El servidor web Apache

Introducció:

El servidor Apache és un eina destinada a establir el model client-servidor en la que el client serà un navegador web i el servidor serà una màquina en la que s'estigui executant Apache. Apache destaca pel seu rendiment i estabilitat.

Apache permetrà allotjar en una màquina pàgines HTML en directoris concrets i, a més a més, poder treballar amb tots els tipus de fitxers relacionats amb les pàgines web allotjades (programes CGI, XML,...). Haurà de permetre gestionar la comunicació en els dos sentits que es produirà entre el servidor i els clients (navegadors). També serà molt important que aquest servidor tingui una gran capacitat d'adaptació a tots els nous formats que vagin apareixent.

Tot el que hem descrit anteriorment haurà de fer-se tenint sempre e compte la seguretat. Caldrà possibilitar protocols segurs (https) i oferir eines d'encriptació (eines SSL).

Aquest servidor és de programari lliure i acapara una quota de mercat de més del 60%.
Les seves principals característiques són:

1. És multiplataforma, per tant el podríem utilitzar tant en màquines sota Linux com amb màquines windows.
2. Segueix el protocol HTTP/1.1
3. És modular. Té mòduls diferents per adaptar-se a diferents entorns. A més, té una API de programació de mòduls
4. Extensible. Al ser modular, es pot anar ampliant, com per exemple, les ampliacions per suportar PHP.
5. Serveix documents a Internet i a intranets seguint el protocol HTTP

Apache va néixer al 1995 quan un grup d'informàtics van decidir realitzar uns parxes sobre un altre servidor web (NSCA), que implementava httpd.

Les versions més utilitzades són la 1.3 i la 2.0, que destaca per la millora en el tractament de màquines no Linux i en temes de multiprocessador. Amb Red Hat es distribueix Apache 2.0.

Per consultar la instal·lació en entorn Windows, ho tenim ben explicat a http://httpd.apache.org/docs/2.2/platform/windows.html , això sí, en anglès.

El protocol HTTP (Hypertext transfer protocol):

Aquest protocol és el motor d'internet. És un protocol a nivell d'aplicació. L'esquema bàsic de funcionament segueix el diàleg client/servidor establert pel protocol TCP.

En la versió HTTP1.0 el client només disposava de tres comandes cap al servidor:

- GET: demanar un pàgina al servidor
- HEAD: demanar la capçalera d'una pàgina per saber la seva darrera actualització, tamany,.... Ho utilitzen els servidors de caché de pàgines
- POST: enviar dades a una URL. Enviar dades al servidor

Cada crida d'aquestes implicarà una connexió amb el servidor i, posteriorment, la desconnexió. Per tant, el servidor no manté estat de la connexió. Cada sol·licitud és independent i no té cap relació, vist des del punt de vista del servidor.

HTTP només veu bytes. Per tant, per la seva part, no entra en el format de la informació que envia. El client ja sabrà interpretar el que envia i rep (MIME: Multipurpose Internet Mail Extension, permet enviar tot tipus d'arxius dins d'un correu electrònic).

Els passos d'una connexió HTTP són:

- l'usuari accedeix a una URL, demanant un document HTML
- el servidor descodifica la URL obtenint les diferents parts (protocol, DNS o IP del servidor, port (per defecte 80) i objecte sol·licitat
- S'obre una connexió TCP entre client i servidor pel port especificat
- Es realitza la petició, enviant-se la comanda (GET, HEAD, POST), l'objecte requerit, la versió del protocol HTTP i la informació addicional necessària
- El servidor respon: retorna un codi d'estat i la informació sol·licitada si tot ha anat bé
- Es tanca la connexió TCP

Si es sol·licita al servidor un document HTML que conté quatre fotos, es provocaran cinc connexions amb el servidor. Una per l'HTML i una per a cadascuna de les fotos. Això, en les darreres versions (HTTP keep alive) s'ha millorat per optimitzar-ho, fent que tot es faci en una única connexió per minimitzar el nombre de connexions i desconnexions, donant pas a la versió HTML/1.1, que ofereix:

- connexions persistents. No es tanquen després de cada enviament. Reduïm el nombre de connexions TCP
- cada client, per la connexió establerta pot realitzar diverses peticions simultànies
- negociació del contingut: permet assignar diverses característiques de la comunicació en curs (quantitat de degradació de la connexió...)
- nous mètodes:
  - DELETE: permet esborrar un recurs del servidor
  - TRACE: seguir el tràfic amb el servidor
  - PUT: enviar dades. Similar a POST però en el comando indiquem la URL a on volem que s'actualitzi la informació
  - PATCH: aplicar correccions
  - COPY: copiar documents d'una destinació a una altra
  - MOVE: idem però movent
  - LINK: crear relacions entre documents del servidor
  - UNLINK: desfà la relació
  - OPTIONS: obtenir característiques del servidor
  - WRAPPED: unir peticions encapçulant-les amb filtrat de seguretat (encriptació)
- Nou mètode d'autentificació encriptant informació per la xarxa

Conceptes relacionats:

Cachés de pàgines:
Mecanisme que fa que el client web es guardi una pàgina a la que ha accedit. Si es torna a demanar la pàgina, farà un HEAD i esbrinarà si la pàgina ha variat o no. Si no és així, tornarà a mostrar la que va guardar en local. En cas contrari farà el GET.

