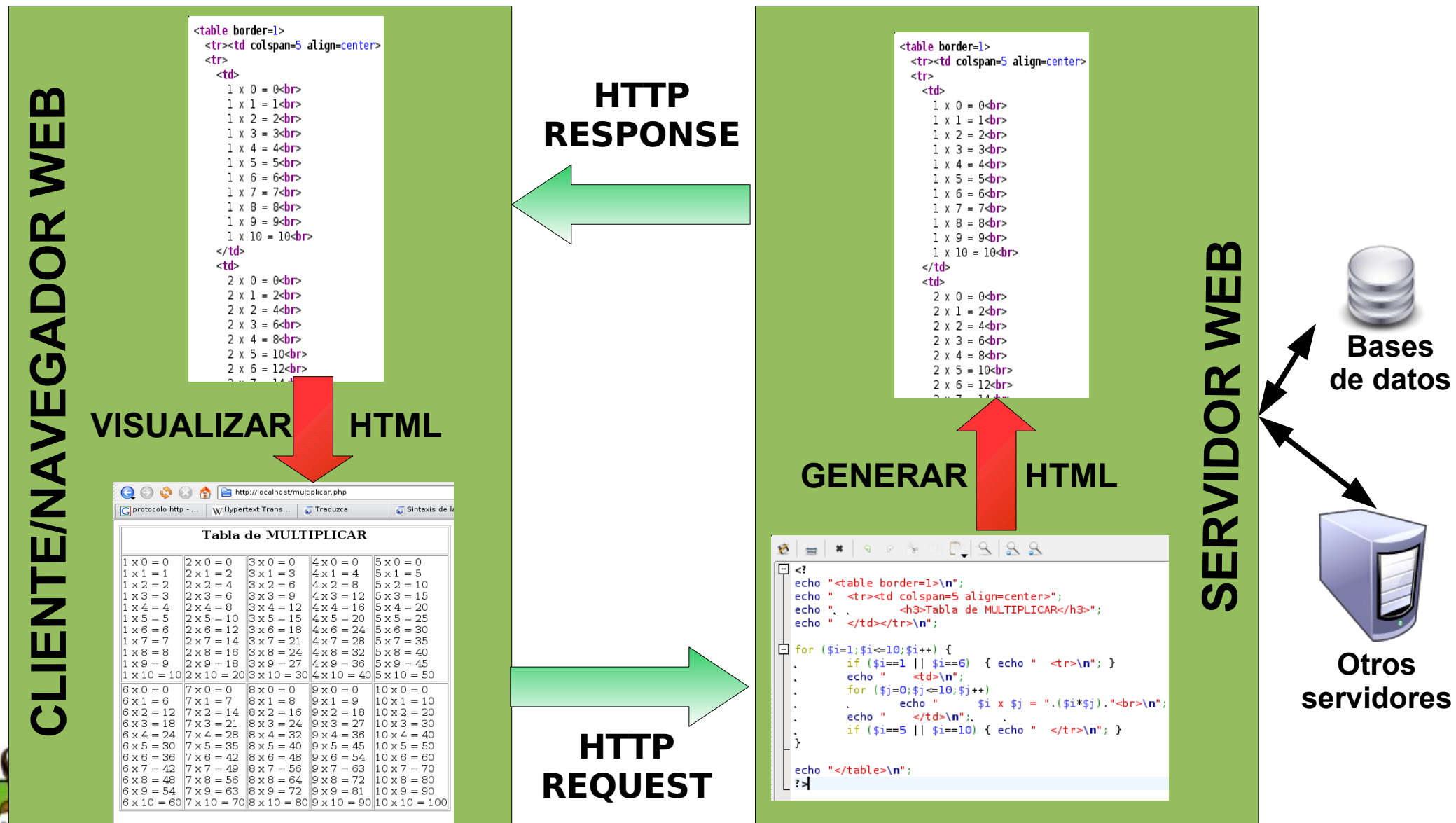
Ejecutado en el servidor: PHP es ejecutado en el servidor y nunca es enviado al cliente. El servidor es el que se encarga de procesar la petición del cliente, ejecutando el código PHP y devolviendo el HTML resultante



