

Servidores Web – Apache

Aplicaciones Web/Sistemas Web

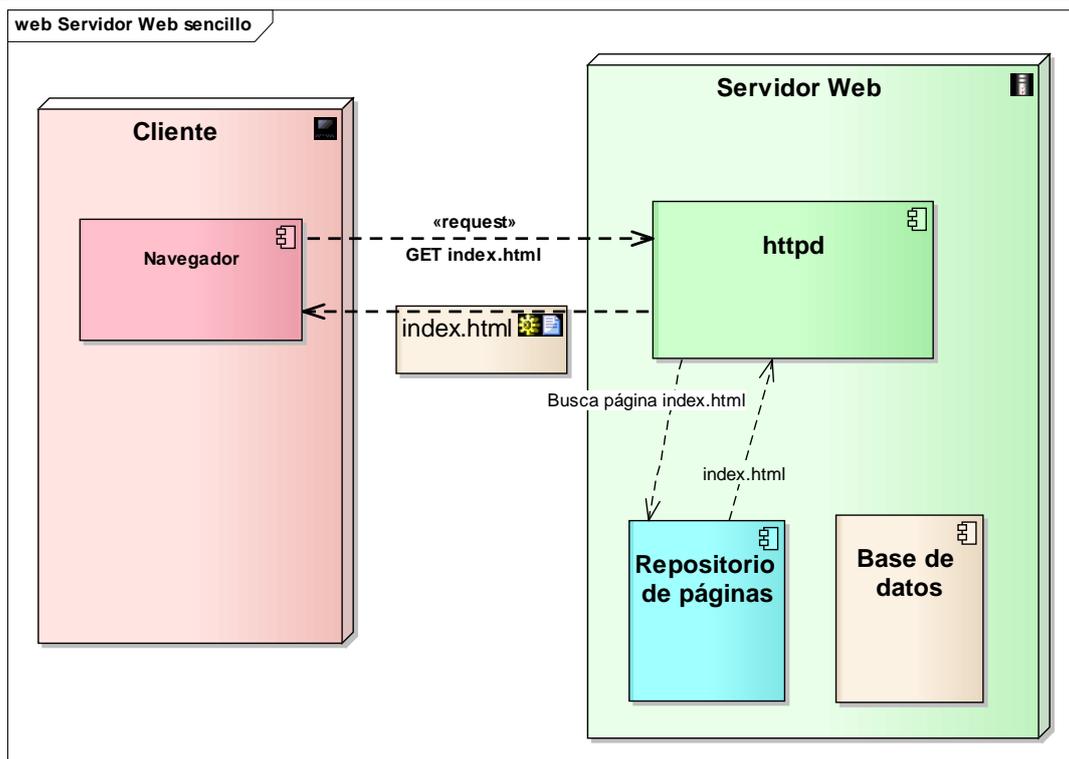


Juan Pavón Mestras
Dep. Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense Madrid

