
Unidad 9-a: Servidores WEB

**Curso de Introducción a la administración de
servidores GNU/Linux**

**Centro de Formación Permanente
Universidad de Sevilla**

Contenidos

1. Introducción a la Web
2. Apache 2
3. Persistencia de Procesos
4. Configuración y puesta en marcha
5. Soporte SSL
6. Servidores virtuales
7. Autenticación
8. Scripting
9. Aplicaciones Web

Contenidos

1. Introducción a la Web

1. Un poco de historia
2. Componentes de la Web
3. El protocolo HTTP
4. El mercado de servidores Web

Un poco de historia

- 1989-1990 – Tim Berners-Lee inventa la WWW en el CERN
 - Un nuevo modo de transferir información
- Protocolo de transferencia Cliente/Servidor
 - Comunicación a nivel de aplicación
 - Sobre la base de la capa de red estándar
- Lenguaje de marcado de hipertexto
 - El culpable de la explosión de la Web
 - No lo inventó Berners-Lee
 - Requiere un cliente que lo renderice

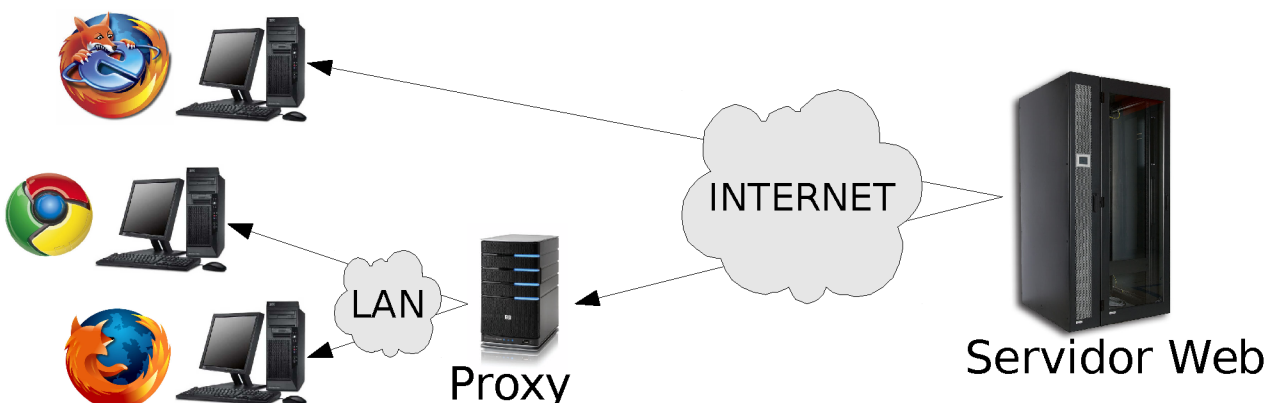


Un poco de historia (2)

- 1994 – Mark Andreessen desarrolla Mosaic en la NCSA
 - Primer navegador gráfico
 - Primera “killer-app” de Internet
 - Gratuito, se convirtió en Netscape
- 1995 – El tráfico web se hace dominante
 - Crecimiento exponencial
 - Creación del consorcio W3C
 - Las empresas crean su infraestructura web
 - Inicios del comercio electrónico

