

El servidor HTTP Apache (I): Introducció i posada en marxa

Un servidor "web" és una aplicació que normalment roman escoltant en segon pla per tal d'oferir en el moment que els clients -normalment remots- ho demanin no només pàgines web (d'aquí el nom) sinó molts altres recursos de diferents tipus que sol tenir emmagatzemats al seu disc dur (com ara imatges, vídeos, fitxers de text, etc, etc). Els navegadors són els clients "web" més habituals. Tant els servidors "web" (per rebre i interpretar correctament la petició i generar i retornar la resposta pertinent) com els navegadors (per enviar la petició i interpretar correctament la resposta rebuda) han de poder establir comunicació entre ells mitjançant un determinat protocol que ambdós entenguin: el protocol HTTP (<https://http2.github.io>), i és per això que a aquest tipus de servidor també se'ls anomena "servidors HTTP".

El servidor HTTP Apache (<http://httpd.apache.org>) és el més utilitzat en el món, tal com es pot veure a les "Web Server Survey" mensuals disponibles a <https://news.netcraft.com> (on es pot anar seguint la popularitat dels diferents servidors més importants). A més a més, és lliure, multiplataforma i gratuït. Aquest document explicarà les característiques, configuracions i usos més habituals d'aquest programa, però si és necessària qualsevol ampliació d'informació, tot el que necessitareu serà consultar la excel·lent documentació oficial, disponible a <http://httpd.apache.org/docs/current>.

NOTA: Altres servidors HTTP importants són Nginx (<https://nginx.org>) o Caddy (<https://caddyserver.com>), entre altres.

Instal·lació de l'Apache i posada en marxa

A Ubuntu

Un cop està tot el sistema actualitzat (`sudo apt update && sudo apt upgrade`) procedirem a instal·lar l'Apache simplement executant: `sudo apt install apache2`

NOTA: Si anéssim a l'apartat de descàrregues de la web de l'Apache veuríem que allà hi ha una versió més moderna que la que es s'instal·la des dels repositoris de l'Ubuntu, però només està disponible en forma de codi font (és a dir, l'hauríem de compilar per tenir l'executable) i el procés, tot i que no és complicat, no ens val la pena ja que els canvis entre la versió més moderna i la que ofereix Ubuntu són mínims

Un cop instal·lat, el servidor es posarà en marxa automàticament (escoltant al port 80) sense haver de fer res especial. Per comprovar-ho un mateix, es pot obrir un navegador qualsevol i accedir a la direcció <http://127.0.0.1> (o bé <http://nom>, on "nom" és el nom associat a la ip 127.0.0.1 dins del nostre arxiu "/etc/hosts" -normalment aquest nom ja hi és i és "localhost"-) : si tot ha anat bé s'hauria de veure una pàgina web explicativa del programa. Si hom volgués connectar-se al servidor Apache d'un company, llavors al navegador hauria d'escriure <http://IP.LAN.del.company> (o bé <http://nom>, on "nom" és el nom associat a la IP del servidor del company dins del nostre arxiu "/etc/hosts").

Per iniciar, aturar o reiniciar manualment el servei s'ha de procedir com qualsevol altre servei Systemd del sistema, respectivament: `sudo systemctl start apache2` ; `sudo systemctl stop apache2` ; `sudo systemctl restart apache2` També està la comanda `systemctl status apache2` per veure si el servei està encès o no. Per fer que s'iniciï sempre automàticament en arrencar el sistema, s'ha d'executar `sudo systemctl enable apache2` (i per deshabilitar-lo permanentment -fins que no es torni a habilitar-, `sudo systemctl disable apache2`).

A Fedora

Un cop està tot el sistema actualitzat (`sudo dnf upgrade`) procedirem a instal·lar l'Apache simplement executant: `sudo dnf install httpd` Un cop instal·lat, caldrà posar en marxa el servidor (amb `sudo systemctl start httpd`) i/o habilitar-lo (amb `sudo systemctl enable httpd`). La resta d'accions Systemd també estan disponibles, òbviament: "stop", "restart", "disable", "status", etc (per més informació, pots consultar `man httpd.service`). Un cop posat en marxa, es pot comprovar que, efectivament, el servidor està escoltant al port 80, de la mateixa manera que s'ha descrit a l'apartat anterior: obrint un navegador qualsevol i escrivint a la barra de direccions la URL <http://127.0.0.1> (s'obté la pàgina web de mostra del programa).

NOTA: Una altra manera de comprovar que el servidor web està escoltant és veure si el port nº80 de la màquina on aquest servidor hauria d'estar funcionant està, efectivament, obert. Això es pot fer observant la sortida de la comanda `ss -tln`

NOTA: Si no es pot accedir a la pàgina de mostra des del navegador, comproveu que no sigui el tallafocs del Fedora qui ho estigui implicant. Per veure-ho ràpid, podeu aturar-lo (`sudo systemctl stop firewalld`) i tornar-ho a intentar

Estructura global de carpetes i fitxers de configuració

A Ubuntu

Dins de la carpeta `/etc/apache2` es troben la majoria de subcarpetes i fitxers de configuració de l'Apache (els quals són simples fitxers de text pla amb un determinat contingut). Concretament, els fitxers més importants són:

apache2.conf : Fitxer de configuració principal del programa. Quasi tota la configuració general es pot establir dins d'aquest fitxer, encara que es recomana (per simplicitat) utilitzar fitxers separats específics (ubicats a les subcarpetes `mods-enabled/`, `conf-enabled/` i `sites-enabled/`), els quals de seguida estudiarem. Per tant, quasi bé mai caldrà canviar res d'aquest fitxer en concret.

ports.conf : Fitxer (referenciat des de `apache2.conf` amb la directiva *Include*) on s'especifica el/s port/s per on escoltara/n els diferents "virtual hosts" que haguem definit (no hi ha cap problema en què tots ells escoltin per un únic port i una mateixa IP). Un "virtual host" representa un lloc web que funciona (i està configurat) de forma totalment independent respecte a un altre lloc web -és a dir, un altre "virtual host"-, però que està definit juntament amb la resta de "virtual hosts" dins un únic servidor Apache (és a dir, l'únic servidor "real", d'aquí el nom). Bàsicament, el que ens permeten els "virtual hosts" és tenir un lloc anomenat, per exemple, `www.hola.com` i un altre anomenat `www.adeu.com` gestionats pel mateix servidor Apache. Aquest fitxer `ports.conf` és important tenir-lo en compte sobre tot a l'hora de configurar servidors HTTPS (és a dir, servidors HTTP combinats amb la capa de seguretat i encriptació TLS), tal com estudiarem més endavant.

envvars : Fitxer on es defineixen les variables d'entorn que es necessiten tenir establertes per a què l'Apache pugui funcionar correctament (ja que el seu valor s'utilitza al llarg de la resta de fitxers de configuració). Per exemple, `APACHE_RUN_USER` i `APACHE_RUN_GROUP` (que serveixen per establir l'usuari i grup sota els que s'executarà el servidor -i per tant, els que definiran els permisos que té el programa-; ambdues variables valen `www-data`, un usuari "virtual" específic que es crea en instal·lar l'Apache) o `APACHE_PID_FILE` (que serveix per establir la ruta del fitxer temporal que conté el PID del procés Apache en aquest moment; val `/var/run/apache2/apache2.pid`) o `APACHE_LOG_DIR` (que serveix per establir la ruta de la carpeta on s'emmagatzemaran els diferents arxius de registres -els "logs"- del servidor; val `/var/log/apache2`), entre altres. Nosaltres no necessitarem modificar aquest fitxer mai.

