

A2: Serveis Web i transferència de fitxers



Referències:

https://ioc.xtec.cat/materials/FP/Recursos/fp_asx_m08_/web/fp_asx_m08_htmlindex/index.html

<http://acacha.org/mediawiki/index.php/Apache>

<http://debian-handbook.info/browse/es-ES/stable/sect.http-web-server.html>

<http://httpd.apache.org>

https://es.opensuse.org/SDB:Configuraci%C3%B3n_archivos_Apache

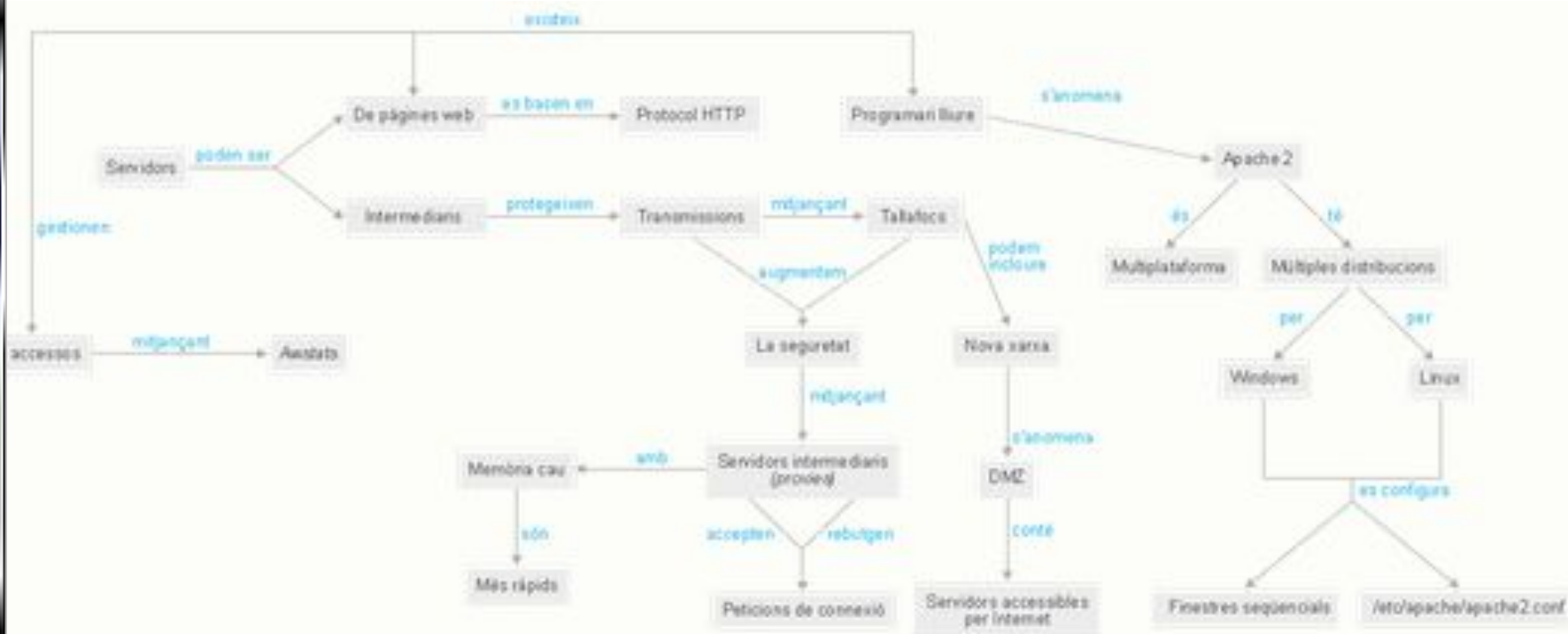
<http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/s1-apache-config.html>

http://informatica.iessanclemente.net/manuais/index.php/Instalaci%C3%B3n_de_Servidor_Web_Apache_en_Debian

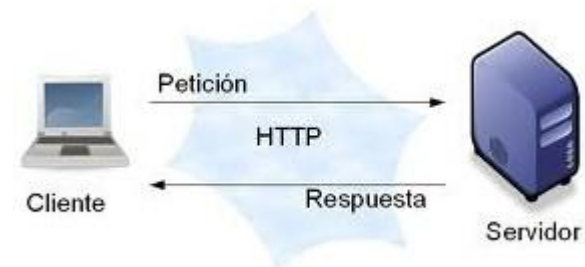
http://sopa.dis.ulpgc.es/ii-aso/portal_aso/leclinux/administracion/apache/apache_t.pdf

0. Introducció

En la unitat “**Servidors web i intermediaris**”, es mostra com l’administrador pot posar en funcionament i mantenir actualitzats, d’una banda, **servidors web per publicar continguts a Internet**, i de l’altra, **servidors proxy per optimitzar l’accés a xarxes públiques**.



0. Introducció



Les **pàgines web**, en la seva majoria en **format HTML (Hyper Text Markup language)**, requereixen ser **allotjades en màquines que disposin d'espai en disc** per emmagatzemar els arxius HTML, imatges, vídeo ... en directoris específics i **d'un servidor web per poder oferir-les als clients** (navegadors, això sempre que no les visualitzem en local)

Davant els perills d'Internet **cal incorporar protocols de comunicació segura** com HTTPS que fa servir xifrat SSL (secure sockets layer)

Els **requeriments més rellevants dels servidors són l'espai de disc i una bona connexió de xarxa** perquè el consum de la CPU sigui prou baix. **Tenen consums de recursos puntuals** perquè podem estar un temps sense peticions i, de cop i volta, tenir una allau de peticions. Això fa que els servidors web **acostumin a tenir un nombre baix de processos en espera**. A mesura que esdevenen necessaris, se'n van arrencant de nous (**inetd, xinetd**).

No totes les peticions consumeixen el mateix. Aquelles webs que executin programes d'interacció amb l'usuari o requereixen xifrat (HTTPS) consumeixen més recursos que altres webs amb menys interacció.

1. Servidor Web



Actualment **un dels serveis** més difosos i de gran utilitat és el **www** (world wide web). No és més que **un ampli i divers conjunt de documents** que es **distribueixen per tot el món mitjançant la xarxa, Internet**.

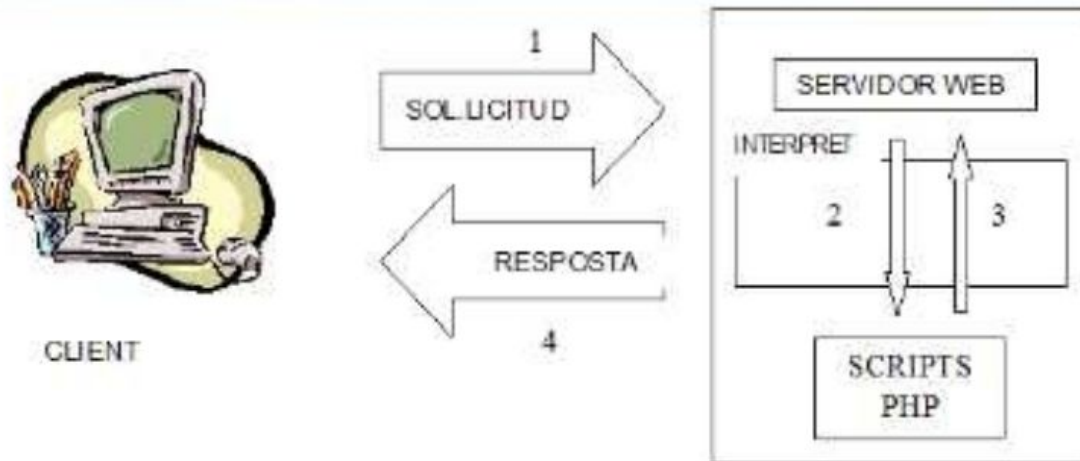
Un servidor web és “l'equip” on està instal·lat el **software** (aquest és realment qui dona servei) **que permet l'accés a un conjunt de webs**.

Els servidors poden ser locals (es troben a dins de la nostra LAN, formen part de la Intranet, normalment la BW no és problema) **o remots** (no es troben físicament a la nostra xarxa)

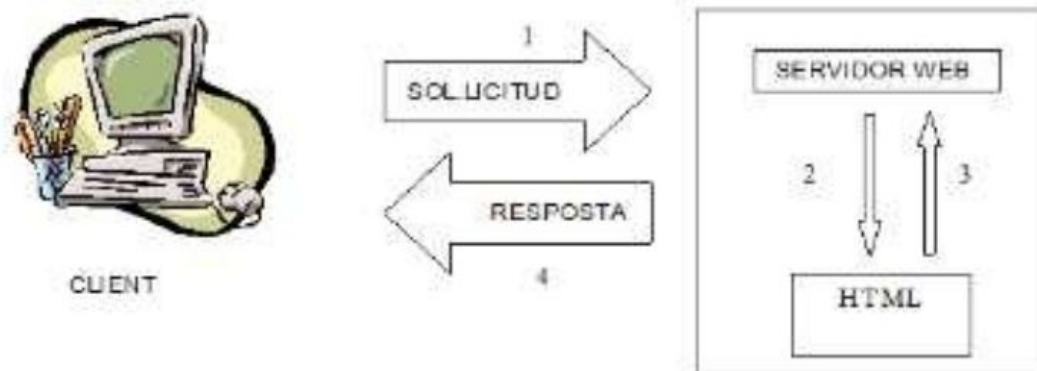
Al començament totes les **webs** eren **estàtiques** (fetes, normalment, només amb HTML i **enfocades a mostrar una informació, sense més interacció amb l'usuari, excepte els links**). A l'actualitat gairebé totes són **dinàmiques** amb el que són necessaris altres llenguatges com PHP, ASP, Aquestes **permeten interactivitat amb l'usuari** com per exemple omplir formularis, comandes on-line ...