Componentes

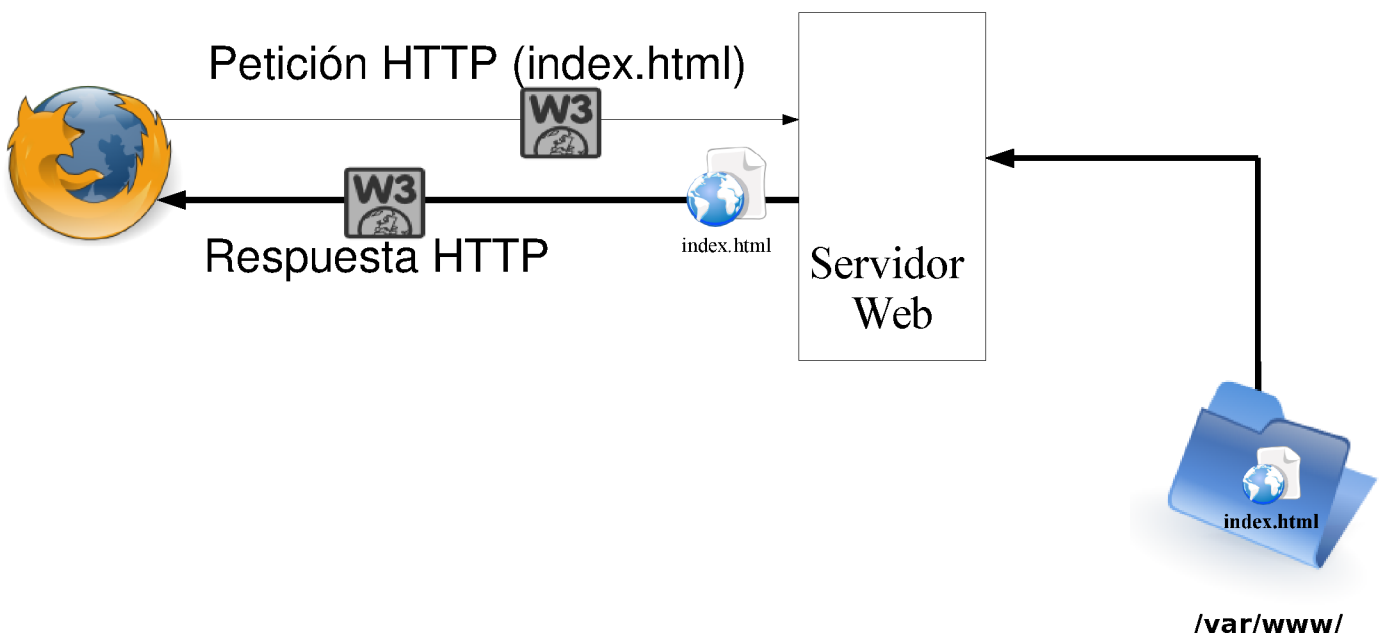
- Elementos estructurales de la web
 - Interconexión: Internet, Redes locales, etc...
 - Clientes o navegadores
 - Servidores
 - Intermediarios: Proxys, cachés, etc...