Servidor proxy:
Mateixa idea però aplicada a diversos usuaris. Permetrà reduir molt el nombre d'accessos a internet, ja que entre un nombre d'usuaris propers, els accessos solen repertir-se molt. Hi ha documents i pàgines que es poden marcar amb una data d'expiració o indicar-se que no es poden passar per proxy i cal recarregar-los cada vegada.

Firewall (tallafocs):
Mecanisme per controlar i limitar les pàgines a les que s'accedeix en una organització. Fa indispensable el seu ús conjuntament amb un servidor proxy.

Webs:

http://httpd.apache.org/ (web d'Apache)
http://modules.apache.org/ (darrers mòduls afegits)
http://www.modssl.org/ (mòduls SSL – secur sockets layer)
hppt://www.apachweek.com/ (informació setmanal sobre Apache)
En castellà:
http://www.desarrolloweb.com/apache
http://www.htmlpoint.com/apache

Principals funcionalitats d'Apache:

1. servidor proxy caché.
2. Soporta scripts CGI
3. Genera cookies per descarregar al client
4. Dynamic Shared Object: càrrega i descàrrega dinàmica de mòduls durant la seva execució. Similar als mòduls del kernel de linux
5. Java: el projecte jakarta permet ficar codi java en les pàgines suportades
6. autentificació a BBDD (mSQL i mySQL)
7. suporta PHP
8. gestiona autentificacions mitjançant contrasenyes
9. Suporta PERL (similar a PHP)
10. SSL (secur sockets layer). Aplicacions de codificació i encriptació
11. redirecció d'URL's
12. Hosting virtual: Apache pot emmagatzemar diferents llocs web
13. Suporta XML (Extensible Markup Language)

Requeriments per instal·lar Apache:

Apache requereix bàsicament molt espai a disc i una bona connexió de xarxa. En quant a CPU, és poc exigent i provoca un consum força baix.

La natura dels accessos a Internet fa que el consum de recursos sigui molt puntual. En un moment donat pot ser pràcticament nul i en un altre pot ser elevadíssim.

Apache executa d'entrada vuit processos que esperaran clients. En funció del nombre de peticions concurrents, anirà obrint més processos.

Les aplicacions CGI (cal executar-les) i les transmissions https (cal encriptar-les) carregaran més el servidor.

En quant als clients, només caldrà que tinguin un navegador web. Per definir noves pàgines, caldrà un editor de HTML i tenir accés a l'espai del servidor. La transferència es farà utilitzant FTP.

Serà convenient que un servidor Apache disposi d'IP fixa, ja que sinó, totes les funcions basades en IP no es podran realitzar.

Instal·lació d'Apache:

Podem instal·lar Apache anant a la web corresponent Apache i descarregant directament la darrera versió. També ho podrem instal·lar directament de la versió de Linux que tinguem. En aquest cas, ho podem buscar com a paquets rpm o instal·lar-lo des de l'entorn gràfic.

Entorn gràfic:

Podem carregar els paquets des de la versió gràfica de l'eina de gestió de paquets que tinguem, però el més senzill serà anar directament a "Configuración del Sistema" i triar l'opció "Añadir/Eliminar aplicaciones". Això ens permetrà veure el mateix llistat de paquets que havíem triat en la instal·lació del sistema. Llavors podrem realitzar els canvis que desitgem.

Un cop instal·lat, l'estructura de directoris resultant serà la següent:

Executables	/usr/sbin
fitxer httpd.pid	/var/run
Logs	/var/log/httpd
arrel estructura web	/home/httpd/html
mapejat inicial cgi-bin	/home/httpd/cgi-bin
Configuració	/etc/httpd/conf
scripts arrencada	/etc/rc.d/
documentació en html	/home/httpd/html/manual (no !!!!) /var/www/manual/index.html.en

Un cop tinguem instal·lat apache, podrem provar d'arrencar-lo i aturar-lo:

Arrencar Apache:
 # /etc/rc.d/init.d/httpd start

Aturar Apache:
 # /etc/rc.d/init.d/httpd stop

Si instal·lem el manual, el podrem consultar per navegador amb http://localhost/manual . Això ens funcionarà sempre que Apache estigui funcionant.

Si està arrencat, podrem fer "ps –e" (comanda que permetia veure tots els processos que s'executen a la màquina, tant d'usuari com de sistema). Haurem de veure 9 processos corresponents a Apache (1 de Apache + 8 preparats per atendre peticions).