I les subcarpetes més importants són:

sites-available/ : Subcarpeta on s'ubiquen els fitxers de configuració corresponents a cada "virtual host" definit. Per defecte Ubuntu ofereix dos "virtual hosts" ja llestos per fer servir: un de tipus HTTP (arxiu `000-default.conf`) i un altre de tipus HTTPS (arxiu `default-ssl.conf`).

NOTA: La raó de què el nom del fitxer `.conf` del "virtual host" HTTP comenci amb "000" és perquè la lectura dels diferents fitxers `.conf` ubicats en aquesta subcarpeta es realitza en ordre alfabètic: d'aquesta manera, `000-default.conf` sempre serà el primer en ser llegit.

sites-enabled/ : Subcarpeta que conté enllaços directes apuntant a fitxers ubicats dins de la subcarpeta `sites-available/`. La existència d'un enllaç concret implica l'activació del "virtual host" associat (o dit d'una altra manera, per desactivar un "virtual host" és suficient amb esborrar l'enllaç d'aquesta carpeta i per tornar-lo a activar només cal refer l'enllaç amb una simple comanda `ln -s`). Per defecte només el "virtual host" HTTP està activat.

mods-available/ : Subcarpeta on s'ubiquen els fitxers corresponents a la gestió dels diferents mòduls ("plugins") que el servidor Apache té disponibles per utilitzar. Els mòduls de l'Apache són "trossos" de programa que, en general, augmenten la funcionalitat del programa base (encara que entre sí són força diferents en termes d'objectius i funcionament). Dins d'aquesta carpeta cada mòdul sol tenir associat un arxiu amb extensió `.load` (imprescindible) i un altre amb extensió `.conf` (opcional), ambdós amb el mateix nom (el del propi mòdul). Els arxius `.load` contenen sols una línia de text com `LoadModule nomModul_module /usr/lib/apache2/modules/nomFitxer.so`, la qual s'encarrega de carregar efectivament el mòdul real (que és el fitxer binari amb extensió `.so` -concretament, es tracta d'una llibreria dinàmica-) a la memòria RAM de la màquina. Els arxius `.conf` contenen la configuració del mòdul pertinent, la qual pot ser molt diferent de mòdul a mòdul.

*"**mods-enabled**" : Subcarpeta que conté enllaços directes apuntant a fitxers ubicats dins de la subcarpeta "*mods-available*". La existència d'un enllaç concret implica l'activació del mòdul associat i, per tant, la possibilitat d'utilitzar la funcionalitat que aquest mòdul aporta (o dit d'una altra manera, per evitar que es pugui fer servir aquesta funcionalitat extra -és a dir, per "desactivar" el mòdul- és suficient amb esborrar l'enllaç d'aquesta carpeta, i per tornar-la a activar només cal refer l'enllaç amb una simple comanda *ln -s*).

*"**conf-available**" : Subcarpeta on s'ubiquen fitxers amb trossos de configuració extra (generalment relatius a aspectes molt específics) que són interpretats com si fossin part de l'"*apache2.conf*". La gràcia està en què si la configuració es troba en algun tros dins d'aquesta carpeta no cal tocar el fitxer de configuració principal. Concretament, a una instal·lació per defecte aquí ens trobem els següents fitxers: "*charset.conf*", "*localized-error-pages.conf*" (ambdós sense contingut efectiu), "*other-vhosts-access-log.conf*" (contenint una sola línia *CustomLog*) i "*security.conf*" i "*serve-cgi-bin.conf*" (ambdós amb contingut relacionat amb l'àmbit indicat pel seu nom)

*"**conf-enabled**" : Subcarpeta que conté enllaços directes apuntant a fitxers ubicats dins de la subcarpeta "*conf-available*". La existència d'un enllaç concret implica l'activació de la configuració extra associada (o dit d'una altra manera, per no tenir en compte aquest tros de configuració -és a dir, per "desactivar-lo"- és suficient amb esborrar l'enllaç d'aquesta carpeta, i per tornar-lo a tenir en compte només cal refer l'enllaç amb una simple comanda *ln -s*).

Cal tenir present que qualsevol canvi que es guardi en algun dels fitxers i subcarpetes anteriors no serà efectiu fins reiniciar el servidor (o bé, de forma alternativa, executar la comanda *systemctl reload apache2*, la qual obliga al servidor a llegir de nou tots els arxius de configuració sense que calgui reiniciar).

En lloc d'haver d'utilitzar manualment la comanda *ln -s* cada cop que volguem activar o desactivar mòduls (i/o "virtual hosts" i/o configuracions extra) podem aconseguir el mateix (és a dir, crear o destruir els enllaços directes pertinents) fent servir el següent conjunt de comandes (instal·lades automàticament amb el servidor) molt senzilles:

<i>a2enmod nomModul</i> : activa el mòdul indicat	<i>a2dismod nomModul</i> : desactiva el mòdul indicat
<i>a2ensite nomarxiuVH</i> : activa el "virtual host" indicat	<i>a2dissite nomarxiuVH</i> : desactiva el "virtual host" indicat
<i>a2enconf nomarxiuConf</i> : activa la configuració indicada	<i>a2disconf nomarxiuConf</i> : desactiva la configuració indicada

Després d'executar alguna de les comandes anteriors, caldrà reiniciar/recarregar el servidor per a què els canvis tinguin efecte.

També és interessant la comanda *a2query*, la qual amb el paràmetre *-m* mostra la llista de mòduls activats en aquest moment, amb el paràmetre *-s* mostra la llista de "virtual hosts" activats en aquest moment i amb el paràmetre *-c* mostra la llista de trossos de configuració extra activats en aquest moment.