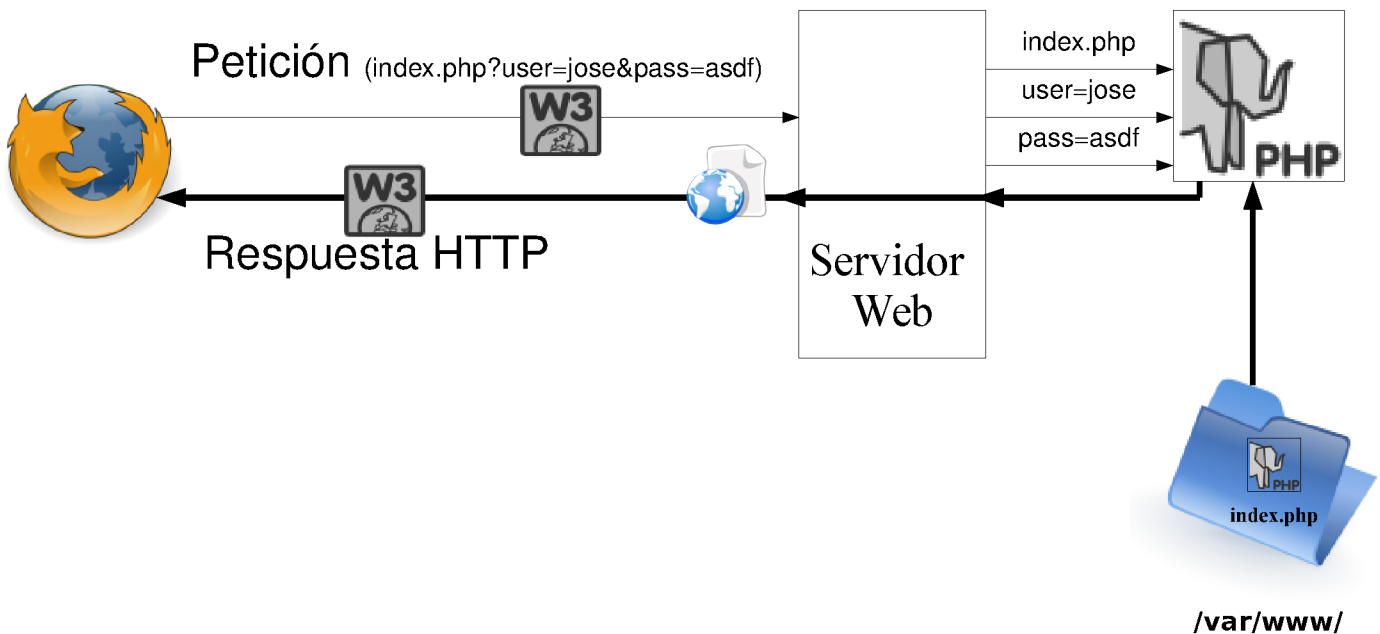
El Protocolo HTTP

- Protocol for client/server communication
 - Es el corazón de la Web
 - Mecanismo simple de petición/respuesta
 - Stateless
 - URIS como mecanismo de nombrado
- Versiones
 - 09/1.0 – Berners-Lee’s original
 - RFC 1945
 - Usado masivamente antes de HTTP/1.1
 - 1.1 – Mejoras de prestaciones, cacheo, compresión, etc...
 - RFC 2068

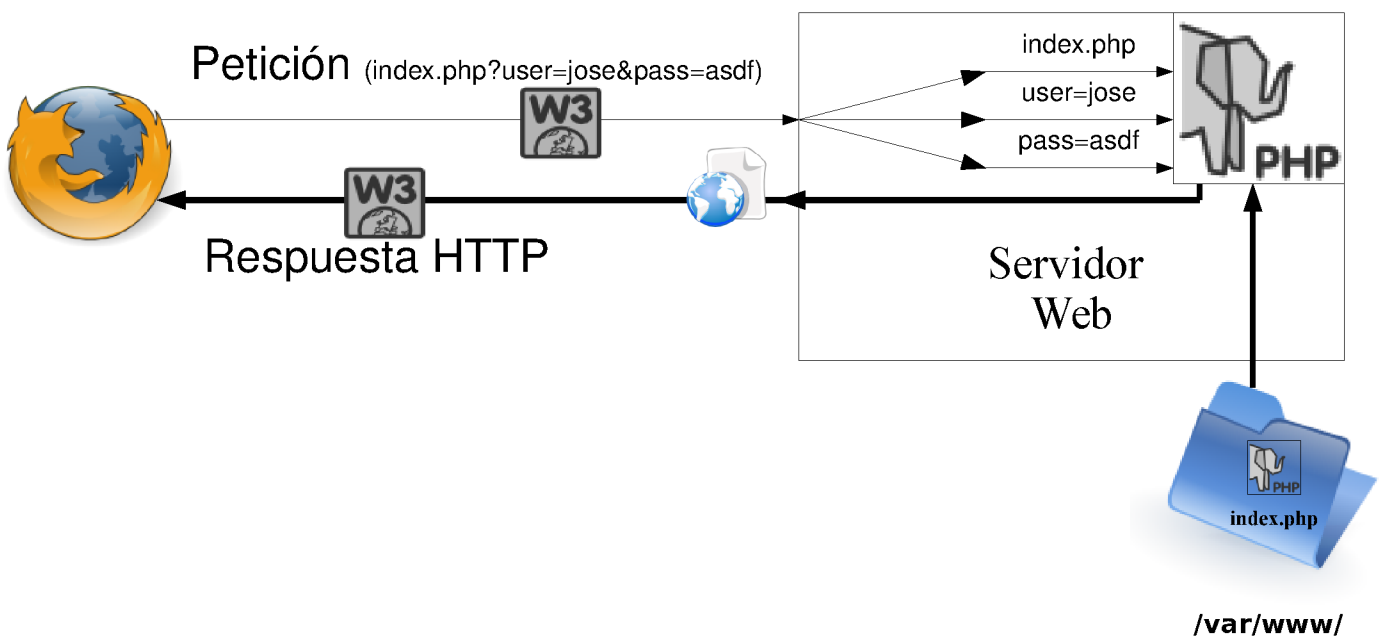
El Protocolo HTTP (2) Transacción estática



El Protocolo HTTP (3) Transacción dinámica (CGI)



El Protocolo HTTP (4) Transacción dinámica (ModPHP)