1. Servidor Web

Webs dinàmiques



SERVIDOR WEB. PÀGINES ESTÀTIQUES



1. Servidor Web

Webs dinàmiques al servidor

Con totes les webs s'interpreten a la nostre màquina mitjançant un navegador. El servidor és qui executa el contingut dinàmic amb les dades enviades pel client (java, javascript, flash ...) i proporciona el codi generat (HTML) al client mitjançant HTTP. El client no pot veure el codi ni accedir a la BD. És més segur i descarrega al client Els llenguatges per webs dinàmiques és molt extens, potser els més coneguts són PHP, ASP, CGI, JSP, Perl ...



1. Servidor Web

Webs dinàmiques al client

Són **executades** amb un script **al client** amb llenguatges com Javascript (interpretat), VBScript (interpretat), Java al client (compilat-interpretat), Scripts de Flash, **generant la visualització de la web**.

El codi es pot veure i modificar, ja que va incrustat (embeded) a la web, enviant al servidor la resposta modificada.

Normalment es fa servir per a tasques de comprovació abans de l'enviament de dades (a un formulari que el correu enviat porti una @ i després un punt)

Lenguaje	Fecha de primera versión estable	Sistema operativo	Última versión estable
PHP	1995	Multiplataforma	5.3.5
ASP.Net	1998	Windows (Algunas versiones)	4.0
Perl	1987	Multiplataforma	5.12.3
Python	1991	Multiplataforma	3.2.0
Ruby	1995	Multiplataforma	1.9.3-p125

http	80	tcp	World Wide Web HTTP
http	80	udp	World Wide Web HTTP

1. Servidor Web

Per norma general els servidors web escolten pel **port 80** (**HTTP** protocol de transferència d'hipertext - HyperText Transfer Protocol) esperant rebre sol·licituds dels clients. HTTP és el protocol que fan servir el client i el servidor per comunicar-se. També es fa servir el **HTTPS** pel port **443**.

Per accedir a una web es fa servir una URL (Uniform Resource Locator), és l'adreça que indica la localització exacta (IP o nom. També es pot indicar el port i un login i password als documents que requereixin autenticació per ser accedits)

https	443	tcp	http protocol over TLS/SSL
https	443	udp	http protocol over TLS/SSL
https	443	sctp	HTTPS

Esquema://màquina/directori/fitxer

Esquema://usuari:contrasenya@màquina:port/directori/fitxer

http://es.wikipedia.org/wiki/Localizador_uniforme_de_recursos

<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>


```
telnet://[usuario[:contraseña]@]servidor[:puerto][/]
```

```
ftp://[usuario[:contraseña]@]servidor/camino
```

1. Servidor Web

```
news: identificador
```

Diferència entre URI (Uniform Resource Identifier) i URL

URI absolut:

```
esquema: identificador
```

```
mailto: dirección
```

La paraula que es troba **abans del separador ":"** és l'**esquema**, que **determina la sintaxi** de l'identificador. Aquesta **sintaxi** també pot permetre l'especificació d'**URI relatius**, en què **s'omet l'esquema i el separador ":"**.

Un URI pot ser un localitzador (URL), si especifica com s'accedeix al recurs, i/o un nom (URN) si identifica el recurs per mitjà d'un conjunt d'atributs. Algunes sintaxis URL típiques són ftp:, news:, mailto:, telnet: i http:.

```
URL-HTTP = URL-HTTP-absoluto | URL-HTTP-relativo
URL-HTTP-absoluto = http://servidor[:puerto]
[camino-absoluto]
URL-HTTP-relativo = camino-absoluto | camino-relativo
camino-absoluto = /camino-relativo
camino-relativo = [camino] * (;parámetro)[? consulta]
[# fragmento]
camino = segmento * (/segmento)
```

1. Servidor Web

Nota

A continuación, presentamos diferentes ejemplos de URL con esquemas diferentes (lo dos últimos son URL HTTP relativos):

`ftp://ftp.uoc.es/pub/doc/README`

`news:comp.infosystems.www.misc`

`mailto:Ernest.Udiant@campus.uoc.edu`

`http://www.uoc.es/`

`http://www.acme.com/%7Eadmin/`

`http://www.acme.com/buscador/busca.cgi?nom=Internet`

`http://www.acme.com/doc/ayuda.html#buscador`

`/doc/ayuda.html#buscador`

`ayuda.html#buscador`

1. Servidor Web

Nota

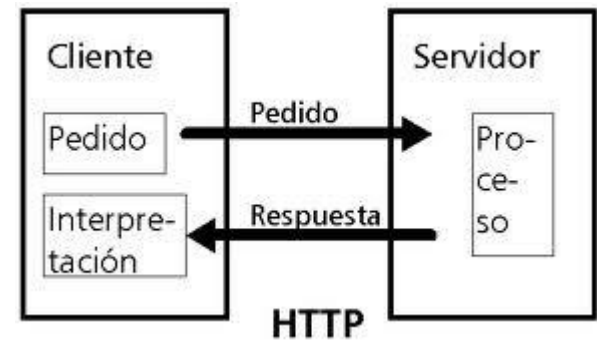
Interpretación del URL relativo

Dada la dirección base `http://www.uoc.edu/extern/ct/home/home.html`, los URL relativos siguientes deberían interpretarse de este modo:

Tabla 16.

URL relativo	URL absoluto
<code>/accesset98/</code>	<code>http://www.uoc.edu/accesset98/</code>
<code>otras/pagina.html</code>	<code>http://www.uoc.edu/extern/ct/home/otras/pagina.html</code>
<code>../../logo.jpeg</code>	<code>http://www.uoc.edu/extern/logo.jpeg</code>
<code>#final</code>	<code>http://www.uoc.edu/extern/ct/home/home.html#final</code>

1. Servidor Web



Quan el client fa una petició al servidor envia un missatge HTTP. Això el fa mitjançant unes capçaleres de petició que indicaran la comanda (GET, POST ...) i informació del client (qui és el client, formats d'arxius que suporta ...) que pot fer servir el servidor per saber com pot respondre a aquest client en particular.

Quan el servidor rep aquesta petició respon amb una capçalera de servidor amb informació del servidor, la quantitat i tipus de dades que transmetrà.

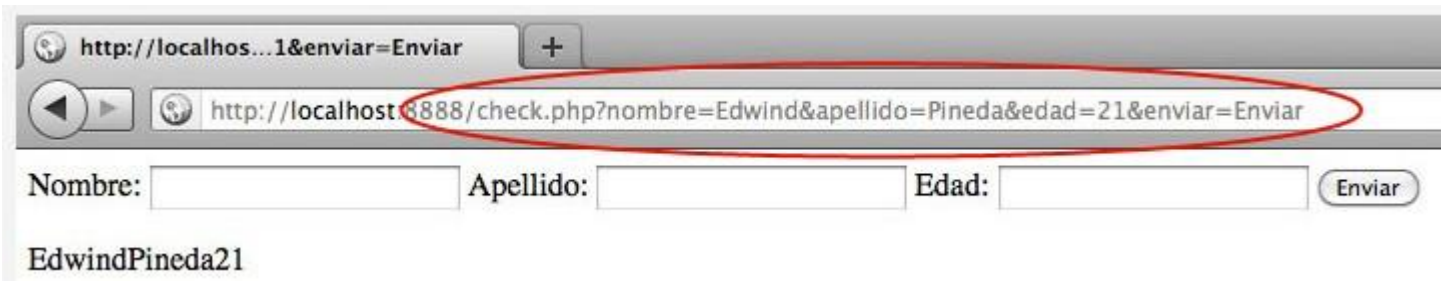
Una vegada establerta la connexió, el protocol TCP s'encarrega de mantenir la comunicació i garantir un intercanvi de dades lliure d'errors.

http://www.cicei.ulpgc.es/ocon/gsi/tut_tcpip/3376c426.html
<http://web.tursos.com/php-desde-cero-metodos-post-get/>

1. Servidor Web

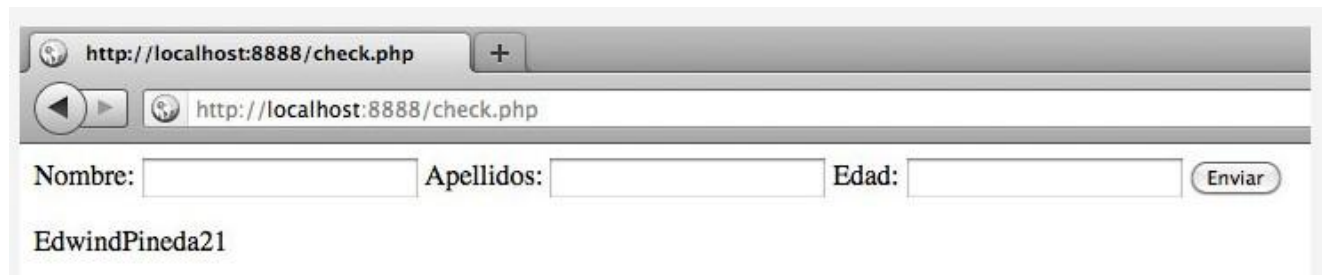
En el cas del protocol HTTP/0.9 només es permet el mètode GET, amb el protocol HTTP/1.0 GET, POST i HEAD i amb el **protocol HTTP/1.1** OPTIONS, GET, HEAD, POST, PUT, DELETE i TRACE.

Comanda GET. S'utilitza per recollir qualsevol tipus d'informació del servidor. El client també pot enviar però serà visible



Comanda POST. S'utilitza per enviar informació al servidor com, per exemple, les dades contingudes en un formulari, enviar un missatge a un grup d'usuaris

...