El fitxer de configuracions httpd.conf:

El fitxer de configuració es trobarà a : /etc/httpd/conf/httpd.conf
Cal recordar que sempre que efectuem alguna modificació en aquest arxiu, caldrà reinicialitzar el servei:

```
# /etc/rc.d/init.d/httpd restart
```

Aquest fitxer conté les principals configuracions d'Apache. Té tres parts:

- paràmetres globals
- directives de funcionament
- hosts virtuals

Alguns dels paràmetres tenen com a àmbit tot l'Apache. D'altres es poden definir només per a uns determinats directoris. Els paràmetres es troben dins de seccions que els agrupen. Les principals seccions són:

<Directory> : Els paràmetres definits a aquesta secció només s'aplicaran al directori especificat i als seus subdirectoris.

<DirectoryMatch>: Igual que Directory, però accepta al nom del directori expressions regulars.

Exemple d'expressió regular:

```
<DirectoryMatch "^/www/.(.+)?[0-9]{3}">
```

equivaldrà als directoris situats a `/www/` que tinguin un nom format pern tres números.

<Files>: Els paràmetres de configuració proporcionen control d'accés als fitxers pel seu nom.

<FilesMatch>: Igual que Files, però accepta expressions regulars al nom del fitxer.

<Location>: Proporciona un control d'accés als fitxers mitjançant la URL

<LocationMatch>: Igual que Location, però accepta expressions regulars al nom del fitxer.

Algunes vegades les directives de funcionament de les seccions anteriors es poden creuar. En aquest cas, tenen el següent ordre de preferència:

1. <Directory> i .htaccess (.htaccess preval davant <Directory>)
2. <DirectoryMatch> i <Directory>
3. <Files> i <FilesMatch>
4. <Location> i <LocationMatch>

Cal indicar també, que el fitxer conté moltíssims comentaris per assegurar la seva correcta utilització. Les línies comentades es marquen amb el símbol #.

Paràmetres globals:

Tots els paràmetres d'aquesta secció són globals pel funcionament del servidor, per tant no admeten estar a cap directiva.

ServerRoot: especifica la ubicació del directori arrel on es troba instal·lat Apache, a partir del qual es crea l'arbre de directoris. Aquesta directiva no s'hauria de modificar, a no ser que es mogui la carpeta d'instal·lació d'Apache a un altre directori.

User: especifica l'usuari que tindrà el procés d'Apache. Per defecte està configurat com Apache.

Group: especifica el grup d'aquest usuari. Per defecte també és Apache

UserDir: Permet assignar URL's directament a directoris d'usuari. Per exemple, podríem assignar la URL http://localhost/(caràcter 126)usuari amb el directori /home/usuari/public_html.

Per a aconseguir això, caldrà que substituïm la directiva:

UserDir disable

Per:

UserDir public_html

Aquesta directiva està desactivada per defecte per evitar que qualsevol de fora pugui saber si un usuari està donat d'alta al sistema o no. Si ho volem utilitzar s'ha d'activar amb el nom de la carpeta definida per a cada usuari.

PidFile: Ubicació del fitxer que contindrà el número d'identificació del procés quan s'encengui el servidor. Es troba disponible a través de diversos mòduls beos, leader, mpm_winnt, mpmt_os2, perchild, prefork, threadpool ó worker

ScoreBoardFile: informació de processos interns, utilitzada per comunicació entre el procés pare del servidor i els processos fills. Per defecte, aquesta informació es guarda a /etc/http/logs/apache_runtime_status.

TimeOut: el valor s'utilitza per configurar tres paràmetres diferents, mesurats en segons,:

1. El temps que pot trigar una petició en ser rebuda sencera
2. La quantitat de temps d'espera entre recepció de paquets TCP
3. La quantitat de temps entre ACK's en transmissions TCP

Passat aquest temps se genera un missatge d'error en el que s'indica que s'ha consumit el temps màxim d'espera. Establir un valor molt petit pot donar lloc a que els usuaris rebin molts missatges d'error. En canvi, establir un valor molt gran donarà lloc a una sobrecàrrega de la màquina. Es troba disponible a través del mòdul Core.

KeepAlive: especifica si s'utilitzaran connexions persistents: En aquest cas, totes las peticions d'un usuari s'atendran amb la mateixa connexió. Es troba disponible a través del mòdul Core.

MaxKeepAliveRequests: nombre màxim de connexions persistents. (número màxim d'usuaris concurrents si KeepAlive és ON). Per establir aquest paràmetre cal tenir en compte l'ample de banda de sortida del servidor. Si s'estableix un valor molt gran respecte l'ample de banda, el temps de resposta es veurà incrementat per a cada usuari. Es troba disponible a través del mòdul Core.