A Fedora

En aquest cas, és dins de la carpeta "**/etc/httpd**" on es troben les subcarpetes i fitxers de configuració de l'Apache. Concretament, els fitxers més importants són:

*"**conf/httpd.conf**": Arxiu de configuració principal, el qual inclou l'equivalent al contingut dels fitxers l'"**apache2.conf**", el "**ports.conf**" i el "**000-default.conf**" existents als sistemes Ubuntu

*"**conf.d**": Subcarpeta (referenciada des de "*httpd.conf*" amb la directiva *Include*) que conté diversos arxius "**.conf*" amb trossos de configuració extra. La gràcia està en què si la configuració en qüestió es troba en algun tros dins d'aquesta carpeta no cal tocar el fitxer de configuració principal. Però no només hi podem trobar configuracions genèriques; si es volguessin afegir més "virtual hosts", també caldria fer-ho cadascun en un fitxer ".conf" separat dins d'aquesta carpeta. Podem concloure, per fer un símil, que el contingut d'aquesta subcarpeta seria equivalent al de les subcarpetes "**/etc/apache2/conf-enabled**" i "**/etc/apache2/sites-enabled**" (menys "*000-default.conf*"), i també al dels arxius "**/etc/apache2/mods-enabled/*.conf**". A una instal·lació per defecte ens trobem els següents fitxers: "*autoindex.conf*", "*userdir.conf*" i "*welcome.conf*", els quals estudiarem properament.

*"**conf.modules.d**": Subcarpeta (referenciada des de "*httpd.conf*" amb la directiva *Include*) que conté diversos arxius "**.conf*" el contingut dels quals, però, és equivalent al dels fitxers "**/etc/apache2/mods-enabled/*.load**" de l'Ubuntu (és a dir, contenen múltiples línies *LoadModule*, essent en aquest cas la carpeta "*/usr/lib64/httpd/modules*" la ubicació on es troben els mòduls pròpiament dits).

A Fedora no disposem de les comandes `a2{en|dis}mod/a2{en|dis}site/a2{en|dis}conf` que hi ha a Ubuntu per tal de activar/desactivar mòduls, "virtual hosts" i configuracions, respectivament. Per aconseguir el mateix objectiu, l'única manera és, respectivament descomentar/comentar a mà les línies adients dins del fitxer en qüestió (i tot seguit reiniciant/recarregant el servei).

A Fedora tampoc tenim la comanda `a2query` d'Ubuntu. Per saber els mòduls actualment carregats hem de fer servir llavors la comanda `httpd -M` Per saber els "virtual hosts" activats en aquest moments i els trossos de configuració extra activats en aquest moment no hi ha equivalent, però.

Ordre de preferència de les directives

Moltes directives de configuració del servidor Apache poden aparèixer escrites en diferents arxius de configuració a la vegada. Segons en quin arxiu hi siguin, tindran una afectació més o menys global. Així:

A Ubuntu
<p>*Les directives escrites dins l'arxiu "apache2.conf" o dins d'algun arxiu ".conf" ubicat dins de la subcarpeta "conf-available", afecten a tots els "virtual hosts" sense distinció. També afectaran globalment si apareixen dins d'algun arxiu "mods-available/*.conf", ja que els mòduls es carreguen (amb el corresponent arxiu "*.load") globalment.</p> <p>NOTA: Si apareix la mateixa directiva a l'arxiu "apache2.conf" i en un altre arxiu ubicat dins de la carpeta "/etc/apache2/conf-available/", s'aplicarà sempre l'existent en aquest darrer (això és degut a què aquests arxius s'inclouen mitjançant la línia <i>Include</i> al final de l'arxiu "apache2.conf", i <u>Apache sempre aplica la darrera directiva trobada</u> entre vèries iguals). Igualment, si apareix la mateixa directiva en arxius ubicats tots dins de la carpeta "/etc/apache2/conf-available/", s'aplicarà sempre la directiva que estigui escrita al darrer arxiu, ordenant-se aquests alfabèticament a partir del seu nom (el motiu és el mateix que l'explicat al paràgraf anterior)</p> <p>*Les directives escrites dins d'algun arxiu ".conf" ubicat dins de la subcarpeta "sites-available/" només afecten al "virtual host" en qüestió, ja qu en principi haurien d'anar escrites sempre dins d'una secció anomenada <VirtualHost> (o similar), que de seguida estudiarem</p> <p>*Les directives escrites dins de les seccions <Directory> o <File> (o similars) presents en qualsevol dels arxius anteriors (i que de seguida estudiarem) només afecten a aquella carpeta o fitxer en qüestió concret respectivament. Les directives escrites dins d'aquestes seccions sempre tindran preferència; cal tenir present aquest fet, per exemple, en establir directives de registre (<i>ErrorLog</i>, <i>CustomLog</i>,...), etc.</p>
A Fedora
<p>*Les directives escrites dins l'arxiu "httpd.conf" o dins d'algun arxiu ".conf" ubicat dins de la subcarpeta "/etc/httpd/conf.d", afecten a tots els "virtual hosts" sense distinció.</p> <p>NOTA: Si apareix la mateixa directiva a l'arxiu "httpd.conf" i en un altre arxiu ubicat dins de la carpeta "/etc/httpd/conf.d/", s'aplicarà sempre l'existent en aquest darrer (això és degut a què aquests arxius s'inclouen mitjançant la línia <i>Include</i> al final de l'arxiu "httpd.conf", i <u>Apache sempre aplica la darrera directiva trobada</u> entre vèries iguals). Igualment, si apareix la mateixa directiva en arxius ubicats tots dins de la carpeta "/etc/httpd/conf.d/", s'aplicarà sempre la directiva que estigui escrita al darrer arxiu, ordenant-se aquests alfabèticament a partir del seu nom (el motiu és el mateix que l'explicat al paràgraf anterior)</p> <p>*Les directives presents o bé en l'arxiu "httpd.conf" o bé en algun arxiu ".conf" ubicat dins de la subcarpeta "/etc/httpd/conf.d/" però que estiguin escrites dins d'una secció anomenada <VirtualHost> (o similar) només afecten al "virtual host" en qüestió</p> <p>*Les directives presents o bé en l'arxiu "httpd.conf" o bé en algun arxiu ".conf" ubicat dins de la subcarpeta "/etc/httpd/conf.d/" però que estiguin escrites dins de les seccions <Directory> o <File> (o similars) només afecten a aquella carpeta o fitxer en qüestió concret respectivament. Les directives escrites dins d'aquestes seccions sempre tindran preferència; cal tenir present aquest fet, per exemple, a l'hora d'establir directives de registre (<i>ErrorLog</i>, <i>CustomLog</i>, ...), entre altres casos.</p>

Per trobar d'un cop totes les línies que continguin una directiva concreta al llarg dels diferents arxius on poden aparèixer es pot executar la comanda `grep -ri "nomDirectiva" /etc/apache2/*` (a Ubuntu) o `grep -ri "nomDirectiva" /etc/httpd/*` (a Fedora)

Directives de configuració general

Al fitxer "apache2.conf" (a Ubuntu) o "httpd.conf" (a Fedora) hi ha unes quantes directives globals interessants que poden ser modificades si cal, com per exemple:

ServerRoot /etc/apache2 (o *ServerRoot /etc/httpd*): Aquesta línia no caldrà canviar-la quasi mai. Indica la ruta de la carpeta sota la qual es troben els arxius de configuració, registre i error del servidor. Qualsevol altra directiva que inclogui una ruta relativa com a valor (per exemple, *Include* o *LoadModule*, entre altres) prendrà la ruta de *ServerRoot* com el seu origen absolut

PidFile ..., User ..., Group ...: Aquestes línies apliquen diversos valors especificats a l'arxiu "envvars" (en el cas d'Ubuntu) o valors explícitament especificats allà mateix (en el cas de Fedora). La més important per nosaltres serà la directiva *User*, la qual, recordem, fa que, a pesar de què el servidor arrenqui sempre com a root (perquè necessita vincular-se al port nº80, port que, com tots els ports menors de 1024, és privilegiat i només el pot obrir l'usuari root) tots els processos fill del servidor adquireixin els permisos de l'usuari allà indicat. Aquest usuari, per tant, ha de poder llegir els directoris que volguem publicar i ha de ser el propietari del directori de logs. Per defecte, a Ubuntu aquest usuari s'anomena "**www-data**" i a Fedora, "**apache**".