El Protocolo HTTP (5)

Ejemplo de Conexión a bajo nivel

- Vamos a conectarnos a un servidor web...
 - El servidor escucha el puerto 80

- `telnet teclix.dte.us.es 80`

- Realizaremos amablemente una petición.

```
GET /enlaces HTTP/1.1
Host: www.dte.us.es
(salto de línea)
```

- Un buscador web a través de Telnet
 - Recurso de `www.dte.us.es`:

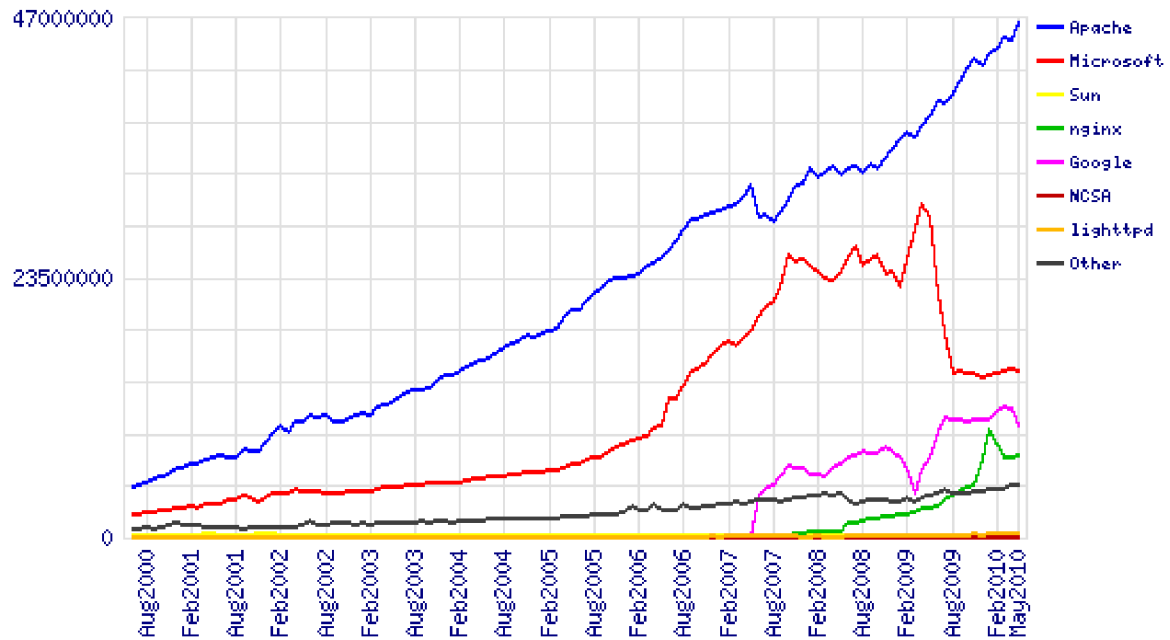
- `/search?SearchableText=linux`

El Mercado de Servidores WWW

- El más utilizado es Apache pero hay multitud de servidores WEB
 - `boa`: Utilizado en sistemas empujados: Routers, WebCams.
 - `lighttpd`: Usado en YouTube
 - `nanoweb`
 - `webfs`: Sólo contenido estático
 - `cherokee`: the fastest free web server out there!