1. Servidor Web

Exemples de servidores Web

Software [\[editar\]](#)

Algunos servidores web importantes son:

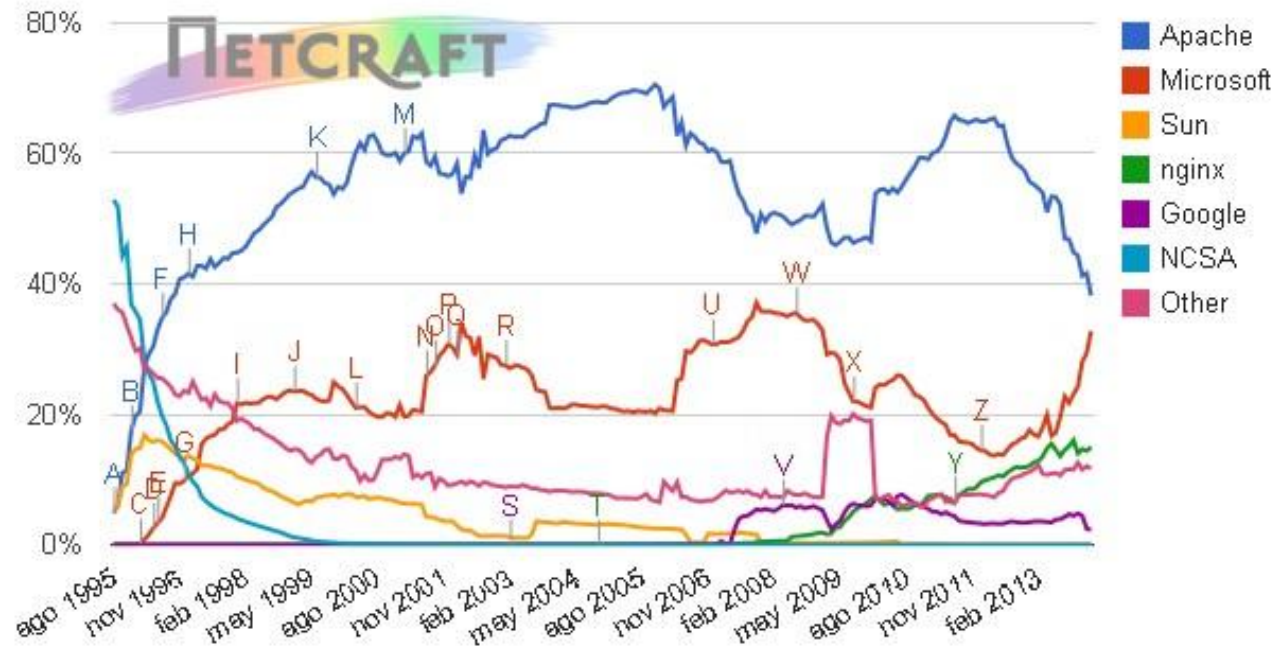
- *Nginx*
- *Apache*
- *Internet Information Services (IIS)*
- *Cherokee*
- *Tomcat*

Otros servidores, más simples pero más rápidos, son:

- *lighttpd*
- *thttpd*

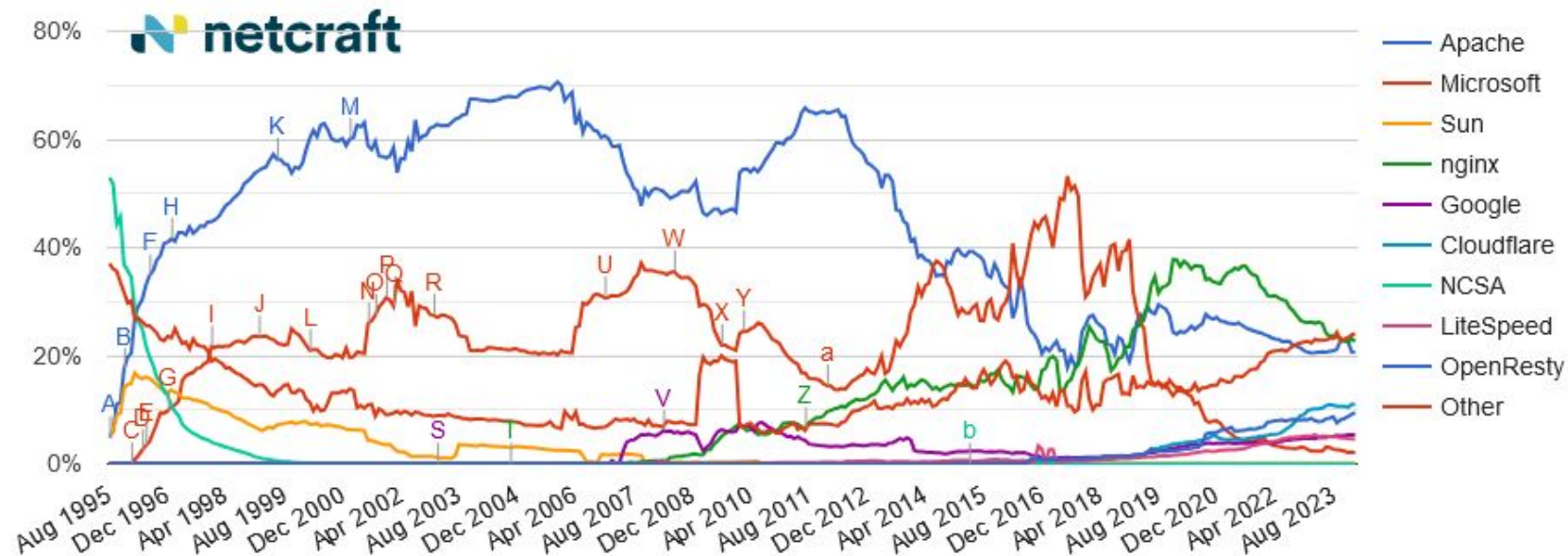
1. Servidor Web

Web server developers: Market share of all sites



Developer	January 2014	Percent	February 2014	Percent	Change
Apache	358,669,012	41.64%	351,700,572	38.22%	-3.41
Microsoft	253,438,493	29.42%	301,781,997	32.80%	3.38
nginx	124,052,996	14.40%	138,056,444	15.00%	0.60
Google	21,280,639	2.47%	21,129,509	2.30%	-0.17

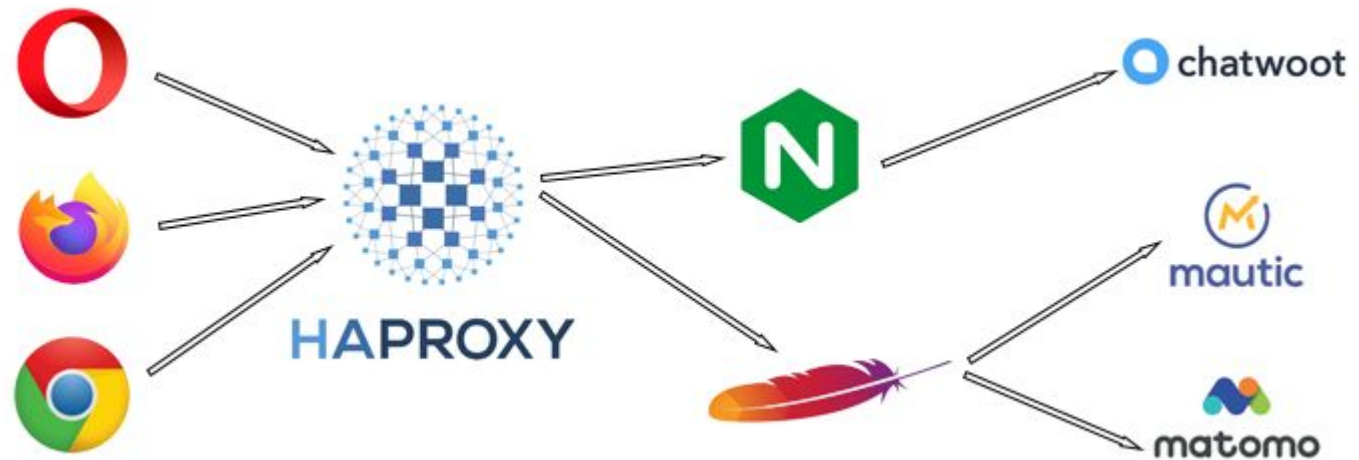
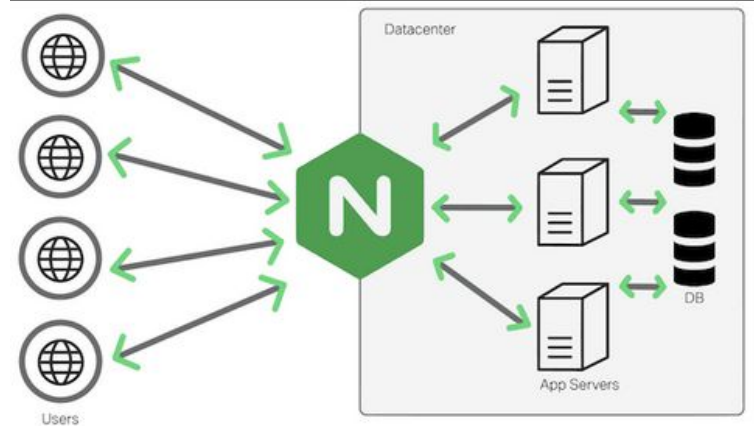
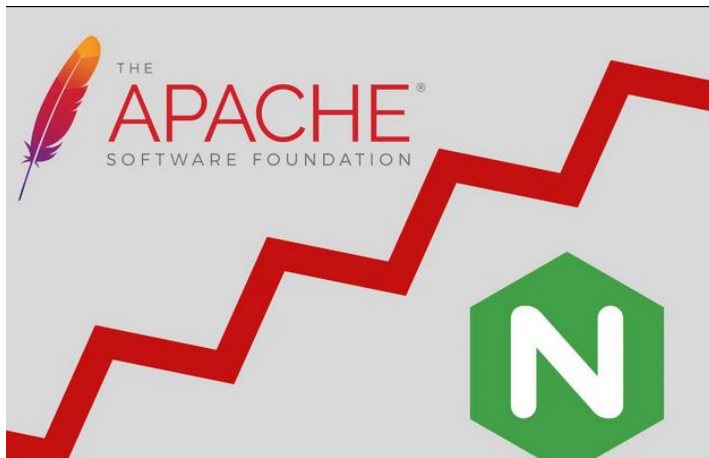
Web server developers: Market share of all sites