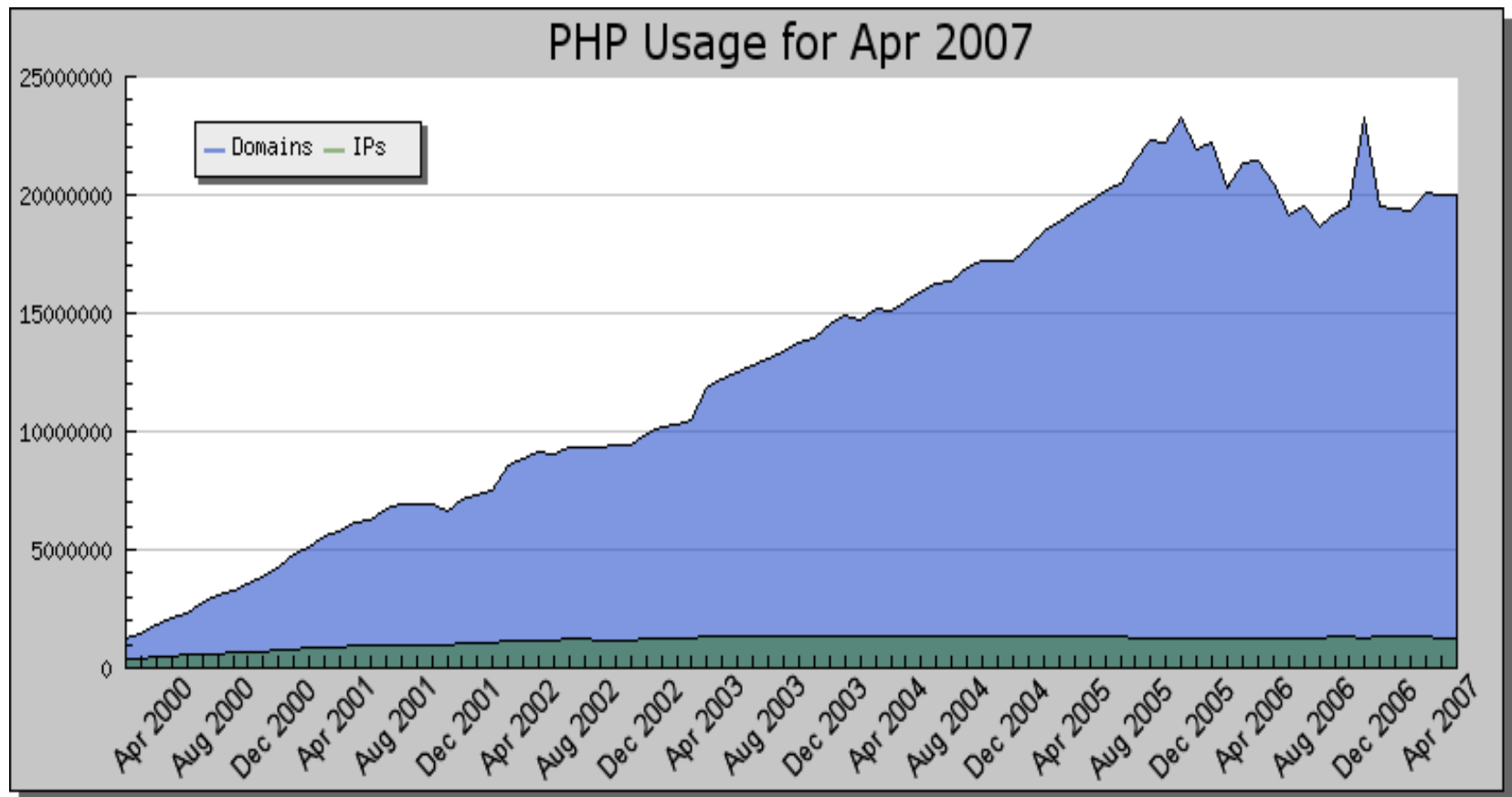
PHP - Como funciona



PHP – Como funciona



Estadísticas de uso de PHP



MySQL

MySQL es un Sistemas Gestores de bases de Datos

Multiusuario: Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones

Seguridad: en forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas.