KeepAliveTimeout: temps que espera, en segons entre peticions d'un usuari abans de considerar que aquest ha terminat i tancar la seva connexió.

Si el valor és molt petit provocarà que alguns usuaris no puguin visualitzar la pàgina. Si s'estableix un valor molt gran s'estarà utilitzant molts recursos de la màquina. Es troba disponible a través del mòdul Core.

Listen: Permet fixar la IP i el port a utilitzar per atendre les peticions. Per defecte s'utilitza el port 80 (www). També permet especificar quines direccions IP s'atendrà. Per defecte totes. Per a atendre dues direccions IP diferents, amb diferents ports, s'utilitzaria:

Listen 192.168.255.5:80
Listen 192.168.255.8:8080

Es troba disponible a través de diferents mòduls beos, leader, mpm_winnt, mpmt_os2, perchild, prefork, threadpool ó worker

LoadModule: Directiva que serveix per carregar mòduls que inclouen diferents funcionalitats. La sintaxi es:

LoadModule nomModul ubicacioFitxer

Es troba disponible a través del mòdul mod_so.
La distribució en mòduls d'Apache es fa seguint l'estratègia de Dynamic Shared Objects (DSO).

IfDefine: permet executar una sèrie de directives si s'acompleix una condició

StartServers: Especifica el nombre de processos es crearan al arrencar._

MaxRequestsPerChild: Permet especificar un màxim nombre de clients que atendrà un procés fill. Per defecte es fixa a 4000. Es fa per evitar que un procés, al atendre a molts clients, acabi gastant molta memòria.

Directives de funcionament:

Aquesta és la principal secció de configuració del servidor, hi podrem trobar las següents opcions:

ServerAdmin: especifica l'adreça de correu electrònic de l'administrador, aquesta adreça apareix a tots els missatges d'error, per permetre a l'usuari notificar un error al administrador. No pot estar dins de cap secció. Es troba disponible a través del mòdul Core.

ServerSignature: Afegeix una línia que contindrà la versió Apache i el ServerName a tots els documents generats pel servidor (missatges d'error). Per defecte està activat (on). Es pot desactivar o posar-li el valor EMail, que afegeix una línia indicant que s'enviï un e-mail a ServerAdmin.

ServerName: especifica el nom i el port que el servidor utilitza per identificar-se. Normalment es determina automàticament, però és recomanable especificar-lo explícitament per evitar problemes al iniciar el servidor. Si el servidor no té un nom registrat a les DNS, es recomana posar el seu número IP. No pot estar dins de cap secció.

La sintaxis es:

ServerName adreçaIP:Port p.e. ServerName localhost:80

www.exemple.com:80

Es troba disponible a través del mòdul Core.

UseCanonicalName: fa que el servidor s'hagi de referir a sí mateix pel nom especificat a ServerName i Port. Per a això, cal assignar-li el valor on. Si val off, s'hi referirà pel nom donat pel client. Per defecte està a off.

DocumentRoot: carpeta arrel que s'ubica al servidor, des de la que es serviran els documents. Per defecte, totes las peticions, tindran com arrel aquesta carpeta, a no ser que s'utilitzin alies (directoris virutals en IIS)

Per defecte, la carpeta arrel és la carpeta /var/www/html. No pot estar dins de cap secció.

Si es modifica aquest directori per un altre, és molt important posar el nou valor, no només a aquesta línia, sinó també a la secció <Directory> en la que s'estableixen els paràmetres de configuració d'aquest directori.

Es troba disponible a través del mòdul Core.

DirectoryIndex: especifica el fitxer per defecte que cercarà en cada directori, en cas de que no s'especifiqui cap. Per defecte és index.html. Es a dir, si posem al navegador: www.lamevaweb.com el servidor, per defecte servirà www.lamevaweb.com/index.html

Aquesta directiva permet especificar més d'un fitxer, la sintaxis es la següent:

DirectoryIndex fitxer1 fitxer2 fitxer3

L'ordre en el que s'especifica el nom de fitxer determinarà la prioritat a l'hora de decidir el fitxer a mostrar.
Aquesta directiva es pot trobar fóra de qualsevol secció, dins d'una secció o a un fitxer .htaccess.

Es troba disponible a través del mòdul mod_dir.

AccessFileName: Indicarà el nom del fitxer de configuració que es cercarà per a cada directori del servidor per conèixer la configuració del mateix. Aquest fitxer permet configurar el

comportament de cadascun dels directoris individualment. Per fer que aquesta configuració funcioni, la directiva AllowOverride ha de tenir un valor que ho permeti. No pot estar dins de cap secció.

El nom de fitxer que s'especifica per defecte és ".htaccess".

Options: Controla característiques del servidor disponibles en un directori concret. Pel directori arrel només permet FollowSymLinks (pot seguir enllaços simbòlics al directori arrel).