Include ..., IncludeOptional ...: Ambdues línies serveixen per indicar la ruta (relativa a *ServerRoot*) d'un fitxer (o més, ja que admeten el comodí "*") el contingut del qual s'inclourà com a tros de configuració tal com si estigués directament escrit dins de l'arxiu "apache2.conf/httpd.conf". La diferència entre les dues directives és que *IncludeOptional* no genera cap error si l'arxiu o arxius indicats no es troben perquè la ruta indicada no existeix (però *Include* sí). En tot cas, ja hem comentat abans que aquestes directives són les que es fan servir per incloure les configuracions que hi ha repartides per la resta de fitxers (a Ubuntu, el "ports.conf" i tots els ".conf" ubicats dins de les carpetes "mods-enabled", "sites-enabled" i "conf-enabled" i a Fedora, tots els ".conf" ubicats dins de la carpeta "/etc/httpd/conf.d").

AccessFileName .htaccess: Indica el nom predeterminat (en aquest cas, ".htaccess", fixeuvos en el punt inicial -és un axiu ocult, per tant-) de tots els possibles "arxius d'accés". Sobre els arxius d'accés en parlarem a l'apartat d'aquest document que explica les seccions <Directory>.

Algunes de les directives que també podem trobar en "apache2.conf/httpd.conf" són de més baix nivell (més específicament, relacionades amb la gestió de connexions de xarxa a nivell de protocol TCP) i, per tant, no caldrà modificar el seu valor si realment no se sap què s'hi està modificant. Per exemple:

Timeout n°: Indica el n° de segons que el servidor esperarà sense rebre o transmetre cap dada abans de tancar la connexió

KeepAlive on/off: Indica si s'acceptaran (o no) connexions TCP persistents (és a dir, connexions TCP que, en comptes de encabir només una parella petició-resposta HTTP i tancar-se, permeten encabir diverses parelles peticions (provinents d'un mateix client)-respostes HTTP al llarg del temps gràcies a mantenir-se obertes més enllà de la primera parella petició-resposta HTTP processada). Les connexions persistents disminueixen la càrrega d'haver de crear una nova connexió a cada petició, cosa que la RAM (i la CPU) del servidor agraeix.

MaxKeepAliveRequests n°: Indica el n° màxim de peticions HTTP permeses dins una connexió TCP persistent ("0" vol dir il·limitat).

KeepAliveTimeOut n°: Indica el n° de segons que el servidor esperarà sense rebre cap petició des de la última rebuda d'un mateix client dins d'una connexió TCP persistent abans de tancar aquesta. Això simplement significa que la propera vegada que aquest client faci una nova petició, el servidor tornarà a crear una nova connexió TCP per respondre-la.

A l'arxiu "apache2.conf/httpd.conf" també hi podem trobar unes quantes altes directives que estan relacionades amb la gestió dels "logs" (i que explicarem en profunditat a l'apartat "Directives de logs"). D'altra banda, el fitxer "ports.conf" (a Ubuntu; a Fedora seguiria sent "httpd.conf") conté la directiva:

Listen [direccioIP:]port: Estableix la tarja de xarxa (amb la IP indicada) i port pel qual s'escoltaran peticions. Es poden escriure diverses línies *Listen*, però llavors caldrà tenir en compte la configuració dels diferents "virtual hosts" (veure més avall). Fixeu-vos que també poden aparèixer línies *Listen* dins de condicionals que comproven si un determinat mòdul està activat o no: això permet obrir un port només si el mòdul adequat (normalment relacionat amb les connexions segures TLS) està activat.

Als arxius que hi ha dins de la subcarpeta "conf-available" (a Ubuntu; a Fedora seguiria sent "httpd.conf") també podem trobar algunes directives interessants, com ara:

AddDefaultCharset UTF-8 : Estableix la codificació predefinida de les pàgines web a servir a UTF-8 (una evolució ampliada i millorada del vell estàndar ASCII) sobrescrivint llavors la que estigui especificada dins d'elles (amb l'etiqueta *<meta charset...>*, per exemple). Si aquesta línia està comentada, llavors prevaldrà la codificació indicada dins de la pròpia pàgina web.

ErrorDocument n°error { "text error" | /etc/apache2/fitxer | http://ruta/remota/resultat/dun/script } : Estableix el missatge de text que s'enviarà al client si el servidor detecta l'error HTTP indicat. En comptes del missatge de text també es pot establir la ruta d'un fitxer (llavors s'enviaria al client el seu contingut) o la URL d'un script -PHP per exemple- (llavors s'enviaria al client el resultat d'haver executat aquest script). Per exemple, si el n° d'error fos el 404, el document a retornar hauria de ser la pàgina mostrant "pàgina no trobada".

Configuració de "Virtual hosts"

La configuració d'un "virtual host" en particular (la qual, recordem, pot estar indicada en un fitxer ".conf" ubicat, a Ubuntu, dins de la subcarpeta "/etc/apache2/sites-available" i, a Fedora, ubicat dins de la subcarpeta "/etc/httpd/conf.d") ha d'estar envoltada sempre de l'etiqueta *<VirtualHost ip:port>* al principi i una etiqueta *</VirtualHost>* al final. Aquí "ip:port" indica, respectivament, la direcció IP concreta per on estarà accessible aquest "virtual host" (o bé el símbol "*" si volem que ho estigui a través de qualsevol IP que tingui el nostre servidor) i el port per on estarà escoltant (el qual sol ser el n°80 o bé el n°443 si és un servidor segur però, en tot cas, aquest port ha d'estar prèviament "activat" mitjançant la directiva *Listen*). En tot cas, un cop haguem establert la configuració d'un nou "virtual host", recordem també que, en el cas d'Ubuntu, a més caldrà activar-ho amb la comanda *a2ensite* (a Fedora això no cal) i, seguidament (això sí que cal fer-ho en tots els sistemes), reiniciar el servei Apache per tal que finalment es reconegui aquest nou "virtual host".