Developer	January 2024	Percent	February 2024	Percent	Change
nginx	250,509,224	23.21%	246,113,438	22.64%	-0.57
Apache	223,346,407	20.70%	224,808,405	20.68%	-0.01
Cloudflare	118,627,424	10.99%	120,502,966	11.09%	0.09
OpenResty	98,390,136	9.12%	103,106,166	9.49%	0.37



<https://www.netcraft.com/blog/february-2024-web-server-survey/>



Nginx (**pronunciat “Engine X”**) és un servidor proxy invers i HTTP, un servidor proxy de correu i un servidor proxy TCP/UDP genèric que s'utilitza per allotjar llocs web i aplicacions de totes les mides. Comprat per F5 Networks al 2019

Com triar entre Nginx i Apache

No hi ha un guanyador clar entre aquests gegants de servidors web. **Depèn principalment del que necessiti que administri el servidor web.**

Trieu Apache Si:

Sistemes heretats i compatibilitat: opció preferida per a sistemes que ho han estat utilitzant durant molt de temps. Si la vostra infraestructura depèn en gran mesura de .htaccess, cosa que permet una configuració descentralitzada i flexibilitat en entorns d'allotjament compartit.

Processament de contingut dinàmic: les vostres aplicacions requereixen un processament intensiu de contingut dinàmic, com PHP, Python o Ruby, Apache integra processadors directament dins dels seus mòduls. Això simplifica l'execució d'scripts dinàmics al servidor.

Facilitat d'ús i documentació: Apache té una gran quantitat de documentació i suport de la comunitat. És més fàcil trobar guies, tutorials i suport per a Apache-Problemes relacionats, cosa que la converteix en una opció més fàcil d'usar per a principiants o aquells que prefereixen una gran base de coneixements.

<https://hostscore.net/es/learn/nginx-vs-apache/>

Com triar entre Nginx i Apache

Utilitzeu Nginx si:

Alt rendiment i escalabilitat: Nginx està optimitzat per manejar una gran quantitat de connexions simultànies, que el fa ideal per a llocs web o aplicacions d'alt trànsit que requereixen escalabilitat. La seva arquitectura basada en esdeveniments us permet manejar moltes connexions amb baix ús de memòria, la qual cosa ho fa eficient per al lliurament de contingut estàtic.

Equilibri de càrrega i proxy invers: Nginx s'utilitza sovint com a proxy invers i equilibrador de càrrega, distribuint el trànsit entrant entre múltiples servidors backend (excel·lent opció per entorns d'alta disponibilitat que requereixen una distribució de càrrega eficient)

Lliurament de contingut estàtic: Si la vostra aplicació ofereix una gran quantitat de contingut estàtic, com imatges, CSS o arxius JavaScript, Nginx sobresurt per la seva velocitat i eficiència a l'hora d'oferir arxius estàtics directament.

Simplicitat de configuració: Si bé Nginx no és compatible amb .htaccess, la seva sintaxi de configuració és senzilla i fàcil d'entendre. Aquesta simplicitat pot resultar beneficiosa per als que volen mantenir configuracions de servidor netes i senzilles.

<https://hostscore.net/es/learn/nginx-vs-apache/>

Com triar

Ús combinat



vs



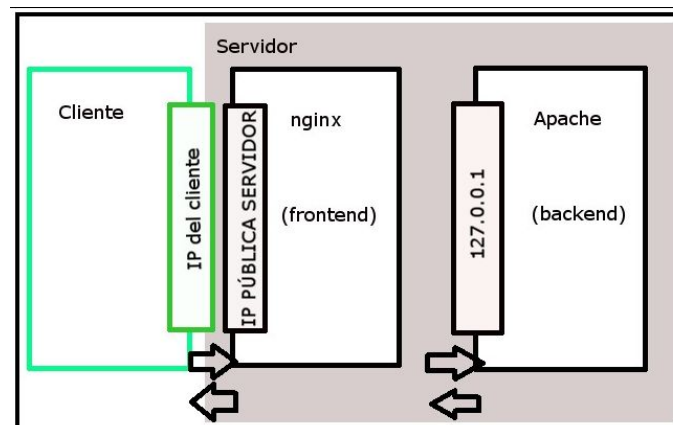
vs



Per a moltes configuracions, utilitzar tots dos Apache i Nginx junts poden aprofitar les fortaleces de cadascun.

Una configuració comuna és col·locar Nginx com a proxy invers davant d'Apache. Això permet a Nginx manejar contingut estàtic i connexions simultànies de manera eficient, mentre que Apache gestiona el processament dinàmic de continguts.

Aquest enfocament híbrid pot optimitzar el rendiment i la utilització de recursos en entorns web complexos.



<https://hostscore.net/es/learn/nginx-vs-apache/>

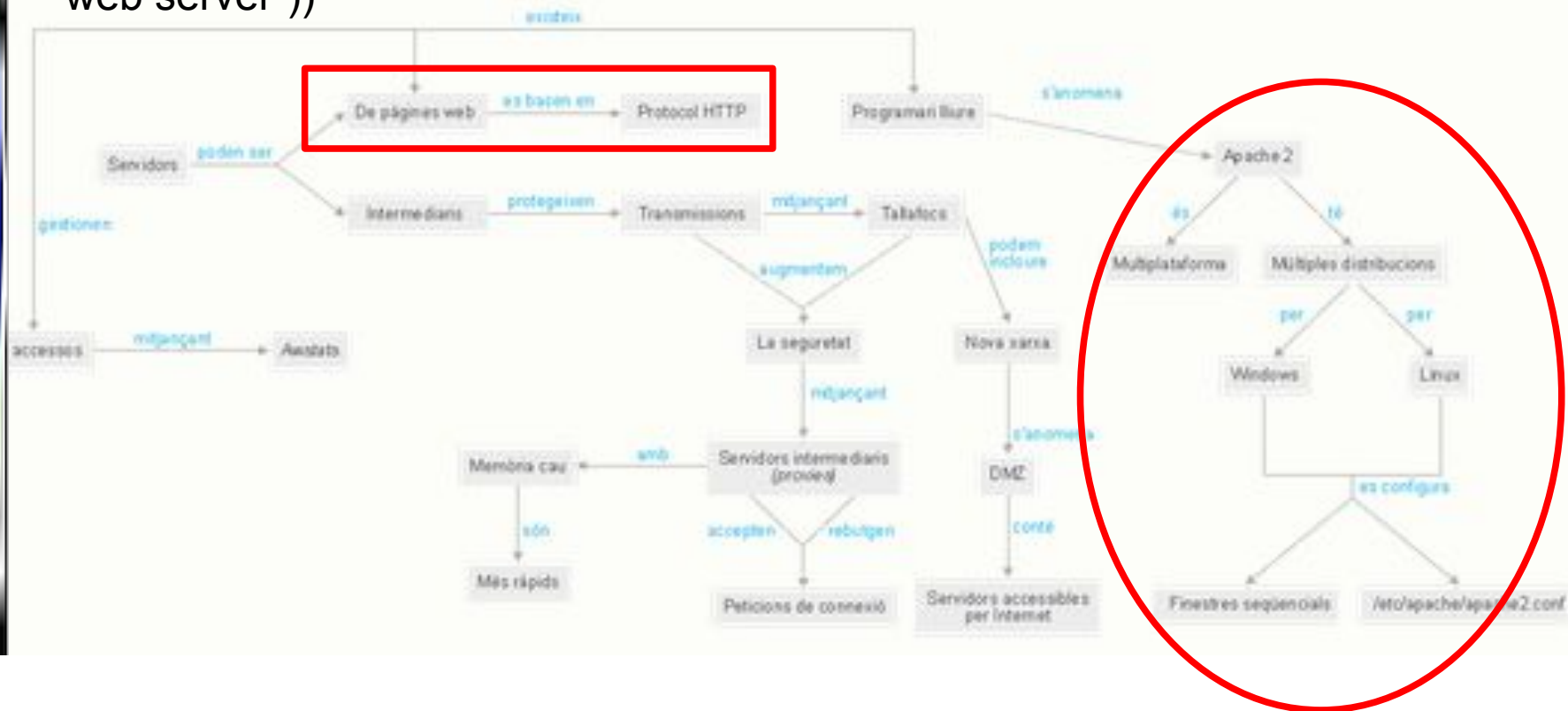
<https://rackset.com/blog/litespeed-vs-nginx-vs-apache-which-is-best-web-server/>

<https://colaboratorio.net/davidochobits/sysadmin/2017/accelera-tu-servidor-web-con-nginx-y-apache/>

És un servidor de **programari lliure GPL multiplataforma** ajustat a **HTTP/1.1**. Bookworm porta incorporada la versió 2.4.62 d'Apache.