AllowOverride: Indica si es poden sobreescriure les opcions per arxius .htaccess. Per tant, si AllowOverride val none, no es tenen en compte els .htaccess. Si li posem AuthConfig, tindrà en compte els arxius indicats

Exemple:

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

Order: indica l'ordre en que s'avaluaran allow i deny, que permeten activar o eliminar l'accés al directori DocumentRoot.

Com a mesura de seguretat la configuració d'Apache estableix que no es mostri la existència d'aquest fitxer a cap usuari, encara que estigui establerta l'opció de llistat de directoris. Si es decideix modificar el nom, caldrà redefinir la seguretat per fer que no es mostri el contingut del nou fitxer. Això es fa al fitxer httpd.conf en una secció File com la que es presenta a continuació en la que establim que tots els fitxers que comencin por .ht no es mostrin.

```
<Files ~ "^\.ht">
   Order allow,deny
   Deny from all
</Files>
```

Es troba disponible a través del mòdul Core.

TypesConfig: especifica el nom del fitxer que conté la llista de tipus mime que coneix el servidor, i que determinarà, depenent de les extensions para generar las capçaleres http. No pot estar dins de cap secció.

MIME: Multipurpose Internet Mail Extensions. Estàndard definit per unificar l'enviament de fitxers a través d'internet en sistemes operatius que no sempre seran iguals. MIME proporciona que la informació que no sigui caràcters ASCII amb 7 bits (128 valors) es codifiqui com si fossin caràcters i s'enviin dins el correu. Un cop arribats, el mateix MIME serà l'encarregat de decodificar el fitxer i tornar-lo al seu format original. Qualsevol eina que permeti treballar amb MIME haurà de definir els formats que suporta. També permet que es puguin enviar en correus els caràcters extesos (ç,ñ,...).

Es troba disponible a través del mòdul mod_mime.

DefaultType: tipus mime que es servirà per defecte en cas de no conèixer l'extensió del fitxer que s'està servint. Per defecte, s'indicarà que se serveix texte plà amb el valor text/plain. La directiva es pot trobar fora de qualsevol secció, dins d'una secció o dins d'un fitxer .htaccess.

Sintaxi: DefaultType tipusMime

Es troba disponible a través del mòdul Core.

HostnameLookups: s'utiliza en fitxers de registre. Per defecte quan es produeix un accés, se guarda simplement el seu número IP, si aquesta directiva es troba en On, el servidor cercarà la correspondència d'aquesta IP amb el seu nom, i guardarà el nom.
Establir aquesta configuració a ON provocarà que al menys calgui fer una petició al servidor de noms per a cadascuna de les peticions d'usuario. Això implica que el rendiment de la màquina es pot veure decrementat. Aquesta directiva es pot trobar dins d'una secció o fora.

Es troba disponible a través del mòdul Core.

ErrorLog: especifica la ubicació del fitxer que conté el registre d'errors. Per defecte a la carpeta logs. Aquesta directiva només es pot trobar fora de qualsevol secció.

Es troba disponible a través del mòdul Core.

LogLevel: especifica el tipus de missatges que es guardaran al fixer de registre d'errors, depenent dels valors especificats, es guardaran més o menys. Aquesta directiva només es pot trobar fora de qualsevol secció.

Valor que es poden triar són: debug, info, notice, warn, error, crit, alert, emerg

Es troba disponible a través del mòdul Core.

LogFormat: Permet definir el format que s'utilitzarà per emmagatzemar els registres. A cada format se li pot assignar un nom, utilitzant-ho després per crear diferents tipus de fitxers de registre. Poden existir diversos logFormat.

Sintaxi:

LogFormat "configuracióError" nom

Aquesta directiva ha de trobar-se fora de qualsevol secció.

Es trob disponible a través del mòdul mod_log_config.

CustomLog: S'utilitza per especificar la ubicació i el tipus de format que s'utilizarà als fitxers de registre. Poden existir diversos fitxers de registre diferents amb configuracions distintes. Per fer això, simplement caldrà posar més d'una línia customlog

Sintaxi: CustomLog fitxer format

Aquesta directiva es trobarà fora de qualsevol secció.

Es troba disponible a través del mòdul mod_log_config.

ServerTokens: Aquesta directiva estableix la informació que es torna dins la capçalera http que envia el servidor. Possibles valors de menor a major informació son:
-Pord
-Min
-Os
-Full

Aquesta directiva es trobarà fora de qualsevol secció.

Es troba disponible a través del mòdul Core.

IndexOptions: Controla l'aparièncía dels llistats generats pel servidor. Farà que, si es demana un directori sense índex, aparegui els fitxers del directori per triar-ne un. L'opció per defecte és FancyIndexing, que permetrà ordenar el llistat d'arxius del directori clickant a les capçaleres. Se li poden afegir més paràmetres per controlar l'aparièncía del llistat.