NOTA: Per desactivar un "virtual host", a Ubuntu ens seria suficient amb executar la comanda *a2dissite* (i reiniciar el servei Apache) mentre que a Fedora caldria eliminar (o comentar tot el seu contingut) l'arxiu ".conf" corresponent (i reiniciar igualment el servei Apache)

Entre les etiquetes *<VirtualHost ip:port>...</VirtualHost>* les directives imprescindibles que cal escriure per definir un "virtual host" són:

ServerName nom.DNS : Indica el nom DNS del "virtual host". Aquesta dada és la que distingeix un "virtual host" d'un altre que hi estigui definit dins del mateix servidor Apache.

NOTA: El valor d'aquesta directiva es comparà, cada cop que es rebí una petició HTTP d'un client, amb el valor d'una capçalera que incorpora aquesta anomenada "Host", la qual precisament serveix per saber a quin "virtual host" va dirigida la petició en qüestió

NOTA: Si no disposem d'un servidor DNS a la nostra infraestructura de xarxa (o bé no tenim contractat cap servidor DNS extern que apunti a la IP de la màquina on s'està executant l'Apache) no podrem tenir accessible el nostre servidor Apache a Internet. No obstant, sí que el podem fer servir dins de la nostra LAN: per això hauríem, per exemple, de configurar convenientment l'arxiu "/etc/hosts" a cadascun dels ordinadors (clients i servidors) presents a la nostra xarxa de forma que reconeguin mútuament els seus respectius noms.

ServerAlias unNomDiferent unAltre...: Indica un nom o noms DNS alternatiu/s a l'indicat a *ServerName* . D'aquesta manera, els clients podran accedir al "virtual host" fent servir diferents noms. És molt típic trobar que *ServerName* tingui el valor *www.domini.com* i *ServerAlias* tingui el valor *domini.com*, per exemple. En qualsevol cas, el servidor DNS hauria d'estar configurat per apuntar al servidor Apache també amb aquests noms alternatius. Aquesta directiva no és obligatòria.

DocumentRoot /ruta/carpeta : Indica la ruta de la carpeta arrel des d'on pengen les pàgines web (i la resta de documents) que serveix aquest "virtual host". Al "virtual host" definit per defecte en una instal·lació d'Apache (és a dir, a "000-default.conf" en Ubuntu o a "httpd.conf" directament en Fedora) aquesta carpeta és "/var/www/html". Això vol dir que la pàgina web que apareix quan

accedim a <http://127.0.0.1> (o <http://ip.dun.altre.servidor>) només havent instal·lat l'Apache es troba allà (concretament és la pàgina anomenada "index.html"). El mateix passaria si escrivíssim <http://127.0.0.1/index.html> (o <http://ip.dun.altre.servidor/index.html>) ja que "index.html" és el nom de pàgina web a mostrar que Apache pren per defecte si a la petició no s'especifica cap. Si volguéssim publicar qualsevol altre pàgina o fitxer (per exemple, un document anomenat "hola.pdf") només caldria copiar-lo dins d'aquesta carpeta: anant llavors a <http://ip.del.servidor.web/hola.pdf> ja el tindríem disponible.

NOTA: Dins de la carpeta arrel es poden crear tantes subcarpetes com es vulgui, les quals també s'hauran d'indicar convenientment a la URL demanada; per exemple, si creem una carpeta "/var/www/html/fotos" i allà copiem l'arxiu "platja1.png", per accedir a aquesta foto via navegador hauríem d'escriure a la seva barra de direccions la següent URL: <http://ip.del.servidor.web/fotos/platja1.png>. Com es pot comprovar, tot el que queda fora de "/var/www/html" és invisible al visitant: per això s'anomena "carpeta arrel".

NOTA: Per a què tot funcioni correctament, els permisos de la carpeta arrel i el de les seves subcarpetes han de permetre l'accés a l'usuari propi de l'Apache ("www-data" a Ubuntu, "apache" a Fedora). És a dir, han de tenir el permís d'execució "-x"- pel grup "d'altres" (ja que per defecte el propietari i el grup d'aquestes carpetes és l'usuari "root", tant a Ubuntu com a Fedora). D'altra banda, els permisos dels fitxers ubicats dins de la carpeta arrel i dins de les seves subcarpetes el que han de permetre és la lectura a l'usuari propi de l'Apache. És a dir, han de tenir el permís de lectura "-r"- pel grup d'altres. Aquestes configuracions ja estan ben establertes per defecte, però, en tot cas, recomano repassar les comandes *chmod* i *chown* per si fos necessari el seu ús en algun altre moment

Directives de logs

Altres directives comunes que poden definir-se a nivell global però que solen ser definides més sovint per cada "virtual host" en particular són les relacionades amb els missatges de registre d'errors...:

ErrorLog /ruta/error.log : Indica la ruta de l'arxiu on es guardaran els missatges dels errors que puguin ocórrer durant el funcionament del servidor. La configuració per defecte fa servir el fitxer **"/var/log/apache2/error.log"** (a Ubuntu) o **"/var/log/httpd/error_log"** (a Fedora).

NOTA: Per evitar que l'arxiu "error.log" vagi creixent de forma incontrolada a mida que es van emmagatzemant més i més missatges de registre, hi ha la possibilitat d'indicar a *ErrorLog* que es vol utilitzar un programa especial de rotació de logs. La idea és anar omplint el fitxer fins un límit (donat per temps o per tamany de fitxer), moment en el qual aquest fitxer es guarda comprimit i s'obre un nou fitxer on es comença un altre cop a emmagatzemar noves dades. Per evitar acabar amb un munt de fitxers comprimits, llavors s'estableix un màxim d'aquests -per exemple, 5-, de forma que en arribar al sisè, se sobrescriurà el fitxer comprimit més antic, i així successivament; per això es diu "rotació de logs". Encara que es poden fer servir altres (com LogRotate: <https://github.com/logrotate/logrotate>), l'Apache ja incorpora un programa propi de rotació de logs anomenat *rotatelogs* (<http://httpd.apache.org/docs/current/programs/rotatelogs.html>) el qual es pot invocar així :

ErrorLog "|bin/rotatelogs -n 5 /ruta/error.log 86400" on -n 5 indica que hi haurà 5 fitxers (l'actual i 4 comprimits) i 86400 indica l'interval de temps (en segons) cada quan es farà la compressió del fitxer actual -sobrescrivint el comprimit més antic en aquell moment- i l'obertura d'un nou fitxer log blanc.

ErrorLog "|bin/rotatelogs -n 5 /ruta/error.log 10M" on 10M indica el tamany que haurà d'aconseguir el fitxer de log actual per ser comprimit i obrir-ne un de nou. Es poden escriure els sufixes B -Bytes-, K -Kilobytes-, M -Megabytes- i G -Gigabytes-.