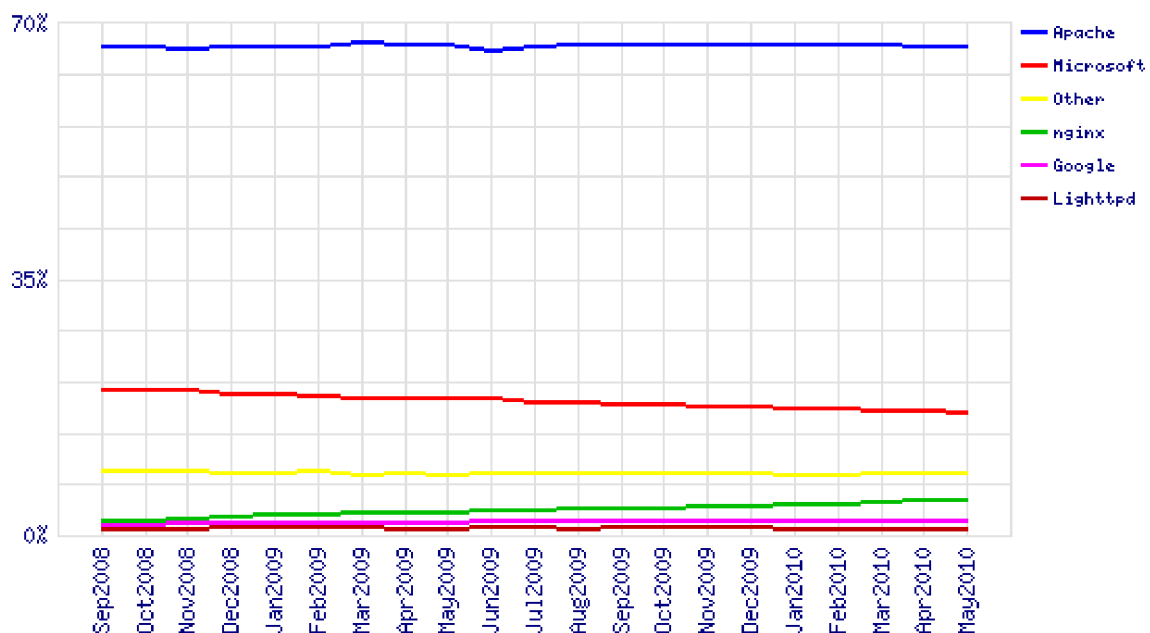
El Mercado de Servidores WWW (2)

Totals for Active Servers Across All Domains
June 2000 - May 2010



El Mercado de Servidores WWW (3)

Market Share for Top Servers Across the Million Busiest Sites
September 2008 - May 2010



Apache2

- Instalación: `apt-get install apache2`
- Página de documentación: www.apache.org
- Configuración dispersa en ficheros:
 - `/etc/apache2`
- Funcionalidad en diferentes módulos enlazados dinámicamente:
 - `libapache2-mod-*`

Procesos Persistentes

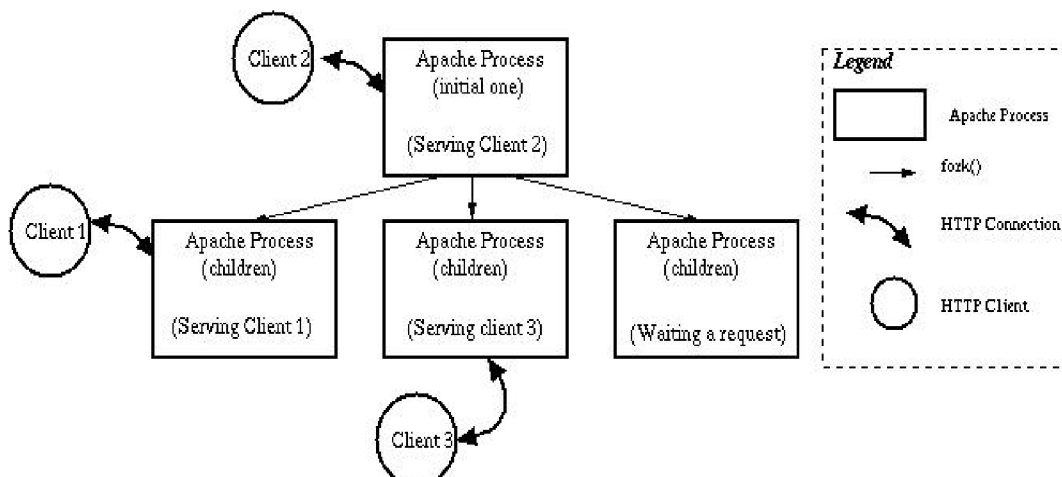
- La 'Persistencia de Procesos' es un concepto que explica el mecanismo de concurrencia en Apache
- Para atender gran cantidad de peticiones, los servidores crean nuevas instancias del proceso mediante la llamada `fork()`.
- Sin embargo la sobrecarga de la creación de gran cantidad de nuevos procesos hijos simplemente satura el sistema

Procesos Persistentes (2)

- Para solucionar esto Apache usa 'Persistencia de Procesos'
- Genera un determinado número de hijos desde el principio.
- Apache controla dinámicamente el número de procesos existentes en función de la carga de trabajo.