Sintaxis:

IndexOptions [+|-]opcion [[+|-]opcion] ... (Apache 1.3.3 en endavant)

Entre les opcions que es poden posar, destaca:

FancyIndexing: que mostra els noms dels fitxers, amb icones etc..

Es troba disponible a través del mòdul mod_autoindex.

FoldersFirst: Fa que primer es mostrin els directoris. Aquesta opció només es pot establir en el cas que FancyIndexing estigui activa.

Aquesta directiva es pot trobar dins del fitxer .htaccess, dins d'una secció <Directory> i fora de qualsevol altra.

Es troba disponible a través del mòdul mod_autoindex.

AddIconByEncoding: Permet associar un icone a un tipus mime, de forma que quan la directiva fancyIndexing estigui activada, es mostrarà al costat del fitxer l'icone corresponent.

Sintaxi:

AddIconByEncoding icon MIME-encoding…

Exemple:

AddIconByEncoding/icons/compressed.gif x-compress

Aquesta directiva es pot trobar dins del fitxer .htaccess, dins d'una secció <Directory> y fora de qualsevol altra.

Es troba disponible a través del mòdul mod_autoindex.

AddIconByType: Associa un icone a un fitxer depenent del tipus mime, de forma que quan la directiva fancyIndexing estigui activada, es mostrarà al costat del fitxer l'icone corresponent.

Sintaxi:

AddIconByType icon MIME-encoding…

Exemple:

AddIconByType /icons/text.gif text/*

La diferència entre AddIconByType i AddIconByEncoding es troba en que mentre que en la primera es determina el tipus mime basant-se en la codificació del fitxer. AddIconByType determina el tipus mime basant-se en el nom del fitxer.

Ambdues directives es poden trobar dins del fitxer .htaccess, dins d'una secció <Directory> o fora de qualsevol altra.
Es troba disponible a través del mòdul mod_autoindex.

AddIcon: indica al servidor l'icone a mostrar en un directori en funció de l'extensió del fitxer

DefaultIcon: icone per defecte quan un fitxer no casi amb cap definició AddIcon

AddDescription: Permet associar una descripció a un tipus de fitxer, que es mostrarà al llistar un directori. Aquesta directiva es pot trobar dins del fitxer .htaccess, dins d'una secció <Directory> o fora de qualsevol altra.

Sintaxi:

AddDescription cadena , fitxer

Es troba disponible a través del mòdul mod_autoindex.

AddDefaultCharset: Defineix la codificació de caràcters que s'utilitzarà de forma predeterminada pels documents. Per defecte s'estableix el valor ISO-8859-1. Aquesta directiva es pot trobar dins de qualsevol secció i als fitxers .htaccess.

Es troba disponible a través del mòdul Core.

ErrorDocument: Estableix la configuració del servidor quan es produeix un error. Es poden establir quatre configuracions diferents:

1. Treure un texte d'error
2. Redirigir a un fitxer al mateix directori
3. Redirigir a un fitxer al nostre servidor
4. Redirigir a un fitxer fora del nostre servidor

Cal indicar que si el texte d'error s'envia a Internet Explorer, aquest haurà de tenir almenys 512 Bytes, ja que en cas contrari Internet Explorer mostrarà la seva pròpia pàgina d'error.

Sintaxi:

ErrorDocument NúmeroError Acció

Aquesta directiva es pot trobar tant dins del fitxer .htaccess, dins de la secció <Directory> o fora de qualsevol altra secció.

Exemple:

ErrorDocument 404 /error404.html.

En cas de no trobar-se un fitxer, es mostrarà el fitxer error404.html

Es troba disponible a través del mòdulo Core.

CacheRoot: estableix el directori on es trobaran els fitxers de la cache d'Apache.

Es troba disponible a través del mòdul mod_disk_cache

CacheSize: Tamany de la cache en Kilobytes.

Es troba disponible a través del mòdul mod_disk_cache

CacheGcInterval: Estableix la freqüència d'hores amb la que es verificarà el tamany dels fitxers de la cache per comprobar si es corresponen amb el tamany establert a CacheSize. El valor accepta números flotants, per tant, podrem establir els intervals en minuts. Quant més gran sigui el valor d'aquesta directiva, més probable serà sobrepassar el valor establert a CacheSize.

Es troba disponible a través del mòdul mod_disk_cache

CacheMaxExpire: Permet fixar el màxim nombre d'hores que els fitxers romandran a la cache.

Es troba disponible a través del mòdul mod_cache

CacheLastModifiedFactor: Serveix per calcular la caducitat d'un fitxer a la cache. El factor que marcarà la caducitat es calcularà multiplicant aquest falor per la hora de darrera modificació.