NOTA: A la directiva *ErrorLog* també es podria indicar, en comptes de la ruta d'un fitxer, la paraula "journald" (d'aquesta manera, es delega al sistema de logs de Systemd la recopilació i gestió dels missatges d'error) però això només és possible si s'usa el mòdul "mod_journald", disponible només a partir de la versió 2.5 d'Apache (http://httpd.apache.org/docs/trunk/mod/mod_journald.html)

LogLevel nivell : Indica el nivell mínim d'importància que els errors han de tenir per ser guardats a l'arxiu especificat a *ErrorLog*. Els nivells són (de menys a més importants): "debug", "info", "notice", "warn", "error", "crit", "alert" i "emerg". També es pot especificar un nivell mínim només per errors relacionats amb un determinat mòdul (o més) diferents del nivell mínim per defecte; això es fa indicant el nom del mòdul seguit de dos punts i el nivell, així per exemple: *LogLevel notice rewrite:info* (on s'estableix el nivell "notice" per tots els errors excepte els relacionats amb el mòdul "rewrite" -que ja estudiarem-, els quals han de tenir el nivell "info").

ErrorLogFormat "format" : Aquesta línia defineix quina informació es vol guardar a l'arxiu indicat a *ErrorLog* seguint el format especificat. Els indicadors de format possibles estan documentats a <http://httpd.apache.org/docs/current/mod/core.html#errorlogformat> . No obstant, aquesta directiva no se sol utilitzar gaire (de fet, per defecte no apareix escrita) perquè l'Apache ja incorpora un format predefinit pels missatges d'error que sol ser adequat a la majoria d'ocasions; concretament, consta dels següents camps delimitats per claudàtors: la data i hora del missatge, el nom del mòdul que produeix el missatge i el seu nivell d'importància (veure *LogLevel*), el PID del procés que ha experimentat l'error, la direcció del client que ha fet la petició i una frase detallada explicant l'error.

...i també amb els de registre d'events en general:

LogFormat "format" nomFormat : Defineix un format i li assigna un nom per poder-lo referenciar més endavant en una línia *CustomLog* (que és la que realment s'encarrega de guardar la informació en un arxiu de registre personalitzat, amb el format referenciat). Poden existir múltiples línies *LogFormat*, cadascuna assignant un determinat format a un nom. Un exemple de línia d'aquest tipus podria ser: *LogFormat "%h %u %t |"%r|" %>s %b %{User-agent}i" unformat* .Els indicadors de format possibles a http://httpd.apache.org/docs/current/mod/mod_log_config.html#formats estan documentats (i a <http://httpd.apache.org/docs/current/logs.html#accesslog> podem trobar-ne exemples) però tot seguit mostrem els valors típics:

%a : direcció IP del client
%h : nom del client
%H : el protocol de la petició (HTTP, HTTPS,...)
%m : el mètode de la petició (GET, POST,...)
%U : la url sencera demanada, sense querystring
%q : la querystring (el que va després del ?)
%r : primera línia de la petició (és a dir, la línia *<MÈTODE><URI><VERSIO>*)
%u : usuari remot (si es requereix protecció per contrasenya; veure més endavant)
{NomCapçaleraClient}i : valor de la capçalera de client (petició) indicada
%s : estat de la petició
{NomCapçaleraServidor}o : valor de la capçalera de servidor (resposta) indicada
%B : tamany del cos de la resposta. També pot servir *%b*
%t : l' hora en la què es va rebre la petició (en format anglès)
{format}t: l' hora en la què es va rebre la petició (en format strftime: <http://strftime.org>)
%D : temps trigat en servir la petició, en microsegons
%T : temps trigat en servir la petició, en segons

NOTA: Els modificadors *< i >* s'usen en peticions que poden ser diferents entre l'original i la final; per exemple, *%>s* es pot usar per guardar l'estat final de la petició

NOTA: El noms dels clients (*%h*) el servidor només els pot conèixer si a partir de la IP del client en qüestió (obtinguda de la petició feta per aquest) en fa una petició DNS inversa a algun servidor DNS que tingui configurat. Aquest procés, activat (o no) per la directiva de configuració del servidor *HostnameLookups* segons si valgui *On* o *Off* respectivament, és molt costós en recursos, així que, per tant, està desactivat per defecte (i, per tant, el valor *%h* no se sol obtenir (apareix el símbol "-" en el seu lloc).

CustomLog /ruta/fitxer "format" ó CustomLog /ruta/fitxer nomFormat : Registra informació escollida amb un determinat format (definit directament fent servir les mateixes opcions de la llista anterior *-%r, %s, ...-* o bé indicant el seu nom si ja ha sigut prèviament definit i batejat en alguna línia *LogFormat* anterior). Pot haver-hi múltiples línies *CustomLog* . Per defecte l'Apache té definit una línia *CustomLog* que genera un arxiu anomenat *"/var/log/apache2/access.log"* (a Ubuntu) o *"/var/log/httpd/access_log"* (a Fedora) amb la informació més típica que podem esperar d'un arxiu de registre de peticions i respostes HTTP; aquest fitxer serà el que més sovint consultem quan volguem saber l'activitat del nostre servidor quina és.

NOTA: A la directiva *CustomLog* també es podria indicar, en comptes de la ruta d'un fitxer:

- La invocació a un programa extern de gestió de logs, com *rotatelog*: *"|bin/rotatelog -n n° /ruta/fitxer xxxx"*.
- La paraula "journald"; d'aquesta manera es delegaria al sistema de logs de Systemd la recopilació i gestió dels missatges. Això, però, només és possible si s'usa el mòdul "mod_journald" , disponible a partir de la versió 2.5

NOTA: Existeixen múltiples programes que permeten visualitzar d'una forma gràfica i entenedora les dades emmagatzemades en els fitxers de registre que se li indiquin (com ara l'"access.log"). Alguns exemples són: Awstats (<https://github.com/eldy/awstats>), Analog (<https://github.com/c-amie/analog-ce>) o GoAccess (<https://github.com/allinurl/goaccess>), entre altres. Els veurem als exercicis.

NOTA: Una manera ràpida de veure en temps real els nous missatges que es van emmagatzemant a un determinat arxíu log (això també val per l'arxíu "error.log") és executar en un terminal la comanda `tail -f /ruta/fitxer`

Exemple

A continuació es presenta un exemple d'arxíu de configuració ".conf" (a ubicar dins de la carpeta "sites-available/" -Ubuntu- o de "conf.d" -Fedora-) on es defineixen dos "virtual hosts" d'un servidor Apache (amb una única IP i port escoltant), cadascun d'ells identificat per un nom DNS diferent ("www.hola.com" i "www.hola.org"), noms convenientment establerts prèviament per a què apuntin, en ambdós casos, a la mateixa màquina (IP):

```
# Això és un comentari: cal assegurar-se que existeixi una línia Listen 80 (a "ports.conf", normalment)
<VirtualHost *:80>
    ServerName www.hola.com
    DocumentRoot "/var/www/html/web1"
    #Altres directives, com ara ErrorLog, CustomLog o seccions <Directory>, <Files>, etc
</VirtualHost>
<VirtualHost *:80>
    ServerName www.hola.org
    DocumentRoot "/var/www/html/web2"
    #Altres directives, com ara ErrorLog, CustomLog o seccions <Directory>, <Files>, etc
</VirtualHost>
```