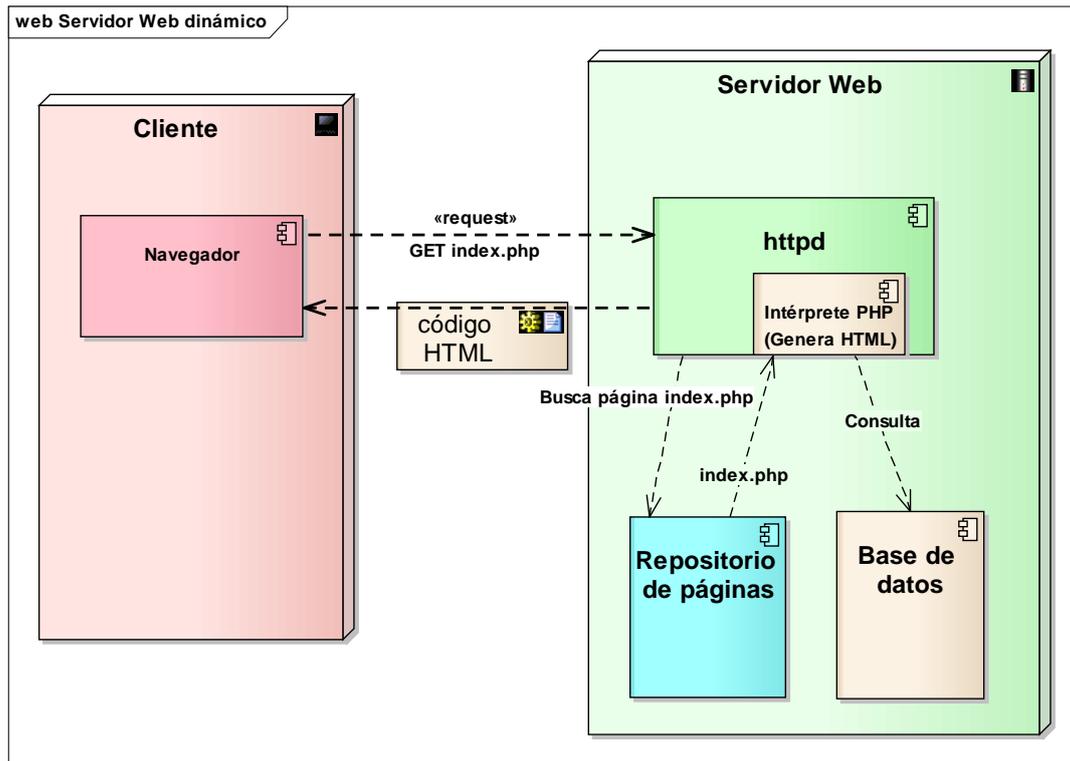
Material bajo licencia Creative Commons



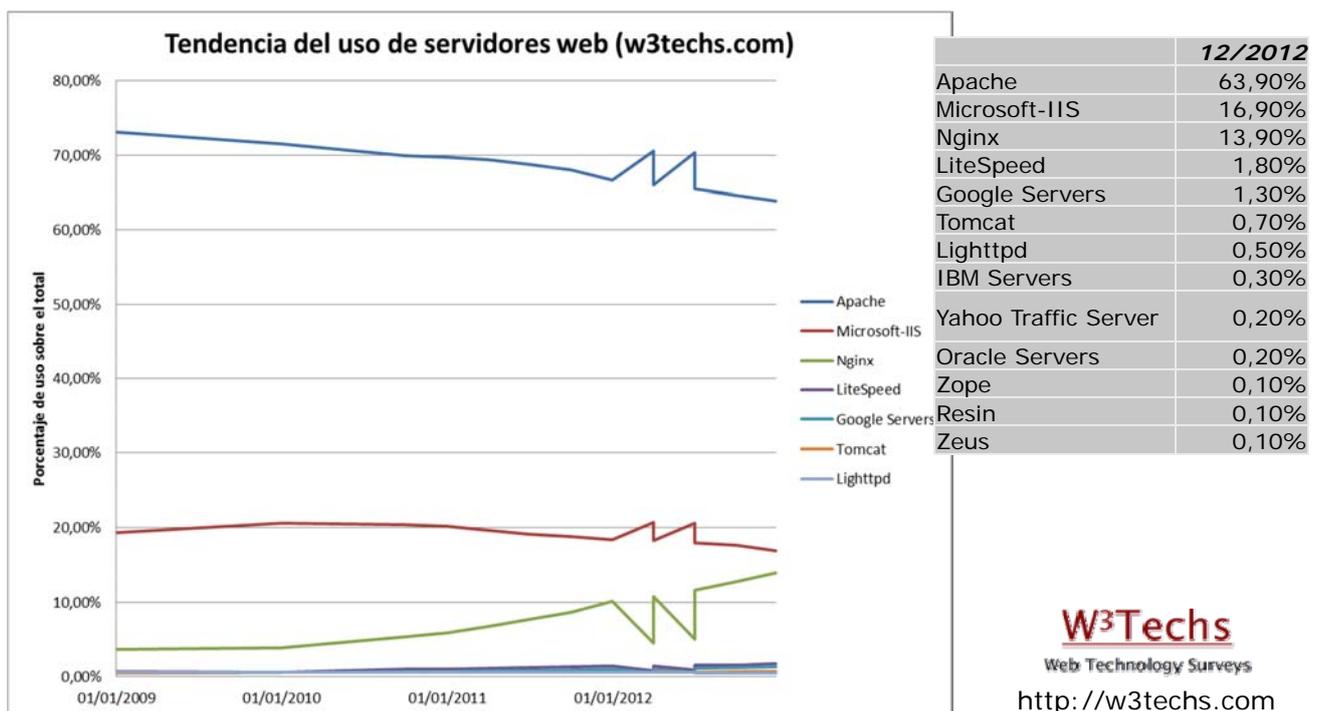
Esquema de funcionamiento de un servidor Web