Procesos Persistentes (3)

- Diferentes versiones:
 - prefork: Varios procesos, mucha memoria, las peticiones no tienen efectos laterales
 - worker: Un proceso con muchos hilos, rápido, poco consumo de memoria



Contenidos

4. Configuración y puesta en marcha

1. Ficheros de configuración
2. Comandos de control
3. Módulos
4. Directivas

Ficheros de configuración

- **apache2.conf**: Fichero principal de configuración con opciones comunes:
 - Limitación de recursos
 - Usuario y grupo con el que se ejecuta el servidor

Ficheros de configuración

- **conf.d/**: Todos los archivos de este directorio son incluidos en el fichero `apache2.conf` mediante las directivas `include`
- **httpd.conf**: Fichero vacío
- **magic**: Configuración del módulo `mod_mime_magic`. Se encarga de transferir ficheros de determinados tipos.
- **ports.conf**: Directivas para indicar los puertos y las IPS en las que escuchar

Ficheros de configuración

- **mods-available/**: Directorio con ficheros para cargar módulos de apache. Por cada módulo hay dos ficheros:
 - **.load**: Directivas de apache encargadas de cargar el módulo correspondiente
 - **.conf**: fichero de configuración del módulo de apache
- **mods-enabled/**: Enlaces simbólicos a los módulos que se quieren activar en apache (ficheros `.load`)

Ficheros de configuración

- **sites-available/**: contiene configuraciones de diferentes sitios web. Se recomienda poner cada uno en un fichero.
- **sites-enabled/**: Funcionamiento similar a mods-enabled. Enlaces simbólicos a los sitios que se desean activar en el directorio sites-available

Comandos de control

- Parada inicio del servidor:
service apache2 start|stop|status|restart|reload|...
- apache2ctl (de apache-2.2-common):
 - apache2ctl configtest
 - apache2ctl status
- apache2-ssl-certificate: Crear certificado SSL. Utilizado en el módulo SSL
 - sudo mkdir /etc/apache2/ssl
 - sudo /usr/sbin/make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/ssl/apache.pem
- Comprobar si funciona: netstat -ltnp

Módulos

- Añaden funcionalidad al servidor WEB
- Están como paquetes: libapache2-mod-*
- Utilizaremos:
 - **cgi**: Ejecución de programas
 - **php4 o php5**: Ejecución de scripts php
 - **ssl**: Soporte SSL
- Cuando un módulo está activo, se pueden incluir directivas del módulo en los ficheros de configuración.

Directivas: formato

- Como ejemplo: sites-enabled/default
- Se definen ámbitos en las que se aplican un conjunto de directivas utilizando un formato parecido la HTML

```
<Directory /var/www/default>  
    Directivas  
</Directory>
```

Directivas: ámbitos

- Las directivas se aplican en el ámbito en el que se encuentran.
- Existe un conjunto de directivas por cada modulo cargado con apache.
- Es necesario consultar en la documentación para comprobar en que ámbitos es posible aplicar cada directiva

Directivas

- Globales para todos los sitios activos:
 - User: Usuario con que se ejecuta el servidor
(*fichero: apache2.conf*)
 - Group: Grupo con el que se ejecuta el servidor
(*fichero: apache2.conf*)
 - Listen: Puertos en los que escuchar
(*fichero: ports.conf*)
 - AddDefaultCharset: Cuidado con la página de códigos del servidor: **UTF-8, ISO-8859-1, ISO-8859-15, ...** (*fichero: conf.d/charset*)

Directivas

- Locales para cada sitio (sites-available/*)
 - ServerAdmin: dirección de correo del administrador.
 - ServerName: url a la que debe responder el servidor.
 - DocumentRoot: Directorio donde se aloja el sitio web.
 - CustomLog: Bitácora de la actividad en el sitio
 - ErrorLog: Bitácora con los errores

Directivas

- Utilizadas habitualmente
 - **Alias**: Redirige una petición a un directorio concreto
 - **ScriptAlias**: como Alias pero para CGIs
 - Control de acceso: **allow from / deny from / order**
 - **AllowOverride**: Permite crear archivos **.htaccess** con directivas apache en diferentes carpetas. Se cargarán de forma dinámica.
 - **DirectoryIndex**: Página por defecto