0. Introducció Apache

Apache → a 1995 es va fer pública la 1a versió d'Apache (0.6.2). A partir de la versió 0.8 és un servidor modular, al nucli només hi ha les funcionalitats bàsiques. Actualment es fa servir la versió 2.4. (**tribu nativa nord-americana i paraula 'pegat' en anglès (patch)** “**servidor web apedaçat**” (“a patchy web server”))

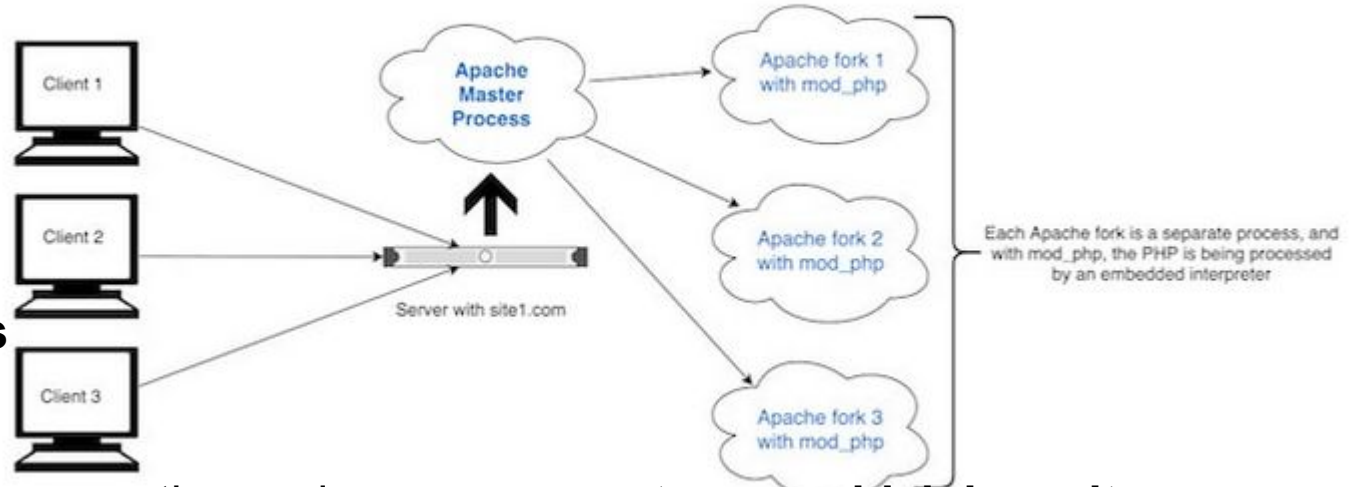


0. Introducció Apache

Server World		Other OS Configs	Privacy Policy
Debian 12 bookworm			
Install Debian	▶	Apache2	
NTP / SSH Server	▶	(01) Install Apache2	(21) Blog System : WordPress
DNS / DHCP Server	▶	(02) Configure Virtual Hostings	(22) Web Mail : RoundCube
Storage Server	▶	(03) Configure SSL/TLS	Nginx
Virtualization	▶	(04) Enable Userdir	(01) Install Nginx
Container Platform	▶	(05) Use CGI Scripts	(02) Configure Virtual Hostings
Cloud Compute	▶	(06) Use PHP Scripts	(03) Configure SSL/TLS
Directory Server	▶	(07) PHP + PHP-FPM	(04) Enable Userdir
Web Server	▶	(08) Basic Authentication	(05) Basic Authentication
Database	▶	(09) Configure WebDAV Folder	(06) Basic Auth + PAM
FTP / Samba / Mail	▶	(10) Basic Authentication + PAM	(07) Basic Auth + Kerberos
Proxy / Load Balance	▶	(11) Basic Authentication + LDAP	(08) Use CGI Scripts
Monitoring	▶	(12) Configure mod_md	(09) Use PHP Scripts
Security	▶	(13) Configure mod_proxy	(10) Nginx Reverse Proxy
Lang / Development	▶	(14) Configure mod_security	(11) Nginx Load Balancing
Desktop / Others	▶	(15) Configure mod_ratelimit	(12) Use Stream module
Others #2	▶	(16) Configure mod_http2	
Sponsored Link		(17) Configure mod_evasive	
		(18) Configure mod_perl	
		(19) Configure mod_wsgi	
		(20) Log Report : AWStats	

1. Apache

Les diferències
entre les variants
d'Apache 2 es
concentren en la



política utilitzada per gestionar el **processament en paral·lel de moltes comandes (aïlla les diferents peticions)**; aquesta política està implementada per un **MPM (abreviació de mòdul de multiprocessament: «Multi-Processing Module»)**. Són els responsables de connectar amb els ports de xarxa de la màquina, acceptar les peticions i generar els processos fills que s'encarreguen de servirles.

Només un mòdul MPM pot estar carregat de manera simultània. Entre els MPMS disponibles, **apache-mpm-worker utilitza fils (processos lleugers)** mentre que apache-mpm-prefork utilitza un conjunt de procés creats per avançat (la forma tradicional i l'única disponible a Apache 1.3).

Per defecte, la instal·lació del paquet apache2 fa que també s'instal·li la versió **apache2-mpm-worker d'Apache**

1. Apache

És un servidor de **programari lliure GPL multiplataforma** ajustat a **HTTP/1.1**. Bookworm porta incorporada la versió 2.4.62 d'Apache.

Implementa moltes possibilitats:

- protecció de documents i directoris mitjançant passwords (**htpasswd**)
- personalització de **missatges d'error** (suport multilinguatge)
- redireccionament (**.htaccess**) i **àlies** per a URLs
- suporta **IPv6 i SSL**
- **suport host virtuals** (proporcionen la capacitat de executar més d'un

lloc web “site” en una sola màquina)

El disseny modular permet escollir les característiques que seran incloses carregant els mòduls necessaris



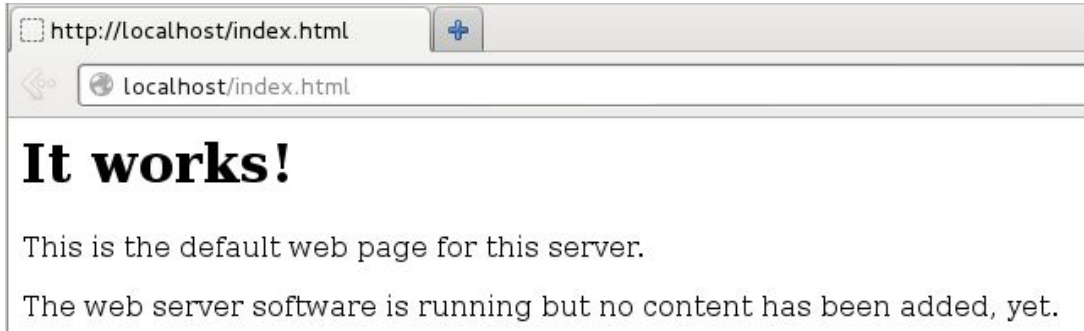
1. Apache

#apt update

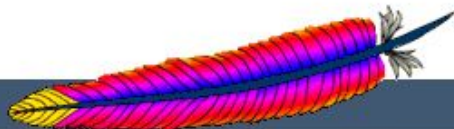
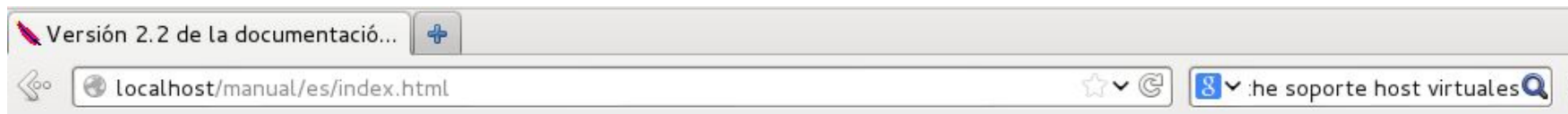
#apt -y install apache2

Per comprovar que tot ha anat bé reiniciem apache **#/systemctl restart apache2** i al nostre navegador piquem **localhost/index.html**

També és interessant instal·lar el paquet **apache2-doc**, que ens permetrà accedir a un manual web ... **localhost/manual** . Com sempre podem fer **#man apache2**



```
# Arranque y parada del servicio:  
service apache2 start | stop | status | restart | reload
```



[Módulos](#) | [Directivas](#) | [Preguntas Frecuentes](#) | [Glosario](#) | [Mapa de este sitio web](#)

Versión 2.2 del Servidor HTTP Apache

[Apache](#) > [Servidor HTTP](#) > [Documentación](#)

Versión 2.2 de la documentación del Servidor de HTTP Apache

Idiomas disponibles: [de](#) | [en](#) | [es](#) | [fr](#) | [ja](#) | [ko](#) | [pt-br](#) | [tr](#) | [zh-cn](#)

Esta traducción podría estar obsoleta. Consulte la versión en inglés de la documentación para comprobar si se han producido cambios recientemente.

squid és un proxy, no l'instal·lis, instal·la **apache2-utils** per fer autenticacions bàsiques

1. Apache

```
root@debianWheezy:/home/wheezy# apt-get install apache2 apache2-doc squid
```

```
Leyendo lista de paquetes... Hecho
```

```
Creando árbol de dependencias
```

```
Leyendo la información de estado... Hecho
```

```
Se instalarán los siguientes paquetes extras:
```

```
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common squid-common  
  squid-langpack
```

```
Paquetes sugeridos:
```

```
  apache2-suexec apache2-suexec-custom squidclient squid-cgi logcheck-database  
  resolvconf winbind
```

```
Se instalarán los siguientes paquetes NUEVOS:
```

```
  apache2 apache2-doc apache2-mpm-worker apache2-utils apache2.2-common squid  
  squid-common squid-langpack
```

```
Se actualizarán los siguientes paquetes:
```

```
  apache2.2-bin
```

```
1 actualizados, 8 se instalarán, 0 para eliminar y 141 no actualizados.
```

```
Necesito descargar 4.507 kB de archivos.
```

```
Se utilizarán 18,6 MB de espacio de disco adicional después de esta operación.
```

```
¿Desea continuar [S/n]? █
```