Esquema de funcionamiento de un servidor Web



Servidores Web



Servidor HTTP Apache

- Servidor HTTP libre, *open source*
 - <http://www.apache.org/>
 - Primera versión en 1995 (como sustituto del *httpd* de NCSA)
 - Apache Software Foundation (1999)
- Ejecutable en varios sistemas operativos
 - Unix, FreeBSD, Linux, Solaris, Novell NetWare, OS X, Microsoft Windows
- Permite múltiples lenguajes de script
 - PHP, Perl, Tcl, Python
- Soporte J2EE
 - Con Tomcat (<http://tomcat.apache.org/>)
- Virtual hosts
 - Un mismo servidor físico para varias IPs y nombres de dominio
- Diseño modular
 - Se pueden crear nuevos módulos con el API de módulos de Apache
- Configuración sencilla
 - Basada en directivas que se editan en ficheros: `httpd.conf`, `access.conf` (en Unix), `.htaccess`, `.htpasswd`

Instalación de Apache - XAMPP

- **Instalación** con XAMPP
 - <http://www.apachefriends.org/es/xampp.html>*
 - Apache con MySQL, PHP y Perl
 - Para Linux, Windows y Mac
 - En Ubuntu: <http://www.ubuntu-guia.com/2011/07/instalar-xampp-en-ubuntu.html>
 - En Windows: hay un autoinstalable
- **Arranque:**
 - En Windows utilizando el panel de control que hay para gestionar los distintos servicios: Inicio->Programas->XAMPP
 - En Linux: Ejecutando `/opt/lampp/lampp start`
- **Prueba:** accediendo en el navegador a `http://localhost` (o lo que es lo mismo `http://127.0.0.1`)
 - Hay varios ejemplos y herramientas accesibles

Advertencia de Seguridad sobre XAMPP

- XAMPP está orientado a desarrollo y no tanto a producción
- **Riesgos de seguridad con XAMPP**
 - El administrador de MySQL no tiene password
 - El demonio de MySQL es accesible desde internet
 - Los usuarios de Mercury y FileZilla (Windows) y ProFTPD (Linux) usan passwords por defecto que son conocidos
 - Por ejemplo en ProFTPD el password "lampp" para el usuario *nobody*
 - PhpMyAdmin es accesible desde internet
 - Los ejemplos se pueden acceder desde la red
 - MySQL y Apache en Linux se ejecutan bajo el mismo usuario (*nobody*)
- Por ello, para producción es mejor instalar Apache y MySQL en un servidor apropiado con una buena configuración de seguridad
 - Ver apéndice sobre configuración de apache2
- Y lo primero es revisar la configuración de seguridad básica
 - Hay ayudas para Linux y Windows (ver la documentación)

Ejercicio: instalación y configuración de XAMPP

- Instala XAMPP en tu computadora
- Arranca los servidores Apache y MySQL
- Prueba a acceder a la página principal (<http://localhost>)
- Aplica las recomendaciones de seguridad que se explican en: <http://robsnotebook.com/xampp-security-hardening>

Directorios y ficheros de trabajo con XAMPP

- En Linux
 - /opt/lampp/htdocs/ Directorio raíz de documentos
 - /opt/lampp/bin/ XAMPP commands home
 - /opt/lampp/bin/mysql MySQL monitor
 - /opt/lampp/etc/httpd.conf Apache configuration file
 - /opt/lampp/etc/my.cnf MySQL configuration file
 - /opt/lampp/etc/php.ini PHP configuration file
 - /opt/lampp/phpmyadmin/config.inc.php phpMyAdmin configuration
- En Windows se pueden encontrar los ficheros en el directorio que se accede con:
Inicio->Apache Friends->XAMPP->XAMPP htdocs folder

Ejercicios

- Localiza cual es el principal directorio para los contenidos web (en principio, por defecto se llama **htdocs**)
- Crea un fichero "prueba.html" en ese directorio y comprueba que se accede correctamente con el URI "http://localhost/prueba.html"
- Crea otro fichero "prueba.php" en ese directorio y prueba que se ejecuta correctamente invocando el URI "http://localhost/prueba.php"
 - Puedes usar un script sencillo como el siguiente:

```
<?php
    echo 'Hola mundo';
?>
```
- Se pueden crear nuevos subdirectorios
 - Coloca el fichero "prueba.html" en un subdirectorio "pruebas" en htdocs y accede con el URI: "http://localhost/pruebas/prueba.html"