Potencia: SQL es un lenguaje muy potente para consulta de bases de datos.

Portabilidad: Disponible para distintas plataformas.

Escalabilidad: óptimos resultados en la manipulación de bases de datos muy grandes.

Conectividad: permite conexiones entre diferentes máquinas con distintos sistemas operativos.



Instalación LAMP

Varias alternativas a la hora de instalar LAMP:

Bajarse los fuentes de cada uno de ellos, compilarlos e instalarlos.

Utilizar el administrador de paquetes específico de la distribución en la que vamos a instalar LAMP.



Instalación: compilar e instalar

APACHE:

1. Descargar: <http://httpd.apache.org/download.cgi>
2. Descomprimir: `$ tar xvfz httpd-2_0_<version>.tar.gz`
3. Ejecutar script configure: `$./configure --prefix=<path_instalación>
--enable-so`
4. Compilar: `$ make`
5. Instalar: `$ make install`
6. Configurar: `$ vi <path_instalación>/conf/http.conf`
7. Comprobar que funciona: `<path_instalación>/bin/apachectl start`



Instalación: compilar e instalar

MYSQL:

1. Descargar: <http://dev.mysql.com/downloads/mysql/5.0.html>
2. Descomprimir: `$ tar xvfz mysql-5<version>.tar.gz`
3. Ejecutar script configure: `$./configure -prefix=<path_instalación> -sysconfdir=<path_config>`
4. Compilar: `$ make`
5. Instalar: `$make install`
6. Creamos las BD propias de mysql: `$./bin/mysql_install_db`
7. Arrancamos el demonio: `./bin/mysqld_safe`



Instalación: compilar e instalar

PHP:

1. Descargar: <http://es.php.net/get/php-5.1.4.tar.gz/from/a/mirror>
2. Descomprimir: `$ tar xvfz php-5<version>.tar.gz`
3. Ejecutar script configure: `$./configure --with-apxs2=<path-apache>/bin/
--with-mysql`
4. Compilar: `$ make`
5. Instalar: `$make install`
6. Copiamos fichero de configuración: `cp php.ini-dist /usr/local/lib/php.ini`
7. Cargar el módulo: `a2enmod php5`
8. Añadir en http.conf:
`AddType application/x-httpd-php .php .phtml`
`AddType application/x-httpd-php-source .phps`
9. Reiniciar apache: `<path_apache>/bin/apachectl restart`



Instalación LAMP Ubuntu 6.10

Instalación apache2

```
$ apt-get install apache2
```

Instalación php5

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

Instalación mysql5

```
$ apt-get install mysql-server php5-mysql
```



CONFIGURACIÓN



Configuración apache2 - directorios

Paths de apache2:

Ficheros de configuración: /etc/apache2

Binarios: /usr /sbin

Librerías: /usr/lib/apache2

Directorio para las web por defecto: /var/www

Directorio de logs -> /var/log/apache2

Directorios y ficheros de configuración:

apache2.conf -> Fichero principal de configuración

mods-available/ -> Módulos disponibles

mods-enable/ -> Módulos habilitados

sites-available/ -> Sitios disponibles

sites-enable/ -> Sitios habilitados

ports.conf -> Lista de puertos en los que escucha apache

http.conf -> añadir módulos con problemas de compatibilidad



Vistazo genérico a apache2.conf

Algunas directivas a destacar de apache2.conf:

ServerRoot "/etc/apache2"

Timeout 300 (Máximo tiempo para procesar una petición)

KeepAlive On (conexión persistente)

MaxKeepAliveRequests 100

KeepAliveTimeout 15

User www-data (usuario específico apache)

Group www-data (grupo específico usuario apache)

AddType MIME-type extension [extension] (Asocia una extensión a un tipo de contenido)

Include file-path|directory-path (incluye ficheros de configuración)



Ficheros de configuración

En los ficheros de configuración de apache son fundamentales tres conceptos

Directiva de configuración: Un comando de configuración que controla uno o más aspectos del comportamiento de Apache

Secciones de configuración: Especifican a que directorio, fichero, host o URL son aplicadas determinadas directivas de aplicación.