1. Apache

Per veure les **aplicacions** de les que disposa apache podem fer:

#dpkg -L apache2-common | grep bin

I per veure els **moduls** i els **arxius de configuració**

#dpkg -L apache2-common | grep etc

De fet l'arxiu principal a Debian és **/etc/apache2/apache2.conf** el qual defineix alguns **INCLUDES**, ja vam veure aquest terme amb DNS

Per comprovar que la sintaxi és correcte podem fer

#apache2ctl configtest

```
root@debianServer:/home/wheezy# apache2ctl configtest
Syntax OK
```

I per saber tot el que es pot configurar

#ls -l /etc/apache2

```
root@debianServer:/home/wheezy# ls -l /etc/apache2/
total 72
-rw-r--r-- 1 root root 9640 ene 31 19:35 apache2.conf
drwxr-xr-x 2 root root 4096 feb 15 09:54 conf.d
-rw-r--r-- 1 root root 1465 ene 31 19:35 envvars
-rw-r--r-- 1 root root 31063 jul 20 2013 magic
drwxr-xr-x 2 root root 4096 feb 15 09:54 mods-available
drwxr-xr-x 2 root root 4096 feb 15 09:54 mods-enabled
-rw-r--r-- 1 root root 750 ene 26 13:13 ports.conf
drwxr-xr-x 2 root root 4096 feb 15 09:54 sites-available
drwxr-xr-x 2 root root 4096 feb 15 09:54 sites-enabled
```


1. Apache

apache2.conf: Fitxer principal de configuració.

Carpeta conf.d: Els fitxers d'aquest directori són automàticament afegits mitjançant un include al fitxer apache2.conf (podem fer que **funcionin els accents** afegint AddDefaultCharset ISO-8859-15 a **/etc/apache2/conf.d/charset** o descomentar **AddDefaultCharset UTF-8**).

envvars: Variables d'entorn per apache2ctl.

ports.conf: Configura els ports pels quals Apache està donant servei.

magic: Dades de configuració del mòdul mod_mime_magic

mods-enabled i mods-available: Carpets de configuració dels mòduls. La carpeta mods-enabled conté els mòduls activats al servidor. La carpeta mods-available conté els mòduls disponibles. Un mòdul s'activa creant un link a un mòdul de la carpeta mods-available dins la carpeta mods-enabled.

sites-enabled i sites-available: Carpets de configuració dels llocs web. De forma anàloga a la gestió de mòduls, la gestió de Hosts Virtuals es realitza mitjançant l'ús d'aquestes dues carpetes.

1. Apache

apache2.conf: Fitxer principal de configuració.

En aquest fitxer trobem configuracions generals d'Apache. La configuració d'Apache funciona mitjançant directives. Típicament **una directiva, configura el valor d'un paràmetre**. Exemple **StartServer 2**

Típicament apache2.conf no s'haurà de tocar ja que les configuracions les farem a altres fitxers que estan inclosos mitjançant Include

Com sempre abans de començar **fem una còpia de seguretat**

#cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf_Backup

I podem visualitzar o afegir a un fitxer el contingut sense comentaris

#cat /etc/apache2/apache2.conf | grep -v '^#'

1. Apache

```
6 LockFile ${APACHE_LOCK_DIR}/accept.lock
7
8 PidFile ${APACHE_PID_FILE}
9
10 Timeout 300
11
12 KeepAlive On
13
14 MaxKeepAliveRequests 100
15
16 KeepAliveTimeout 5
17
18
19 <IfModule mpm_prefork_module>
20     StartServers          5
21     MinSpareServers       5
22     MaxSpareServers      10
23     MaxClients            150
24     MaxRequestsPerChild   0
25 </IfModule>
26
27 <IfModule mpm_worker_module>
28     StartServers          2
29     MinSpareThreads       25
30     MaxSpareThreads      75
31     ThreadLimit           64
32     ThreadsPerChild       25
33     MaxClients            150
34     MaxRequestsPerChild   0
35 </IfModule>
```

Per optimitzar MPM al servidor hem de saber que fa cada directiva

StartServer n°fills → n° processos fills al iniciar el servei. Això es modifica dinàmicament segons la càrrega (no és massa important)

MinSpareThreads n° MaxSpareThreads n° → n° mínim i màxim de fils

ThreadLimit → límit del n° de fils per procés fill.

ThreadsPerChild → n° fils creat per cada procés fill. Ha de ser menor que ThreadLimit però no massa per no tenir molta memòria reservada (allocated)

MaxClients n° → n° màxim de connexions que seran processades de manera simultània (clients alhora). Els que superen el límit han de esperar en cua)

MaxRequestPerChild n° → Límit de peticions que pot atendre un servidor fill. 0 significa il·limitat

....

1. Apache

LogFormat configura com seran les línies del fitxer de log

```
62  HostnameLookups Off
63
64  ErrorLog ${APACHE_LOG_DIR}/error.log
65
66  LogLevel warn
67
68  Include mods-enabled/*.load
69  Include mods-enabled/*.conf
70
71  Include ports.conf
72
73  LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
74  LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
75  LogFormat "%h %l %u %t \"%r\" %>s %O" common
76  LogFormat "%{Referer}i -> %U" referer
77  LogFormat "%{User-agent}i" agent
78
79
80  Include conf.d/
81
82  Include sites-enabled/
```

The characteristics of the request itself are logged by placing "%" directives in the format string, which are replaced in the log file by the values as follows:

Format String	Description
%%	The percent sign
%a	Remote IP-address
%A	Local IP-address
%B	Size of response in bytes, excluding HTTP headers.
%b	Size of response in bytes, excluding HTTP headers. In CLF format, i.e. a '-' rather than a 0 when no bytes are sent.
%{Foobar}C	The contents of cookie <i>Foobar</i> in the request sent to the server. Only version 0 cookies are fully supported.
%D	The time taken to serve the request, in microseconds.
%{FOOBAR}e	The contents of the environment variable <i>FOOBAR</i>
%f	Filename
%h	Remote host
%H	The request protocol
%{Foobar}i	The contents of <i>Foobar</i> : header line(s) in the request sent to the server. Changes made by other modules (e.g. mod_headers) affect this. If you're interested in what the request header was prior to when most modules would have modified it, use mod_setenvif to copy the header into an internal environment variable and log that value with the %{VARNAME}e described above.
%k	Number of keepalive requests handled on this connection. Interesting if KeepAlive is being used, so that, for example, a '1' means the first keepalive request after the initial one, '2' the second, etc...; otherwise this is always 0 (indicating the initial request). Available in versions 2.2.11 and later.
%l	Remote logname (from identd, if supplied). This will return a dash unless mod_ident is present and IdentityCheck is set On.
%m	The request method
%{Foobar}n	The contents of note <i>Foobar</i> from another module.
%{Foobar}o	The contents of <i>Foobar</i> : header line(s) in the reply.
%p	The canonical port of the server serving the request

% <i>{format}</i> p	The canonical port of the server serving the request or the server's actual port or the client's actual port. Valid formats are canonical, local, or remote.
%P	The process ID of the child that serviced the request.
% <i>{format}</i> P	The process ID or thread id of the child that serviced the request. Valid formats are pid, tid, and hextid. hextid requires APR 1.2.0 or higher.
%q	The query string (prepended with a ? if a query string exists, otherwise an empty string)
%r	First line of request
%R	The handler generating the response (if any).
%s	Status. For requests that got internally redirected, this is the status of the *original* request --- %>s for the last.
%t	Time the request was received (standard english format)
% <i>{format}</i> t	The time, in the form given by format, which should be in strftime(3) format. (potentially localized)
%T	The time taken to serve the request, in seconds.
%u	Remote user (from auth; may be bogus if return status (%s) is 401)
%U	The URL path requested, not including any query string.
%v	The canonical ServerName of the server serving the request.
%V	The server name according to the UseCanonicalName setting.
%X	<p>Connection status when response is completed:</p> <div> <p>X = connection aborted before the response completed.</p> <p>+ = connection may be kept alive after the response is sent.</p> <p>- = connection will be closed after the response is sent.</p> </div> <p>(This directive was %c in late versions of Apache 1.3, but this conflicted with the historical ssl %<i>{var}</i>c syntax.)</p>
%I	Bytes received, including request and headers, cannot be zero. You need to enable mod_logio to use this.
%O	Bytes sent, including headers, cannot be zero. You need to enable mod_logio to use this.

https://httpd.apache.org/docs/2.4/mod/mod_log_config.html

2. Hosts Virtuals “sites”

El servidor Apache, com està configurat fins ara, **utilitza els mateixos arxius de registre per a tots els servidors virtuals (pot canviar afegint directives CustomLog en les definicions de servidors virtuals)**. Per tant, té sentit personalitzar el format d'aquest fitxer de registre per incloure el nom del servidor virtual. Pot fer-ho creant un arxiu /etc/apache2/conf.d/customlog que defineix un nou format per a tots els fitxers de registre (amb la directiva LogFormat). També ha d'eliminar (o comentar) la línia CustomLog a l'arxiu /etc/apache2/sites-available/default. Exemple → El fitxer

/etc/apache2/conf.d/customlog

Nou format de registre que inclou el nom del servidor (virtual)

LogFormat "%v %h %l %o %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" \"\" vhost

Ara utilitzem aquest format per defecte

CustomLog /var/log/apache2/access.log vhost

Ejemplo 11.17. El archivo /etc/apache2/conf.d/customlog

```
# Nuevo formato de registro que incluye el nombre del servidor (virtual)
LogFormat "%v %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" \"\" vhost

# Ahora utilicemos este formato de forma predeterminada
CustomLog /var/log/apache2/access.log vhost
```

quan fem el fitxer dins de sites-aviable ha d'acabar en .conf per exemple www.cicles.cat.conf , sinó no funcionarà a2ensite

2. Hosts Virtuals “sites”

Apache considera dos tipus diferents de servidors virtuals: els **basats en l'adreça IP (o port)** i aquells **basats en el nom de domini del servidor web**. El primer mètode requereix reservar una adreça IP (o port) diferent per a cada lloc, mentre que **el segon pot funcionar en només una adreça IP** (i port) i es diferencien els llocs pel nom enviat pel client HTTP.