Directorios y ficheros de trabajo con XAMPP

- Directorios para contenidos específicos
 - **HTML** (.html .htm): \xampp\htdocs
 - **SSI** (.shtml), **Apache::ASP** **Perl** (.asp): \xampp\htdocs
 - **CGI** (.cgi), **PHP** (.php), **Perl** (.pl): \xampp\htdocs y \xampp\cgi-bin
 - **JSP** (.jsp) y **Servlets** Java (.html): \xampp\tomcat\webapps\java
(requiere **Tomcat**)

Configuración del servidor Apache

- El fichero de configuración de Apache es el **httpd.conf**
 - En el subdirectorio "conf" del directorio de instalación
 - httpd.conf es un fichero ASCII con las **directivas de configuración**
- Tres bloques de directivas (pueden aparecer mezcladas)
 - Parámetros globales
 - Directivas de funcionamiento
 - Hosts virtuales
- El ámbito de las directivas puede ser:
 - Global: el conjunto del servidor web
 - Sección:
 - Un conjunto de directorios o ficheros
 - Un servidor virtual

Configuración del servidor Apache

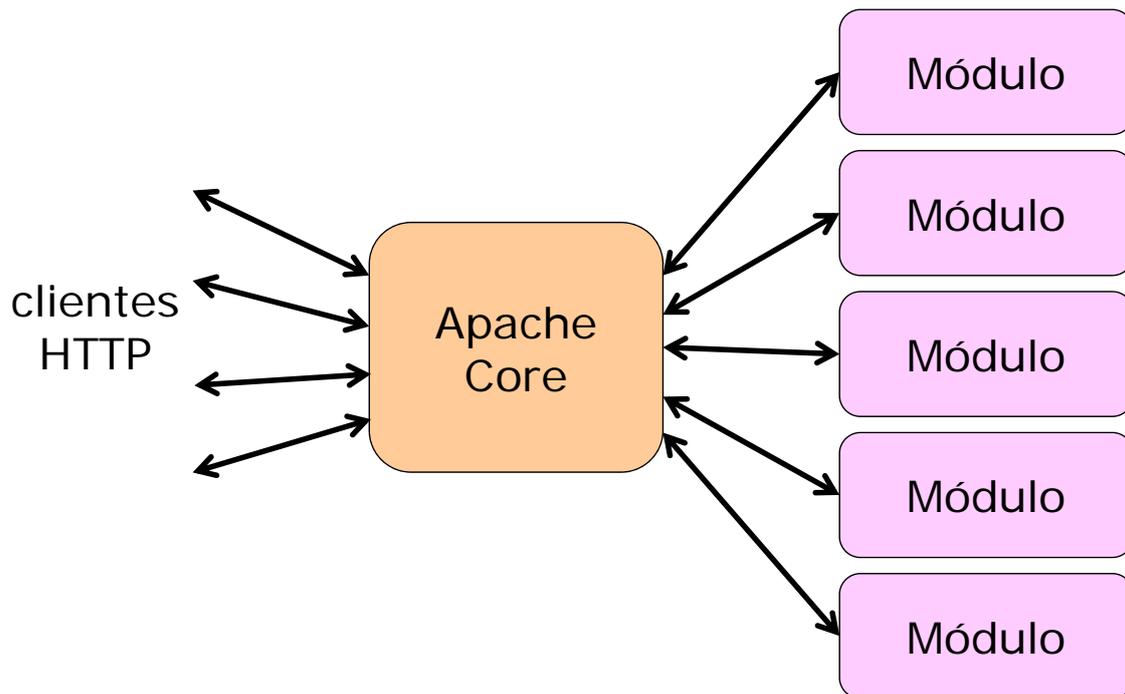
- Especificación de secciones:
 - <Directory>: un directorio y sus subdirectorios
 - <DirectoryMatch>: Un conjunto de directorios según una expresión regular
 - <Files>: Un conjunto de ficheros especificados mediante su nombre
 - <FilesMatch>: Un conjunto de ficheros según una expresión regular
 - <Location>: Ficheros determinados mediante la URL
 - <LocationMatch>: como Location, pero acepta en el nombre expresiones regulares
 - <VirtualHost>: Un host virtual
 - <IfDefine>: sólo se aplica si al arrancar el servidor existe un parámetro concreto definido en la línea de comandos con la opción -D
 - <IfModule>: sólo se aplica si el módulo especificado se encuentra cargado (mediante LoadModule) en el momento de arrancar el servidor