Contexto: Un área en los ficheros de configuración donde están permitidos ciertos tipos de directivas.



Contextos

Server config: La directiva puede ser usada en los ficheros de configuración del servidor (por ejemplo, httpd.conf), pero no dentro de las secciones <VirtualHost> ni <Directory>. También significa que la directiva no puede usarse en los ficheros .htaccess.

Virtual host: Este contexto significa que la directiva puede aparecer dentro de las secciones <VirtualHost> de los ficheros de configuración del servidor.

Directory: Una directiva marcada como válida en este contexto puede usarse en las secciones <Directory>, <Location>, y <Files> en los ficheros de configuración del servidor.

.htaccess: Si una directiva es válida en este contexto, eso significa que puede aparecer en los ficheros .htaccess.



Secciones de configuración

Las directivas presentes en los ficheros de configuración pueden ser de aplicación para todo el servidor, o puede que su aplicación se limite solamente a determinados directorios, ficheros, hosts, o URLs, através secciones de configuración o ficheros .htaccess

Tipos de Secciones de configuración de apache para el módulo básico de apache

<Directory>

<DirectoryMatch>

<Files>

<FilesMatch>

<IfDefine>

<IfModule>

<Location>

<LocationMatch>

<VirtualHost>

Dos tipos de secciones de configuración:

Evalúadas para cada petición que se recibe y aplican las directivas que se incluyen en las distintas secciones.

Evalúan solamente al inicio o reinicio del servidor. Si al iniciar el servidor las condiciones son las adecuadas, las directivas que incluyen estas secciones se aplicarán a todas las peticiones que se reciban. Es caso contrario, esas directivas que incluyen se ignoran completamente.



Configuración de áreas

Las secciones de configuración más frecuentes, son las de configuración de áreas del sistema de ficheros o del espacio web

Área del sistema de ficheros: Árbol de directorios según el sistema operativo
/etc/apache2

Área del espacio web: Contenido del servidor y que el cliente visualiza. De manera que la ruta /dir/ en el espacio web se corresponde con la ruta /var/www/dir/

Secciones relacionadas con el sistema de ficheros:

<Directory> y <DirectoryMatch>: Las directivas incluidas en estas secciones, se aplican al directorio del sistema de ficheros especificado y a sus subdirectorios. El mismo resultado puede obtenerse usando ficheros .htaccess

<Files> y <FilesMatch>: Las directivas incluidas en estas secciones, se aplican al fichero especificado.

Secciones relacionadas con el sistema de ficheros:

<Location> y <LocationMatch>: Las directivas incluidas en estas secciones cambian la configuración para el contenido del espacio web



Directivas de configuración

Veremos las más genéricas, para a continuación, explicar las más específicas en sus correspondientes contextos:

DirectoryIndex: Fichero por defecto de index cuando se especifica un directorio

ServerAdmin: E-mail devuelto junto cualquier mensaje de error que envía el servidor.

DocumentRoot: Directorio raíz

Options: Funcionalidades del servidor disponibles en un directorio en particular:

None : ninguna funcionalidad disponible

All: Todas las opciones excepto MultiViews. Este es el valor por defecto

FollowSymLinks: El servidor seguirá los enlaces simbólicos en este directorio

Includes: Permite el uso de Server-side includes

Indexes: Si se produce una petición a una URL que se corresponde con un directorio, y no hay DirectoryIndex en ese directorio, entonces devolverá una lista con los contenidos del directorio.

MultiViews: Múltiples tipos de ficheros

SymLinksWithOwnerMatch: El servidor seguirá los enlaces simbólicos en los que el fichero o directorio final pertenezca al mismo usuario que el enlace.