L'exemple anterior emprava un tipus concret de "virtual host", el "basat en nom". Però en realitat existeixen dos tipus de "virtual hosts": els "basats en nom" i els "basats en IP". En la gran majoria de casos s'empra els "virtual hosts" basats en nom (com el de l'exemple anterior) perquè són més convenients, ja que un "virtual host" d'aquest tipus pot compartir una mateixa direcció IP i port amb altres "virtual hosts" del mateix tipus (és a dir, tots poden funcionar sense trepitjar-se rera el port 80 d'una sola tarja de xarxa) ja que el que els distingeix entre sí és, tal com ja hem comentat, el nom especificat a la directiva `ServerName`, que és el nom que les peticions dels clients hauran d'incloure a la seva capçalera `Host`: per poder ser redirigides al "virtual host" pertinent. Els "virtual hosts" basats en IP, però, es distingeixen entre sí per tenir associada una direcció IP i/o un port d'escolta diferents, de manera que les peticions dels clients són redireccionades al "virtual host" adequat segon la IP i/o port a la què van dirigides (és a dir, serveixen per a què el servidor Apache amb una sola IP però escoltant en diferents ports o bé amb diferents IPs i/o ports associats cadascun a una tarja de xarxa diferent pugui oferir un lloc web diferent per cada combinació IP/port diferent). Així doncs, quan arriba al servidor Apache la petició d'un document, aquest esbrina pel nom indicat (en el cas de "virtual hosts" basats en nom) o per la IP i port de destí (en el cas de "virtual hosts" basats en IP) quin `DocumentRoot` (entre altres configuracions) ha d'emprar. A continuació es mostren un parell d'exemples de configuració de "virtual hosts" basats en IP: el de l'esquerra representa dos "virtual hosts" basats en IPs diferents (és a dir: estem suposant que el servidor Apache disposa de 2 targetes de xarxa, cadascuna amb una IP diferent - 172.20.30.40 i 172.20.30.50- associades cadascuna a un nom DNS diferent, i oferint un lloc web diferent també), mentre que el de la dreta representa dos "virtual hosts" basats en IPs iguals però ports diferents (és a dir, estem suposant que el servidor Apache disposa d'una tarja de xarxa només -amb una IP associada a un nom DNS- però pot atendre peticions per dos ports diferents, cadascun associat a un "virtual host" -i per tant, un lloc web- també diferent).

```
# Cal que existeixi una línia Listen 80
#(a "ports.conf" -Ubuntu- o "httpd.conf" -Fedora-)
<VirtualHost 172.20.30.40:80>
    ServerName www.hola.com
    DocumentRoot "/var/www/html/web1"
</VirtualHost>
<VirtualHost 172.20.30.50:80>
    ServerName www.hola.org
    DocumentRoot "/var/www/html/web2"
</VirtualHost>
```

```
# Cal que existeixi una línia Listen 80 i una altra Listen 8080
#(a "ports.conf" -Ubuntu- o "httpd.conf" -Fedora-)
<VirtualHost 172.20.30.40:80>
    ServerName www.hola.com
    DocumentRoot "/var/www/html/web1"
</VirtualHost>
<VirtualHost 172.20.30.40:8080>
    ServerName www.hola.com
    DocumentRoot "/var/www/html/web2"
</VirtualHost>
```

EXERCICIS:

1.-Instal·la el paquet "httpd" (a Fedora) o "apache2" (a Ubuntu) en una màquina virtual que tingui la seva tarja de xarxa en mode "adaptador pont" (per a què sigui accessible des de l'exterior). A partir d'aquí, llegeix el següent paràgraf blau i tot seguit contesta les preguntes:

L'executable de l'Apache a Fedora és `/usr/sbin/httpd` (i així és invocat, amb els seus paràmetres pertinents, a través de la directiva `ExecStart=` quan s'executa la comanda `systemctl start httpd`) mentre que a Ubuntu s'anomena `/usr/sbin/apache2` (però en aquest cas la comanda `systemctl start apache2` invoca, mitjançant la directiva `ExecStart=`, no el binari directament sinó un shell script que li fa de "wrapper" anomenat `apachectl` -el qual, de fet, també existeix a Fedora perquè aquest script està inclòs dins del propi paquet del servidor web-, que estableix el valor de certes variables d'entorn necessàries i admet certs paràmetres propis, a més de -en aquest cas només a sistemes Ubuntu- poder traspasar directament al binari real els paràmetres del propi binari).

- a) ¿Quina informació mostra la comanda `{httpd|apachectl} -t -D DUMP_MODULES` ?
- b) ¿Quina informació mostra la comanda `{httpd|apachectl} -t -D DUMP_INCLUDES` ?
- c) ¿Quina informació mostra la comanda `{httpd|apachectl} -t -D DUMP_VHOSTS` ?
- c) ¿Quina informació mostra la comanda `{httpd|apachectl} -t -D DUMP_RUN_CFG` ?
- d) ¿Quina informació mostra `{httpd|apachectl} -t` ? ¿I `{httpd|apachectl} -v` ? ¿I `{httpd|apachectl} -V` ?
- e) ¿Per a què serveix el paràmetre `-C` de la comanda `{httpd|apachectl}` ? ¿I el paràmetre `-c`? ¿I `-d`? ¿I `-f`? (pots consultar o bé la pàgina del manual del binari o bé executar-lo amb el paràmetre `-h` per obtenir una ajuda resumida)

2.-a) Executa la comanda `grep -r "^ServerRoot" /etc/httpd` (a Fedora) o `grep -r "^ServerRoot" /etc/apache2` (a Ubuntu) per esbrinar quin és el valor concret d'aquesta directiva. Consulta els apunts, els comentaris del propi arxiu de configuració on es troba indicada o la referència oficial, disponible a <https://httpd.apache.org/docs/2.4/mod/directives.html> , per esbrinar quin és el significat d'aquesta directiva.

b) Executa la comanda `grep -r "^ServerAdmin" /etc/httpd` (a Fedora) o `grep -r "^ServerAdmin" /etc/apache2` (a Ubuntu) per esbrinar quin és el valor concret d'aquesta directiva. Consulta els apunts, els comentaris del propi arxiu de configuració on es troba indicada o la referència oficial, disponible a <https://httpd.apache.org/docs/2.4/mod/directives.html> , per esbrinar quin és el significat d'aquesta directiva.

c) Executa la comanda `grep -r "^User" /etc/httpd` (a Fedora) o `grep -r "^User" /etc/apache2` (a Ubuntu) per esbrinar quin és el valor concret d'aquesta directiva. Consulta els apunts, els comentaris del propi arxiu de configuració on es troba indicada o la referència oficial, disponible a <https://httpd.apache.org/docs/2.4/mod/directives.html> , per esbrinar quin és el significat d'aquesta directiva.