- En caso de conflicto, el orden de precedencia es:
 1. <Directory> y .htaccess
 2. <DirectoryMatch> y <Directory>
 3. <Files> y <FilesMatch>
 4. <Location> y <LocationMatch>
 - <VirtualHost> se aplica siempre después de las directivas generales, por tanto un VirtualHost puede modificar la configuración por defecto

Directivas de configuración básicas (httpd.conf)

- Directivas globales
 - Dirección de instalación del servidor
 - **ServerRoot** "/opt/lampp/apache"
 - Puertos y direcciones IP donde escuchará el servidor
 - **Listen** 80
 - **Listen** 192.168.1.44:8080
 - Carga de módulos
 - **LoadModule** <módulo> <archivo_módulo>
- Directivas principales (a nivel global o de sección)
 - Nombre y puerto del servidor
 - **ServerName** localhost:80
 - Documento raíz
 - **DocumentRoot** "/opt/lampp/htdocs"
 - Email del administrador (aparece con los mensajes de error)
 - **ServerAdmin** postmaster@localhost
- Incluir otros ficheros de configuración
 - **Include** "conf/extra/httpd-xampp.conf"

Arquitectura del servidor Apache



Módulos Apache

- Módulo base
 - Core: Funcionalidad básica (asignación de peticiones, gestión de conexiones)
- Módulos multi-proceso
 - mpm_common, perchild, prefork, worker: Acepta peticiones por los puertos y las dirige a los hijos responsables
- Módulos adicionales
 - mod_access: control de acceso
 - mod_alias: redirección de URLs
 - mod_rewrite - reescritura de direcciones (transforma páginas dinámicas como php en páginas estáticas html)
 - mod_auth_ldap – Autenticación de usuarios con servidor LDAP
 - mod_perl - Páginas dinámicas en Perl
 - mod_php - Páginas dinámicas en PHP
 - mod_python - Páginas dinámicas en Python
 - mod_ruby - Páginas dinámicas en Ruby
 - mod_ssl - Comunicaciones Seguras vía TLS
 - mod_security - Filtrado a nivel de aplicación, para seguridad

Directivas de configuración de módulos (httpd.conf)

- Los módulos específicos que se van a incluir se definen en la configuración del servidor (httpd.conf) con la directiva LoadModule

- Se pueden definir configuraciones en función de los módulos cargados al arrancar el servidor:

```
<IfModule modulo>
    Configuraciones específicas si está el módulo
</IfModule>
<IfModule !modulo>
    Configuraciones específicas si no está el módulo
</IfModule>
# DirectoryIndex: indica el fichero que servirá Apache si no se
# especifica uno concreto (el orden determina la prioridad)
<IfModule !autoindex_module>
    <IfModule dir_module>
        DirectoryIndex index.php index.html index.htm
    </IfModule>
</IfModule>
```

Directivas de control de acceso

- Se pueden definir dentro de un fichero de configuración o de un fichero de control de acceso en un directorio (*.htaccess*)
 - *.htaccess* se recomienda solo cuando no se tiene acceso al fichero de configuración del servidor
- Determinan quién tiene derecho a acceder al sitio o a una sección del sitio, y quién no
 - Generalmente se especifican direcciones IP o nombres de dominio
 - También se puede especificar fecha y hora
 - **all**: todos los clientes
 - Allow: especifica quién tiene acceso a la sección del servidor
 - **Allow From all**
 - **Allow From** 192.168 – todos los hosts en 192.168.x.x tienen acceso
 - Deny: quién no tiene acceso
 - **Deny From** piratas.com
- Para afinar el funcionamiento de estas directivas, se puede especificar en qué orden se ejecutan:
 - **Order** Allow,Deny
 - **Order** Deny,Allow

Directivas de control de acceso

<pre><Directory "/www"> Order Allow,Deny Deny from all Allow from all </Directory></pre>	⇒ Nadie puede entrar porque Deny se ejecuta al final
<pre><Directory "/www"> Order Deny, Allow Deny from all Allow from all </Directory></pre>	⇒ Cualquiera puede entrar porque Allow se ejecuta al final
<pre><Directory "/www"> Order Deny,Allow Deny from all Allow from ejemplo.com </Directory></pre>	⇒ Solo pueden acceder los que sean de un dominio xxx.ejemplo.com
<pre><Directory "/www"> Order Allow,Deny Allow from all Deny from ejemplo.com </Directory></pre>	⇒ Pueden acceder todos los que no vengan de un dominio xxx.ejemplo.com