Soporte SSL

- Activar el módulo SSL mediante los enlaces simbólicos de mods-enabled
- Por defecto SSL usa el puerto 443
 - Directiva: Listen 443
- Crea el certificado
 - `sudo mkdir /etc/apache2/ssl`
 - `sudo /usr/sbin/make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/ssl/apache.pem`
(Codificado MIME base 64)
 - Si se necesitan varios hay que renombrarlo

Soporte SSL (2)

- Activar SSL en algún sitio web usando las directivas del módulo SSL:
 - `SSLEngine on`
 - `SSLCertificateFile <fichero.pem>`
 - `SSLCertificateKeyFile <fichero.pem>`
- **No es posible utilizar diferentes certificados según el sitio web que se sirva: Una IP <=> Un certificado.**
 - La negociación del protocolo SSL se realiza antes que la petición WEB por tanto no es una limitación de Apache
- Cambiar el puerto del sitio web:
<VirtualHost *:443>

Servidores virtuales

- Alojamiento virtual consiste en alojar varios sitios web en una misma máquina.
- Se pueden basar en IPs o en nombres.
- Directiva **NameVirtualHost**, consigue que se active el alojamiento virtual basado en nombre en la IP y puerto especificado:
 - NameVirtualHost *:80
 - NameVirtualHost 192.168.0.2:8080

Servidores virtuales (2)

- Para responder cada petición basada en un nombre es necesario:
 - Bloques `<virtualhost x.x.x.x:x>...</virtualhost>`
- En el interior del bloque usar las directivas:
 - ServerName: Nombre al que responder
 - ServerAlias: Nombres alternativos
 - DocumentRoot: Raiz del WEB
 - CustomLog
 - Errorlog

Servidores virtuales (3)

```
<VirtualHost *:80>
    ServerAdmin administrador@miservidor.com
    ServerName www.miservidor.com
    DocumentRoot /var/www/www.miservidor.com/
    DirectoryIndex index.html index.htm index.php
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>

    ErrorLog /var/log/apache2/www.miservidor.com-error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    CustomLog /var/log/apache2/www.miservidor.com-access.log combined
    ServerSignature On
</VirtualHost>
```

Autenticación

- Para crear un archivo de usuarios la primera vez:
 - sudo htpasswd -c <archivo > <usuario>
- Para añadir nuevos usuarios.
 - sudo htpasswd <archivo > <usuario>
- Directivas de autenticación:
 - AuthName "<mensaje de petición de clave>"
 - AuthType Basic
 - AuthUserFile <archivo passwd>
 - require valid-user

Scripting

- PHP 5: motor de scripting soportado por apache como CGI o mediante el módulo modPHP.
 - `sudo apt-get install libapache2-mod-php5`
- Perl: lenguaje de programación muy popular para la programación de scripts. Hoy en desuso debido a la existencia de nuevas alternativas y al estancamiento de versiones.
- Python: Alternativa con buen soporte para Apache y creciente popularidad.
- ...

Log y análisis

- Paquetes adicionales a Apache
 - WEBALIZER: Estadísticas de visitas simples
 - AWSTATS: Estadísticas mejoradas
 - IMP: Lector de correo WEB
 - GroupWare: Administrador de grupos de trabajo

Aplicaciones Web

- Existen multitud de aplicaciones web disponibles directamente desde los repositorios:
- Suelen requerir soporte de bases de datos
 - MediaWiki
 - Drupal
 - WordPress
 - Joomla
 - ...
- “**phpmyadmin**” para administrar desde la web una base de datos mySQL Server.

Aplicaciones Web

- Algunas con posibilidad de instalación desde los repositorios:
 - `sudo apt-get install drupal7`
 - Añadir los cambios sugeridos en `/etc/drupal/7/apache.conf` a nuestro virtual host.
 - Reiniciar Apache
 - Acceder <http://localhost/drupal7/install.php>
 - Normalmente poco actualizadas! Y por tanto inseguras!

Aplicaciones Web

- La alternativa es su instalación manual
 - Ir a www.drupal.org
 - Descargar la versión a instalar
 - Instalar y configurar la base de datos (MySQL, PostgreSQL): “mysql-server”
 - Iniciar el script de instalación “install.php” descargado.
 - Borrar el script de instalación