ErrorLog: Ubicación del fichero en el que se almacenan los mensajes de error

ErrorLog: Ubicación del fichero en el que se almacenan los mensajes de error

LogLevel: Nivel de log a partir se registran los logs



<Directory> y <DirectoryMatch>

Se usan para englobar un grupo de directivas que se aplicarán solamente al directorio especificado y a sus subdirectorios

```
<Directory /var/www>
    Options Indexes FollowSymLinks
</Directory>
```

```
<Directory ~ "^/www/.*/[0-9]{3}">
```

```
<DirectoryMatch "^/www/.*/[0-9]{3}">
```

Directivas importantes de Directory:

Order: Control de acceso a ficheros y directorios y el orden en el que se aplican las restricciones
Opciones: *Deny,Allow* | *Allow,Deny* | *Mutual-failure*

Allow & Deny: Controlan que host pueden acceder

uso: Allow from all|host|env=env-variable [host|env=env-variable]

Con variables de entorno

All – Cualquiera

Host – Especificado mediante:

Nombre de dominio (parcial): Allow from apache.org

IP completa: Allow from 192.168.1.104 192.168.1.205

IP parcial: Allow from 192

Máscara de red: Allow from 10.1.0.0/255.255.0.0

Ipv6: Allow from 2001:db8::a00:20ff:fea7:ccea

```
SetEnvIf User-Agent .*Firefox.* nav
<Directory /docroot>
    Order Deny,Allow
    Deny from all
    Allow from env=nav
</Directory>
```



<Files> y <Location>

Las directivas incluidas en una sección <Files> se aplicarán a cualquier fichero cuyo nombre se especifique, sin tener en cuenta en que directorio se encuentra.

```
<Files info.php>
    Order allow,deny
    Deny from all
</Files>
```

Para referirse a archivos que se encuentren en un determinado lugar del sistema de ficheros, se pueden combinar las secciones <Files> y <Directory>

```
<Directory /var/www/>
    <Files info.php>
        Order allow,deny
        Deny from all
    </Files>
</Directory>
```

Una sección <Location> aplica las directivas que contiene según la URL de que se trate

```
<Location /phpmyadmin>
    Order Deny,Allow
    Deny from all
    Allow from 127.0.0.1
</Location>
```



<VirtualHost>

Contiene las directivas que se aplican solo a un nombre de host específico o dirección IP

El hosting virtual basado en IPs usa la dirección IP de la conexión para determinar qué host virtual es el que tiene que servir. Por lo tanto, necesitará tener diferentes direcciones IP para cada host. Si usa hosting virtual basado en nombres, el servidor atiende al nombre de host que especifica el cliente en las cabeceras de HTTP. Usando esta técnica, una sola dirección IP puede ser compartida por muchos sitios web diferentes.

Uso: <VirtualHost addr[:port] [addr[:port]] ...> ... </VirtualHost>

Addr puede ser:

- La dirección IP del host virtual

- Un nombre de dominio completo para la dirección IP del host virtual

- El carácter *, el cual puede usarse en combinación con *NameVirtualHost ** para que equivalga a todas las direcciones IP

- La cadena de caracteres *_default_*, que se usa solo con hosting virtual IP para detectar direcciones IP sin emparejar

```
<VirtualHost *>
  ServerName xornadas.com
  DocumentRoot /var/www/xornadas
  ErrorLog /var/log/apache2/xorn.log
  LogLevel warn
</VirtualHost>
```

```
NameVirtualHost 10.0.0.116

<VirtualHost 10.0.0.116>
  ServerName xornadas.com
  DocumentRoot /var/www/xornadas
</VirtualHost>
```



Alias

Nos permite el acceso a directorios que no están dentro del DocumentRoot

```
Alias /phpmyadmin /usr/share/phpmyadmin
```

```
Alias /phpmyadmin /usr/share/phpmyadmin  
<Directory /usr/share/phpmyadmin>  
    Order allow,deny  
    Allow from all  
</Directory>
```



.htaccess

Ficheros que controlan el acceso en el directorio en el que se encuentran y en todos sus subdirectorios.

Podríamos decir que es un *alias* de <Directory>

```
Order Allow,Deny  
Deny from all
```

El uso de fichero .htacces pueden ser deshabilitados completamente con la directiva AllowOverride None

```
<Directory path_htacces >  
    AllowOverride None  
</Directory>
```



Autenticación, Autorización y Control de Acceso

La autenticación es cualquier proceso mediante el cual se verifica que alguien es quien dice ser.