Ejercicio de directivas de control de acceso

- ¿Cuál será el comportamiento en estos casos?

```
<Directory "/web">
  Order Allow,Deny
</Directory>
```

.htaccess

```
Order Allow,Deny
Allow From all
Deny From 192.168.
```

httpd.conf

```
<Directory "/particular/protegido"
  Order Allow,Deny
  Allow From all
  Deny From 192.168.
</Directory>

<FilesMatch "*.pdf$"
  Order Allow,Deny
  Allow From all
</FilesMatch>
```

Apache: Virtual hosts

- Es posible alojar varios sitios Web (múltiples nombres de dominio y direcciones IP) en un solo servidor Apache
 - `http://www.miweb.com`
 - `http://miblog.miweb.com`
 - `http://www.otraweb.com`
- Pasos para crear hosts virtuales:
 1. Definir los virtual hosts en el fichero **host** (en producción habrá que contratar un dominio y publicarlo en DNS)
 - En el directorio `/etc` o en `c:/Windows/System32/Drivers/etc/`

```
#localhost
127.0.0.1 localhost
#Miweb
127.0.0.1 www.miweb.com
#Mi blog
127.0.01 miblog.miweb.com
#Otraweb
127.0.0.1 www.otraweb.com
```

Apache: Virtual hosts

2. Configurar los hosts virtuales en el fichero `httpd.conf` de Apache
 - Normalmente la información de los virtual hosts se pone en un fichero aparte que se indica en el `httpd.conf` con la directiva:
`Include "conf/extra/httpd-vhosts.conf"`
3. Editar el fichero de configuración de hosts virtuales:
 - Permitir que se usen nombres para virtual hosts
`NameVirtualHost *:80`
 - Definir cada uno de los virtual hosts:
`<VirtualHost [direcciónIP:Puerto]>`
`ServerAdmin`
`DocumentRoot` - dónde se aloja la página web
`ServerName` - el nombre del host virtual
`ServerAlias`
`ErrorLog`
`CustomLog`
`</VirtualHost>`
4. Parar y arrancar de nuevo el servidor Apache para que la nueva configuración tenga efecto

Apache: Virtual hosts

```
#localhost
<VirtualHost *:80>
    DocumentRoot "C:/www"
    ServerName localhost:80
</VirtualHost>

#Miweb
<VirtualHost *:80>
    DocumentRoot "C:/www/miweb"
    ServerName www.miweb.com
    ServerAlias miweb.com
</VirtualHost>

#Mi blog
<VirtualHost *:80>
    ServerAdmin miemail@miweb.com
    DocumentRoot "C:/www/miblog"
    ServerName miblog.miweb.com
</VirtualHost>

#Otra web
<VirtualHost *:80>
    ServerAdmin webmaster@otraweb.com
    DocumentRoot "C:/www/otraweb"
    ServerName www.otraweb.com
</VirtualHost>
```

Ejercicios: hosts virtuales

- Configura dos host virtuales basados en nombre: `www.prueba.com` y `www2.prueba.com`
 - Configura los host virtuales con distintos *DocumentRoot* de manera que puedas verificar que se han configurado de manera adecuada
 - Para poder realizar las pruebas, ya que los host virtuales requieren la resolución de un nombre a una IP, haría falta el acceso a un servidor DNS. Sin embargo, para las pruebas puedes hacer uso del archivo **host** que es controlado por el sistema operativo y que permite asignar IP a un nombre de dominio, de manera que se realiza una resolución de nombre local
 - NOTA: No olvides eliminar las entradas una vez concluido el ejercicio
- Ayudas:
 - <http://httpd.apache.org/docs/2.4/vhosts/>
 - <http://httpd.apache.org/docs/2.4/vhosts/name-based.html>

Logs (httpd.conf)

- Fichero de registro de errores
 - **ErrorLog** "logs/error.log"
- Fichero de registro de accesos
 - **TransferLog** "logs/transfer.log"
- Fichero de registro particularizado
 - **CustomLog** "logs/access.log" nombre_formato
 - Formato del log:
 - **LogFormat** "%h %l %u %t \"%r\" %>s %b" nombre_formato