Es troba disponible a través del mòdul mod_cache

CacheDefaultExpire: Nombre d'hores per defecte a partir de les quals un fitxer caduca. S'aplica en aquells casos en els que no es pot determinar l'hora de creació del fitxer.
Es troba disponible a través del mòdul mod_cache

Totes les directives de la caché, han de trobar-se fora de qualsevol secció.

Proxy:
Per configurar Apache per fer que actuï com a servidor proxy, cal :

1) Desasteriscar:
   LoadModule proxy_module modules/mod_proxy.so
   per fer que es carregui aquest mòdul
2) Posar ProxyRequests a on

A més, podem definir sentències exclusives pel proxy fent:

```
<Proxy *>
    Order deny,allow
    Deny from all
    Allow from .example.com
</Proxy>
```

Podem utilitzar sentències de configuració del proxy (al mig de l'arxiu httpd.conf) (aquí només n'hem posat algunes com a mostra, n'hi ha més):

ProxyRemote http://ServidorRemot:port : permet definir un proxy extern
ProxyBlock adreçaweb : permet especificar una adreça web a la que no es deixarà accedir

També puc restringir l'accés al proxy: a l'arxiu access.conf posaré:

<Directory proxy:* >
Order, deny,allow
deny from all
allow from 192.168.1.0/255.255.255.0 .micasa 200.9.21.200
</Directory>

a on estic dient les màquines que podran accedir al proxy

Està clar que per a assegurar que tot això funcioni, hauré de configurar també els meus clients (navegadors web) per fer que utilitzin el meu proxy.

Creació de directoris virtuals en Apache:

Directori virtual: es un directori que es troba en un directori diferent del que es mapeja a la URL.

Degut a la seva importància, la directiva per crear directoris virtuals només es pot definir al fitxer de configuració httpd.conf. La tractarem apart.

Alias

Permet la definició de directoris virtuals. El directori virtual no ha de trobar-se obligatòriament dins l'arbre de directoris que es crea a partir de DocumentRoot, sinó que es pot trobar en qualsevol altra ubicació, inclús es podria trobar en un altre servidor.

Per exemple, si escric www.lamevapaginaweb.com/manual/php, la carpeta php no ha de trobar-se necessàriament dins de la carpeta manual, que penja de la carpeta arrel lamevapaginaweb, sinó que pot trobar-se en una ubicació diferent, fora de l'arbre de subdirectoris de la directiva DocumentRoot

Sintaxi:
Alias nomFictici ubicacióReal

Exemple:
Alias /manual/php "c:\php"

- el directori php no es troba dins del directori manual dins la carpeta documentRoot, sinó a la carpeta c:\php.

Per defecte, a l'arxiu de configuració tindrem creades dues redireccions amb Alias.
- Icons: per establir la carpeta on es troben els icones que utilitzarà Apache per mostrar el contingut dels directoris
- Manual: que apunta a la carpeta on està instal.lat el manual d'Apache, sempre i quan, en la instal.lació haguem triat també instal.lar el manual.

ScriptAlias: permet definir una carpeta a on es troben els scripts cgi. Per defecte conté:

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

AliasMatch

La utilitat d'aquesta directiva és idèntica a la de la directiva Alias, amb la diferencia que mentre Alias utilitza expressions estandard regulars per especificar la URL que es mapejarà.

Sintaxi:
AliasMatch Expressió regular d'ubicació

Virtual Host a Apache:

Denominarem Hosting Virtual a la capacitat d'allotjar més d'un lloc web a una mateix màquina (www.webmeva1.com i www.webmeva2.com).

Apache ha sigut dels primers servidors web en permete Virtual Host.

Permet llocs virtuals basats en IP (cadascun dels llocs té la seva pròpia IP) o basats en noms (comparteixen IP però cadascun té un nom de domini). El fet que s'executin a la mateixa màquina ha de ser totalment transparent a l'usuari.

El Hosting virtual basat en noms és més senzill. Es necessitarà configurar el servidor de DNS per fer que associï la IP corresponent i configurar Apache per que reconegui els diferents noms de host. D'aquesta forma reduirem també la demanda d'adreces IP, que ja comença a ser un tema problemàtic.

Hi haurà una sèrie de casos en els que haurem d'utilitzar el Hosting Virtual basat en IPs:
- alguns clients antics no suporten hosting virtual basat en noms. Alguns navegadors antics no envien una capçalera host HTTP. Aquest fet és necessari per poder utilitzar hosting virtual basat en noms. Hi ha tècniques per simular aquest fet
- El protocol SSL no suporta host virtual basat en noms
- Alguns SO i alguns elements de xarxa necessiten explícitament diferenciar per IP

Host Virtual basat en noms:

Caldrà definir un bloc NameVirtualHost per a cada web definida.

Cal recordar que si afegim dues webs a un servidor que ja en contenia una, caldrà definir tres blocs. Un per la que ja existia i els dos corresponents a les noves.