L'escassetat d'IPv4 generalment afavoreix el segon mètode, però, és **més complex si els servidors virtuals també necessiten proveir HTTPS** ja que el protocol SSL no sempre es va adequar als servidors virtuals basats en noms, no tots els navegadors són compatibles amb l'extensió SNI (indicació de nom de servidor: «Server Name indication») que permet aquesta combinació. **Quan diversos llocs HTTPS necessiten executar en el mateix servidor, generalment es diferenciaran bé per executar en un port o en una adreça IP diferent** (IPv6 pot ajudar).

Per **defecte d'Apache 2**, activa servidors virtuals basats en nom (amb la **directiva NameVirtualHost *: 80** a /etc/apache2/ports.conf). A més, defineix un servidor virtual per defecte al link /etc/apache2/sites-enabled/000-default.conf; utilitzarà aquest servidor virtual si no es troba cap servidor que coincideixi amb la comanda enviada pel client.

2. Hosts Virtuals “sites”

Exemple per ports diferents (a la nostra pràctica no el farem així).
Hem d'indicar els ports on volem que escolti Apache al fitxer

```
#Vamos a habilitar los puertos dónde va a escuchar nuestro servidor Apache:
nano /etc/apache2/ports.conf

Listen 80
Listen 9999

# Si queremos que Apache escuche en unas IP específicas tendremos que poner también la IP:puerto
Listen 192.168.2.250:80

# Mediante la directiva NameVirtualHost indicaremos que vamos a tener hosts virtuales.
# El asterisco indica que vamos a recibir peticiones de VirtualHost en todas las IP en las que
# está escuchando nuestro servidor web. Si nos interesa una IP específica la indicaremos.
# A continuación pondremos : y el número de puerto.
NameVirtualHost *:80

# Para cada servidor virtual, crearemos una línea como la anterior. Por ejemplo, añadimos una más:
NameVirtualHost *:9999
```

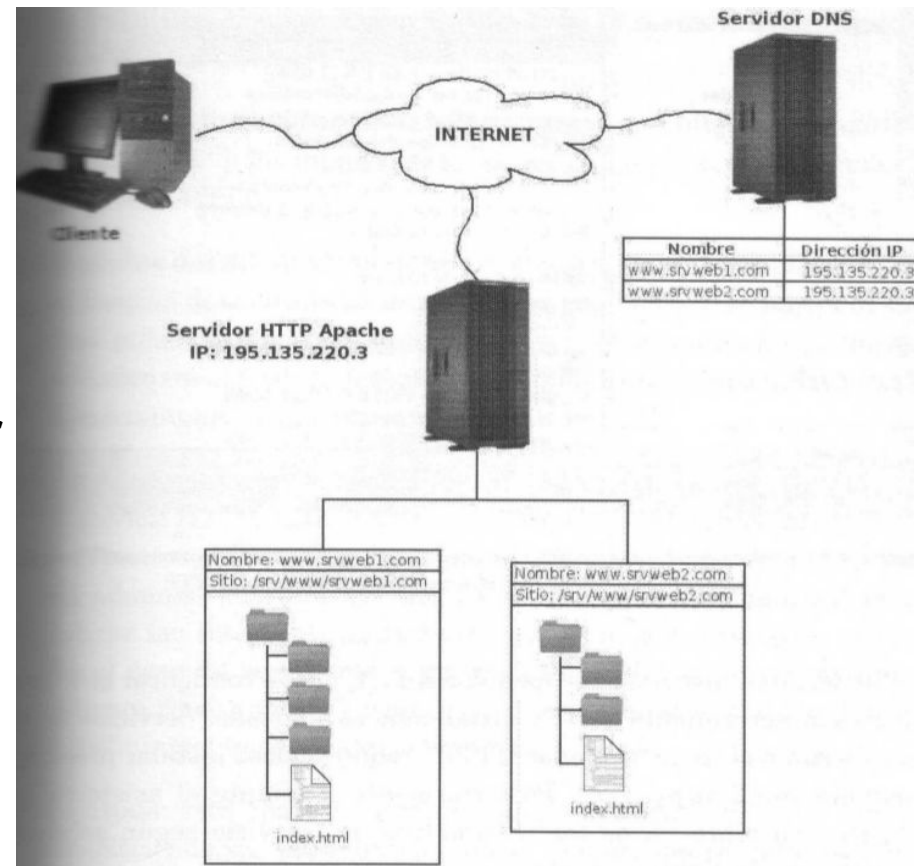
Si volem que es mostrin **webs diferents a l'accedir per un port o un altre**,
editarem les directives corresponents dins del fitxer
/etc/apache2/sites-available/default, que és el lloc habilitat per defecte i
configurarem els paràmetres de DocumentRoot, DirectoryIndex, ServerName
, etc. http://informatica.iessanclemente.net/manuais/index.php/Instalaci%C3%B3n_de_Servidor_Web_Apache_en_Debian

2. Hosts Virtuals “sites”

Apache permet configurar llocs web situats a la mateixa màquina donant la impressió al client d'estar accedint a equips servidors diferents (<http://www.cicles.ins> i a la mateixa màquina configurem <http://www.ciutat.com>)

Per tal que els **clients** puguin **accedir** mitjançant les **URL corresponents**, **necessitem un servidor DNS** (associem cadascun dels noms configurats amb l'adreça IP del servidor, o configurar el fitxer /etc/hosts per fer les proves internes).

Un servidor virtual hereta les propietats del servidor principal.



actualment el directori per defecte és /var/www/html a pràcticament tots els linux

2. Hosts Virtuals “sites”

La configuració d'un site (llocs web) es pot fer de 3 maneres:

/etc/apache2/conf.d: Posant un fitxer configuració en aquesta carpeta i reiniciant el servidor

a2ensite/a2dissite: Guardant la configuració a /etc/apache2/sites-available

/var/www/html: Aquesta és la carpeta per defecte (a sistemes Debian) on es troben les aplicacions web del servidor.

/etc/apache2/apache2.conf: Podeu posar la configuració directament a aquest fitxer. OPCIÓ NO RECOMANDA

Distribució	Directorí arrel per defecte	Fitxer de configuració
Debian	/var/www/	/etc/apache2/apache2.conf
Suse	/srv/www/htdocs/	/etc/apache2/httpd.conf
Red hat (Fedora, CentOS)	/var/www/html	/etc/httpd/conf/httpd.conf

```
<Directory /var/www>
Options Includes FollowSymlinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

2. Hosts Virtuals “sites”

Directives Comuns

L'arxiu de configuració principal generalment inclou diversos **blocs Directory** que permeten diferents comportaments del servidor depenent de la ubicació de l'arxiu que està proveint. **Aquests blocs usualment inclouen directives Options i AllowOverride.**

La directiva Options ha de seguir-se d'una llista d'opcions a activar. El valor None desactiva totes les opcions; All les activa totes excepte MultiViews. Les opcions disponibles inclouen.

- » **ExecCGI** indica que puede ejecutar scripts CGI.
- » **FollowSymlinks** le dice al servidor que puede seguir los enlaces simbólicos y que la respuesta debe contener el contenido del objetivo de dichos enlaces.
- » **SymlinksIfOwnerMatch** también le indica al servidor que siga los enlaces simbólicos, pero sólo cuando el enlace y su objetivo tengan el mismo dueño.
- » **Includes** activa *inclusiones del lado del servidor* (SSI: «Server Side Includes»). Estas directivas se encuentran en las páginas HTML y son ejecutadas en el momento de cada pedido.
- » **Indexes** le indica al servidor que provea una lista del contenido de los directorios si el pedido HTTP del cliente apunta a un directorio sin un archivo de índice (es decir, que no existe en él ninguno de los archivos enumerados en la directiva **DirectoryIndex**).
- » **MultiViews** activa la negociación de contenido; el servidor puede utilizar esto para proveer una página web que utilice el idioma preferido configurado en el navegador.

- DocumentRoot
- NameVirtualHost
- ServerAlias
- ServerName
- ServerPath
- <VirtualHost>

2. Hosts Virtuals “sites”

Les directives més fetes servir per a la configuració dels sites són:

DocumentRoot : indica el directori arrell pel lloc web (site) del servidor. El site per defecte és /var/www , es troba definit a /etc/apache2/sites-available/default

ServerName : Nom i port que fa servir el servidor per definir-se a ell mateix.

ServerAdmin : indica l'adreça de correu que s'enviarà als clients mitjançant els missatges d'error.

DirectoryIndex : indica els fitxers que podran actuar com a pàgina índex del lloc. DirectoryIndex index.php index.html index.htm

Exemple configuració mitjançant /etc/apache2/conf.d (mediawiki acacha)
Suposeu que heu instal·lat l'aplicació web mediawiki a la carpeta /var/lib/mediawiki. Per activar aquest lloc web faríem:

#geany /etc/apache2/conf.d/mediawiki.conf

NOTA: El fitxer ha de tenir la extensió .conf si voleu que funcioni.

2. Hosts Virtuals “sites”

Contingut del fitxer /etc/apache2/conf.d/mediawiki.conf

Alias /mediawiki /var/lib/mediawiki → amb aquesta línia ja funcionaria, però per seguretat es poden afegir més camps