PROCESO


Es necesario que las directivas tratadas a continuación, pertenezcan en el archivo de configuración principal de su servidor, típicamente en una sección del tipo <Directory> o en ficheros .htaccess (e caso de utilizar .htaccess tiene que existir la directiva AllowOverride AuthConfig)

1. Crear un archivo de contraseñas. Este fichero debería estar en un directorio no accesible por web. Para crearlo utilizamos la herramienta htpasswd

`htpasswd -c <archivo de contraseña> <usuario>`

```
toni@tkubuntu:/var/www/prueba$ sudo htpasswd -c pswd toni
New password:
Re-type new password:
Adding password for user toni
```

2. Configurar el servidor para que al acceder al directorio pida la autenticación



```
<Directory /var/www/prueba>
  AuthType Basic
  AuthName "Restricted Files"
  AuthUserFile /var/www/prueba
  Require user vifito
  AllowOverride none
</Directory>
```

```
<Directory /var/www/prueba>
  AllowOverride AuthConfig
</Directory>
```

```
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /var/www/prueba
Require user vifito
AllowOverride AuthConfig
```

Autenticación, Autorización y Control de Acceso

De esta forma solo puede acceder un usuario. Para permitir el acceso a un grupo. Para ello se utiliza la directiva AuthGroupFile

1. Debemos crear un archivo de grupo que asocie nombres de grupo con una lista de usuarios perteneciente a ese grupo. El archivo debe ser como este:

```
GroupName: vifito xornadas
```

2. Para agregar un usuario a un archivo de contraseñas ya existente, escriba:

```
htpasswd /usr/local/apache/passwd/passwords xornadas
```

3. modificar la sección <Directory> o el .htacces

```
<Directory /var/www/prueba>  
    AuthType Basic  
    AuthName "Restricted Files"  
    AuthUserFile /var/www/prueba/pswd  
    AuthGroupFile /var/www/prueba/group  
    Require group GroupName  
</Directory>
```

```
<Directory /var/www/prueba>  
    AllowOverride AuthConfig  
</Directory>
```

```
AuthType Basic  
AuthName "Restricted Files"  
AuthUserFile /var/www/prueba/pswd  
AuthGroupFile /var/www/prueba/group  
Require group GroupName
```



Configuración de php

El fichero de configuración de php se encuentra en:

`/etc/php5/apache2/php.ini`

Algunas variables de configuración

`register_globals=off`

`max_execution_time = 30`

`max_input_time = 60`

`memory_limit = 8M`

`error_reporting = E_ALL & ~E_NOTICE`

`post_max_size = 8M`



Register Globals off

La variable `register_globals` de configuración establece la forma en la que accedemos a las variables pasadas por GET ó POST al servidor.

Si `register_globals=on` podremos acceder a las variables de forma natural.

Ej: *`http://xornadas.com?op=inscripción`*

`<? $op; ?>`

Si `register_globals=off` tendremos que acceder por la variable superglobal específica.

`POST=> $_POST["op"];`

`GET=> $_GET["op"];`



CMS

CMS (Content Management System): Sistema de gestión de contenido que permite la creación y administración de contenidos principalmente en páginas web.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido por una parte y el diseño por otra. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores. Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público.



INSTALACIÓN CMS



Bibliografía

<http://google.com>

<http://www.mysql-hispano.org/>

<http://www.php.net>

<http://es.wikipedia.org>

<http://news.netcraft.com/>

http://www.programacion.com/java/tutorial/servlets_jsp/7/

<http://httpd.apache.org/docs/2.0/>

<http://www.lfcia.org/openprojects/camllets/doc/html/node16.html>



Licencia de la ponencia

Este trabajo se encuentra bajo la licencia **Creative Commons Attribution-ShareAlike 2.5 License**

Autor original: Toni Martínez Carballo.

Modificaciones: Tomás Vilariño Fidalgo.



GRACIAS POR TU
ATENCIÓN

¿PREGUNTAS?