En aquests blocs també definirem la IP corresponent a cada nom. Si volem que el servidor reconegui aquella IP, caldrà tenir un listen per a cadascuna d'elles.
Hem de tenir en compte que si al Host Virtual especifiquem que aquest va lligat a una IP per la que no estem escoltant (no hem fet listen), no s'hi accedirà mai.

```
NameVirtualHost *:80

<VirtualHost *:80>
ServerName www.domain.tld
ServerAlias domain.tld *.domain.tld
DocumentRoot /www/domain
</VirtualHost>

<VirtualHost *:80>
ServerName www.otherdomain.tld
DocumentRoot /www/otherdomain
</VirtualHost>
```

NameVirtualHost defineix la adreça IP i port (enlloc de l'asterisc podríem posar una adreça IP) pels que treballarem i, a continuació, cada bloc VirtualHost (lligat amb el NameVirtualHost donat) indicarà el nom que servirem (com a mínim ha de tenir la directiva ServerName).
Amb ServerAlias permetem que el mateix web (www.domain.tld) sigui accedit amb diferents noms (domain.tld, *.domain.tld).

Per fer que tot això funcioni, caldrà que el servdor DNS estigui correctament configurat per fer que els noms indicats a ServerName i a ServerAlias es corresponguin a les IPs.

Quan es rep una petició, es verifica si s'està utilitzant la adreça IP especificada amb NameVirtualHost. Llavors, la directiva ServerRoot principal del servidor no tindrà efecte, ja que si emparellem la IP rebuda amb la especificada a NameVirtualHost, tindrem en compte els valors definits als blocs virtualHost.

Intentarem casar el nom sol·licitat amb els especificats als diferents VirtualHost. Si no en casem cap agafarem, per defecte, el primer. Per tant, si volem que s'actuï d'una determinada manera si no casem el nom sol·licitat, caldrà definir un bloc VirtualHost per aquest cas i posar-lo el primer de tots.

Host Virtual basat en IP:

Abans de tot, cal tenir en compte que el fet d'utilitzar Host Virtual incrementarà molt el consum de recursos. Més clients, més logs,..... Cal preveure l'impacte d'aquest ús de recursos.

En aquest cas, el servidor tindrà definida una IP per a cada web a la que dona servei.
Podrem fer això de dues maneres diferents:

1) Per a cada Hostname executem un dimoni HTTP. És a dir, executem un servidor Apache per a cada Hostname. Farem això quan les dues webs que volem contenir necessiten explícitament que tota la seva informació sigui disjunta (per raons de seguretat). Llavors cadascuna ha de tenir les seves pròpies directives user, group, listen i serverroot (directives que no poden estar dins d'un bloc virtualhost).
   En aquest cas realitzarem una instal·lació per a cada web. En cadascuna d'aquestes haurem d'especificar la IP atesa amb la directiva listen. Aquí serà recomanable indicar a la directiva listen la IP i no un DNS
   Cadascun d'ells tindrà diferents directives user, group, listen i serverroot
2) Compartirem l'execució del servidor Apache quan les configuracions del servidor necessàries per a cadascuna de les webs siguin compatibles. El fet de definir més d'un servidor provocaria una càrrega massa grossa per a la màquina.

En aquest cas haurem d'utilitzar la directiva VirtualHost:

```
<VirtualHost www.smallco.com>
ServerAdmin webmaster@mail.smallco.com
DocumentRoot /groups/smallco/www
ServerName www.smallco.com
ErrorLog /groups/smallco/logs/error_log
TransferLog /groups/smallco/logs/access_log
</VirtualHost>

<VirtualHost www.baygroup.org>
ServerAdmin webmaster@mail.baygroup.org
DocumentRoot /groups/baygroup/www
ServerName www.baygroup.org
ErrorLog /groups/baygroup/logs/error_log
TransferLog /groups/baygroup/logs/access_log
</VirtualHost>
```

En aquest cas es recomana utilitzar la adreça IP enlloc del DNS (com a l'exemple).

Dins dels apartats VirtualHost podrem posar gairebé totes les directives, excepte aquelles que no estan permeses d'utilitzar dins d'un apartat.

Cal anar amb compte si algú més que l'usuari que executa Apache té accés d'escriptura als directoris de log, ja que això ens podria portar problemes.

Servidor segur:

Apache també pot fer de servidor segur (HTTPS). Per suportar HTTPS Apache fa servir SSL. Per a això té definit un altre port per defecte (443).

Aquesta transmissió és més costosa i més lenta que la HTTP i, per això, cal minimitzar-la a lo estrictament imprescindible. Es recomana no utilitzar virtual host amb connexions segures.

Les configuracions segures tenen les seves pròpies directives de configuració a l'arxiu /etc/hhtp/conf.d/ssl.conf.

Per defecte el servidor segur i el no segur comparteixen el documenroot. És recomanable definir diferents DocumentRoot.