Gestión de errores

- Apache genera los errores estándar de HTTP
 - 301 Moved Permanently
 - 401 Unauthorized
 - 403 Forbidden
 - 404 Not found
 - 500 Internal Server Error
- Se pueden particularizar declarando el fichero correspondiente
 - **ErrorDocument** <código de error> <fichero con página de error>
ErrorDocument 401 /error401.html
ErrorDocument 500 /error500.html

Bibliografía

- Apache ofrece la distribución oficial del servidor web Apache, su código fuente y toda la documentación correspondiente
 - <http://httpd.apache.org/>
 - <http://httpd.apache.org/docs/2.2/>
- XAMPP
 - <http://www.apachefriends.org/>
- Ubuntu (oficial)
 - <https://help.ubuntu.com/12.10/serverguide/httpd.html>
- Inteco CERT (2012). *Guía básica para la securización del servidor web Apache*
 - http://cert.inteco.es/extfrontinteco/img/File/intecocert/EstudiosInformes/inteco_cert_guia_seguridad_apache.pdf
- Pedro Pablo Fábrega Martínez. Configuración y administración de Apache. Una guía básica
 - <http://dns.bdat.net/documentos/apache/book1.html>
- Rich Bowen, Ken Coar (2007). *Apache Cookbook, 2nd Edition. Solutions and Examples for Apache Administration*. O'Reilly Media

Apéndice: Configuración de Apache2 en Ubuntu

- Instalación de lamp
 - Para Ubuntu 12.10:
<https://help.ubuntu.com/12.10/serverguide/web-servers.html>
- Ubuntu server
 - Sobre Ubuntu desktop se pueden instalar los paquetes correspondientes para dar funcionalidad de servidor
 - Instalación de la tarea lamp-server
 - Directamente

```
$ sudo apt-get install lamp-server^
```
 - Con tasksel, una utilidad para instalar tareas del Ubuntu server
<https://help.ubuntu.com/community/Tasksel>

```
$ sudo apt-get install tasksel
```

```
$ sudo tasksel install lamp-server
```

 - Se puede hacer invocando solo *tasksel* que muestra un menú con las tareas disponibles

```
$ tasksel --list-tasks
```

 - muestra los paquetes instalados

Arranque y parada del servidor

- Arranque
\$ sudo /etc/init.d/apache2 start
- Rearranque
\$ sudo /etc/init.d/apache2 restart
- Parada
\$ sudo /etc/init.d/apache2 stop

- Una vez instalado, si se prueba la página `http://localhost` aparecerá una página con el mensaje:
It works!

- También se puede arrancar con la utilidad **apache2ctl**
 - También ofrece información sobre la configuración del servidor:
 - `apache2ctl -t` : Comprueba la sintaxis del fichero de configuración
 - `apache2ctl -M` : Lista los módulos cargados
 - `apache2ctl -S` : Lista los sitios virtuales
 - `apache2ctl -V` : Lista las opciones de compilación

```
$ sudo apache2ctl -V
Server version: Apache/2.2.22 (Ubuntu)
Server built:   Mar  8 2013 15:53:14
Server's Module Magic Number: 20051115:30
Server loaded: APR 1.4.6, APR-Util 1.4.1
Compiled using: APR 1.4.6, APR-Util 1.4.1
Architecture: 32-bit
Server MPM:     Prefork
    threaded:   no
    forked:     yes (variable process count)
Server compiled with....
-D APACHE_MPM_DIR="server/mpm/prefork"
-D APR_HAS_SENDFILE
-D APR_HAS_MMAP
-D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
-D APR_USE_SYSVSEM_SERIALIZE
-D APR_USE_PTHREAD_SERIALIZE
-D APR_HAS_OTHER_CHILD
-D AP_HAVE_RELIABLE_PIPED_LOGS
-D DYNAMIC_MODULE_LIMIT=128
-D HTTPD_ROOT="/etc/apache2"
-D SUEXEC_BIN="/usr/lib/apache2/suexec"
-D DEFAULT_PIDLOG="/var/run/apache2.pid"
-D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
-D DEFAULT_LOCKFILE="/var/run/apache2/accept.lock"
-D DEFAULT_ERRORLOG="logs/error_log"
-D AP_TYPES_CONFIG_FILE="mime.types"
-D SERVER_CONFIG_FILE="apache2.conf"
```