NOTA: Una altra pàgina de referència per directives útil és <https://httpd.apache.org/docs/2.4/mod/quickreference.html>

3.-Fes el necessari per a què el servidor escolti al port 81(i continui funcionant el VirtualHost que hi ha definit per defecte a una instal·lació estàndar del paquet Apache). Comprova-ho. Desfés el canvi després.

4.-a) Crea un arxiu anomenat "pepe.com.conf" i un altre anomenat "manolo.com.conf", ambdós dins de la carpeta "/etc/httpd/conf.d" (a Fedora) o "/etc/apache2/sites-available" (a Ubuntu) amb el següent contingut respectiu, i digues, abans de provar res, per a què creus que serviran. ¿Què fan cadascuna de les directives que hi ha allà indicades, amb el valor concret que tenen assignat?

<pre><VirtualHost *:80> ServerName pepe.com ServerAlias *.pepe.com DocumentRoot /var/www/pepe.com ErrorDocument 404 "T'has equivocat de lloc" LogFormat "%a %U %{Host}i %t" pepelog CustomLog /var/log/access.pepe.com.log pepelog ErrorLog /var/log/error.pepe.com.log LogLevel debug </VirtualHost></pre>	<pre><VirtualHost *:80> ServerName manolo.com ServerAlias *.manolo.com DocumentRoot /var/www/manolo.com ErrorDocument 404 "T'has equivocat de lloc" LogFormat "%m %q %{Server}o %b" manolog CustomLog /var/log/access.manolo.com.log manolog ErrorLog /var/log/error.manolo.com.log LogLevel info </VirtualHost></pre>
---	--

NOTA: Les úniques directives realment imprescindibles dins de la definició d'un VirtualHost són només *ServerName* i *DocumentRoot*; la resta, o bé no cal definir-les (*ServerAlias*) o bé, que serien la resta, ja estan definides amb un valor per defecte en els arxius de configuració general (tal com veurem en un apartat d'aquest exercici)

b) Realitza les següents tasques preparatòries:

*Crea (com a root) les carpetes "/var/www/pepe.com" i "/var/www/manolo.com" (com a root).

NOTA: Opcionalment, a aquestes carpetes se'ls podria assignar com a propietari l'usuari actual (així: *sudo chown -R \$USER:\$USER /var/www/pepe.com*", per xemple) per tal de no haver de fer el pas següent també com a "root"

*Crea (com a root) dins de la primera carpeta anterior un arxiu anomenat "index.html" amb el contingut `<html><body>Hola, sóc Pepe</body></html>` i dins de la segona carpeta un altre arxiu anomenat igual amb el contingut `<html><body>Hola, sóc Manolo</body></html>`

*(Només a Ubuntu): Executa (com a root) la comanda *a2ensite pepe.com manolo.com*

*Reinicia el servei Apache

La configuració del servidor web ja estaria feta. No obstant, en no tenir implementat cap servidor DNS que associï els noms "pepe.com" i "manolo.com" a la IP d'aquest servidor web, haurem de fer un "truc brut" a totes les màquines clients que vulguin fer peticions a algun d'aquests dos dominis, el qual consisteix en modificar el seu respectiu arxiu "/etc/hosts" local per tal d'associar així aquests dominis amb la IP de la màquina on està funcionant l'Apache. Concretament, cal seguir les instruccions del proper apartat

c) A la mateixa màquina virtual (o en una altra de diferent, si vols), afegeix les següents línies a l'arxiu "/etc/hosts"...

```
ip.maq.serv.Apache pepe.com www.pepe.com
ip.maq.serv.Apache manolo.com www.manolo.com
```

...i tot seguit comprova que reps resposta fent *ping pepe.com* i *ping manolo.com*

d) Fes les següents comprovacions per veure que la configuració de l'Apache funciona correctament:

*Obre un navegador a la màquina on has modificat l'arxiu "/etc/hosts" anterior (o, si no té entorn gràfic, executa la comanda *curl*) i vés a la URL <http://pepe.com> ¿Què veus i per què? ¿I si vas a la URL <http://manolo.com>? ¿I si vas a <http://www.pepe.com>? ¿I a <http://www.manolo.com>? ¿I a <http://lalala.pepe.com>?

*Ara vés a la URL <http://www.pepe.com/pag.html> ¿Què veus i per què? ¿I si vas a la URL <http://www.manolo.com/pag.html>?

*¿Quin és el contingut dels arxius "/var/log/error.pepe.log" i "/var/log/error.manolo.log"? ¿Per què en un arxiu apareix més informació que en l'altre?

*¿Quin és el contingut dels arxius `"/var/log/access.pepe.log"` i `"/var/log/access.manolo.log"`? ¿Per què les línies mostren camps diferents?

e) Observa la sortida de la comanda `grep -r "ErrorDocument" /etc/httpd` (a Fedora) o `grep -r "ErrorDocument" /etc/apache2` (a Ubuntu) i dedueix quines pàgines d'error es mostraran segons el context

eII) Observa la sortida de la comanda `grep -r "CustomLog" /etc/httpd` (a Fedora) o `grep -r "CustomLog" /etc/apache2` (a Ubuntu) i dedueix quins arxius de registre s'utilitzaran segons el context

NOTA: Per saber els formats predefinits de línies de registre disponibles per utilitzar a la directiva `CustomLog` es pot consultar la sortida de `grep -r "LogFormat" /etc/httpd` (a Fedora) o de `grep -r "LogFormat" /etc/apache2` (a Ubuntu)

f) Desactiva el `VirtualHost` per defecte per tal que el contingut de la carpeta `"/var/www/html"` no sigui servit quan es faci una petició directament sobre la IP del servidor web (és a dir, sense indicar cap domini). Un cop fet, comprova que, efectivament, el servidor no respongui a peticions que no estiguin dirigides als dominis configurats als `VirtualHosts` `"pepe.com"` i `"manolo.com"`

NOTA: La manera de fer-ho a Ubuntu és simplement executant la comanda `sudo a2dissite 000-default`. Per fer-ho a Fedora caldrà comentar les línies `ServerName` i `DocumentRoot` indicades dins de l'arxiu `"httpd.conf"` (o bé crear un nou `VirtualHost` el `DocumentRoot` del qual apunti a la mateixa carpeta que la del per defecte, `"/var/www/html"`).

fII) La configuració general (en el cas de Fedora) o del `VirtualHost` per defecte (en el cas d'Ubuntu) té associada una línia `CustomLog` que a Fedora té un valor d'una ruta relativa i a Ubuntu incorpora el valor de la variable d'entorn `$APACHE_LOG_DIR`. ¿Quina és, doncs, la ruta absoluta corresponent per l'arxiu en qüestió (anomenat `"access_log"` a Fedora i `"access.log"` a Ubuntu)?