Apache2 – directorios y ficheros de configuración

- Dentro de `/etc/apache2/`:
 - `apache2.conf`: archivo raíz
 - `mods-enabled/*.load` y `mods-enabled/*.conf`: carga y configuración de los módulos de Apache
 - `httpd.conf`: ya no se usa porque está repartido en los demás
 - Actualmente vacío. Puede usarse para poner algunas opciones de configuración específicas que afectan globalmente a Apache2
 - `envvars`: variables de entorno para Apache2
 - `ports.conf`: define en qué puertos escuchará Apache
 - `conf.d/`: directorio que contiene archivos de configuración para cada funcionalidad de apache (`charset`, `php`, `security`, etc.)
 - `sites-available/`: directorio con ficheros de configuración de los Apache2 Virtual Hosts
 - `sites-enabled/`: enlaces simbólicos a los ficheros correspondientes a los virtual hosts en `/etc/apache2/sites-available` que están activos

Configuración básica

- Apache2 viene preparado para manejar hosts virtuales
 - Por defecto hay uno: `/etc/apache2/sites-available/default`
- Para crear nuevos sitios virtuales, copiar el que hay por defecto y editarlo con la configuración del nuevo sitio:

```
$ sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/nuevositio
```
- Para cada sitio editar lo siguiente:
 - `ServerAdmin`: email del administrador
 - `Listen`: Puerto donde escucha (en nuestro caso no hace falta especificar una dirección IP porque solo hay una fija)
 - Normalmente se cambia en `/etc/apache2/ports.conf`
 - `ServerName`: dominio al que atiende. Por defecto no está indicado y se atiende a cualquier petición que le llegue distinta de la que estén indicadas para otros virtual hosts.
 - `ServerAlias` `*.dominio.com` permite que atienda `www.dominio.com`, `dominio.com`, etc.
 - `DocumentRoot`: indica dónde están los documentos que sirve. Por defecto `/var/www`

Configuración de virtual hosts

- Utilidad a2ensite para activar un host virtual:

```
$ sudo a2ensite mynewsite  
$ sudo service apache2 restart
```

- Para desactivar un host virtual:

```
$ sudo a2dissite mynewsite  
$ sudo service apache2 restart
```

Configuración por defecto

- DirectoryIndex
 - En /etc/apache2/mods-available/dir.conf
 - index.html index.cgi index.pl index.php index.xhtml index.htm
 - ErrorDocument
 - /etc/apache2/conf.d/localized-error-pages
 - Transfer log
 - /var/log/apache2/access.log
 - Se puede cambiar con la directiva CustomLog para cada virtual host
 - Options para cada directorio
 - ExecCGI: para permitir la ejecución de scripts CGI
 - Includes: para permitir a un fichero HTML incluir SSI (*server side includes*)
 - Indexes: para mostrar los contenidos del directorio si no hay un DirectoryIndex (p.ej. index.html) en el directorio
 - Por seguridad conviene deshabilitarlo
 - SymLinksIfOwnerMatch: solo se siguen los enlaces simbólicos si el objetivo tiene el mismo owner que el enlace
- Options -FollowSymLinks +SymLinksIfOwnerMatch

Módulos dinámicos en Apache2

- Apache2 en Ubuntu viene compilado para permitir la carga dinámica de módulos
- Las directivas pueden condicionarse a la presencia de un módulo con `<IfModule>`
- Para instalar un módulo (p.ej. el módulo de autenticación de MySQL):

```
$ sudo apt-get install libapache2-mod-auth-mysql
```
- Módulos disponibles en `/etc/apache2/mods-available`
- Utilidad `a2enmod` para activar un módulo

```
$ sudo a2enmod auth_mysql  
$ sudo service apache2 restart
```
- Utilidad `a2dismod` para desactivar un módulo

```
$ sudo a2dismod auth_mysql  
$ sudo service apache2 restart
```

Configuración HTTPS

- Módulo `mod_ssl` para encriptar las comunicaciones
- Para activarlo:

```
$ sudo a2enmod ssl
```
- Configuración de HTTPS por defecto en `/etc/apache2/sites-available/default-ssl`

```
$ sudo a2ensite default-ssl
```
- Para HTTPS hacen falta
 - Un certificado
 - Un fichero con la clave
 - Se generan con el paquete `ssl-cert`
 - Conviene usar uno específico para el servidor:
<https://help.ubuntu.com/12.10/serverguide/certificates-and-security.html>
 - Por defecto se ponen en los directorios `/etc/ssl/certs` y `/etc/ssl/private`
- Una vez configurado hay que volver a arrancar el servidor

```
$ sudo service apache2 restart
```
- Se accederá a las páginas con **https**://dominio/directorio