```
<Directory /var/lib/mediawiki/>
```

```
Options +FollowSymLinks
```

```
AllowOverride All
```

```
order allow,deny
```

```
allow from all
```

```
</Directory>
```

some directories must be protected

```
<Directory /var/lib/mediawiki/config>
```

```
Options -FollowSymLinks
```

```
AllowOverride None
```

```
</Directory>
```

```
<Directory /var/lib/mediawiki/upload>
```

```
Options -FollowSymLinks
```

```
AllowOverride None
```

```
</Directory>
```

Ara només quedaria **#/etc/init.d/apache2 reload** i accedir amb el navegador a <http://localhost/mediawiki>

2. Hosts Virtuals “sites”

És convenient **organitzar el lloc web mitjançant directoris**. Se sol definir una **estructura anomenada contenidor per a cada un dels directoris als quals accedirà Apache**, incloent una per al directori arrel del lloc del servidor **principal**.

Aquestes estructures, es reconeixen per una etiqueta d'inici amb el format <Directory nom_directori> i una altra de final </Directory>.

Dins d'aquestes estructures es defineixen directives que en general, estableixen diferents permisos sobre el directori que té associat.

La configuració d'un senzill servidor web, requerirà modificar la directiva DocumentRoot, indicant el directori que exercirà com a arrel del lloc web. Aquesta directiva tindrà associada una estructura contenidora el nom de directori coincidirà amb l'indicat en DocumentRoot.

2. Hosts Virtuals “sites”

Directives que defineixen els permisos de les estructures

Directiva	Opciones	Función
Options	None	No establece ninguna opción.
	All	Establece todas las opciones.
	Indexes	Permite visualizar páginas índice existentes en el directorio.
	FollowSymlinks	Permite seguir los enlaces simbólicos del directorio.
	ExecCGI	Admite la ejecución de scripts CGI.
AllowOverride	None	No establece ninguna opción. El servidor no leerá los archivos .htaccess
	All	Establece todas las opciones. Permite usar las directivas especificadas en .htaccess
	FileInfo	Muestra la información de los archivos del directorio.
Order	allow,deny	Primero aplicará los permisos de allow y luego los de deny.
	deny,allow	Primero aplicará los permisos de deny y luego los de allow.
Allow	from all	Admite cualquier acceso al directorio.
	from IP	Admite cualquier acceso al directorio proveniente de la dirección IP indicada.
	from dominio	Admite cualquier acceso al directorio desde el dominio especificado.
Deny	from all	Deniega cualquier acceso al directorio.
	from IP	Deniega cualquier acceso al directorio proveniente de la dirección IP indicada.
	from dominio	Deniega cualquier acceso al directorio desde el dominio especificado.

2. Hosts Virtuals “sites”

**Exemple configuració
/etc/apache2/sites-available**

**Els fitxers que col·loqui a
aquesta carpeta són
configuracions de llocs web
que es poden activar i
desactivar amb les ordres
a2ensite i a2dissite.**

Suposem el mateix cas
anterior (mediawiki acacha).
Hauríem d'editar el següent
fitxer amb el mateix contingut
que abans

**#geany
/etc/apache2/sites-available/
mediawiki**

```
Alias /mediawiki /var/lib/mediawiki
<Directory /var/lib/mediawiki/>
    Options +FollowSymLinks
    AllowOverride All
    order allow,deny
    allow from all
</Directory>
# some directories must be protected
<Directory /var/lib/mediawiki/config>
    Options -FollowSymLinks
    AllowOverride None
</Directory>
<Directory /var/lib/mediawiki/upload>
    Options -FollowSymLinks
    AllowOverride None
</Directory>
```

quan fem el fitxer dins de sites-available ha d'acabar en .conf
per exemple www.cicles.cat.conf , sinó no funcionarà a2ensite

2. Hosts Virtuals “sites”

Però ara per activar-lo farem

#a2ensite mediawiki → Aquesta comanda l'únic que fa és crear un enllaç simbòlic del fitxer /etc/apache2/sites-available/mediawiki a /etc/apache2/sites-enabled/mediawiki:

#ls -la /etc/apache2/sites-enabled/mediawiki.conf

lrwxrwxrwx 1 root root 28 2010-03-21 11:33

/etc/apache2/sites-enabled/mediawiki -> ../sites-available/mediawiki

#/etc/init.d/apache2 reload → **actualment** → **systemctl reload apache2**

Ara ja podeu accedir a → <http://localhost/mediawiki>

Imaginem ara que fem el lloc **#geany**

/etc/apache2/sites-available/www.cicles.ins.conf

<VirtualHost *:80>

ServerName www.cicles.ins

DocumentRoot /var/www/html/cicles.ins

ServerAdmin admin@cicles.ins

</VirtualHost>

Ara hauríem de fer **#a2ensite www.cicles.ins** i tenir un fitxer index.htm dins de /var/www/cicles.ins amb alguna informació dins i

#systemctl reload apache2

quan fem el fitxer dins de sites-aviable ha d'acabar en .conf
per exemple www.cicles.cat.conf , sinó no funcionarà a2ensite

2. Hosts Virtuals “sites”

Primer de tot configurem DNS o /etc/hosts

```
root@debianServer:/etc/apache2/sites-available# geany /etc/hosts
```



```
< interfaces x hosts x resolv.conf x sources.list x hostname x nsswitch.conf x smbusers x
1 127.0.0.1 localhost
2 127.0.1.1 debianServer.estatut.ins debianServer
3 192.168.8.100 debianServer.estatut.ins debianServer www.estatut.ins www.rubi.com
```

Si el fem amb el fitxer hosts també l'hem de configurar al client



```
< interfaces x hosts x resolv.conf x sources.list x hostname x nsswitch.conf x smbusers x
1 127.0.0.1 localhost
2 127.0.1.1 debianWheezy.estatut.ins debianWheezy
3 192.168.8.100 debianServer.estatut.ins debianServer www.estatut.ins www.rubi.com
4 192.168.8.21 debianWheezy.estatut.ins debianWheezy
5 192.168.8.11 PDC.estatut.ins PDC
6 192.168.8.22 debianClient3.estatut.ins debianClient3
```

Després hem de crear les carpetes i els fitxers web (**actualment s'han de crear a /var/www/html**) S'hauria d'adaptar a cicles.ins i ciutat.com

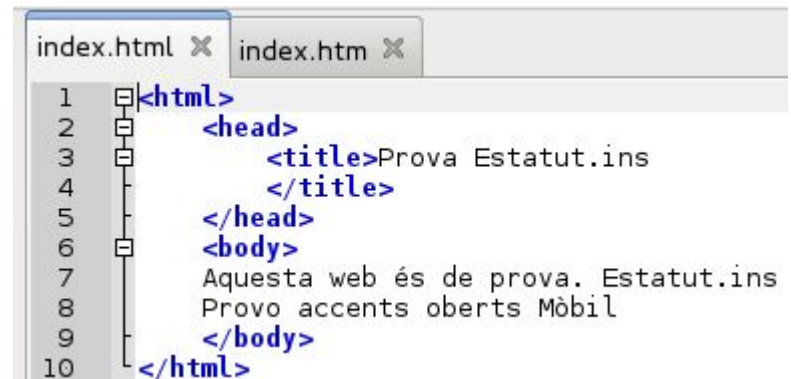
```
root@debianServer:/home/wheezy# mkdir /var/www/estatut.ins
root@debianServer:/home/wheezy# geany /var/www/estatut.ins/index.html
root@debianServer:/home/wheezy# mkdir /var/www/rubi.com
root@debianServer:/home/wheezy# geany /var/www/rubi.com/index.htm
root@debianServer:/home/wheezy# ls -l /var/www
```


2. Hosts Virtuals “sites”

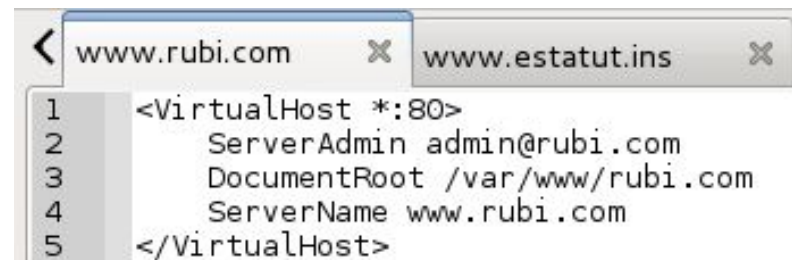
Contingut dels arxius `/etc/apache2/site-available/www.cicles.ins.conf` i [www.ciutat.com.conf](#) (configuració dels sites, aquest any els farem amb cicles i ciutat). Contingut del html i htm. També podíem haver inclòs la directiva `ServerAlias` `cicles.ins`.



```
< www.rubi.com x www.estatut.ins x
1 <VirtualHost *:80>
2     ServerAdmin admin@estatut.ins
3     DocumentRoot /var/www/estatut.ins
4     ServerName www.estatut.ins
5 </VirtualHost>
```



```
index.html x index.htm x
1 <html>
2   <head>
3     <title>Prova Estatut.ins
4   </title>
5   </head>
6   <body>
7     Aquesta web és de prova. Estatut.ins
8     Provo accents oberts Mòbil
9   </body>
10 </html>
```



```
< www.rubi.com x www.estatut.ins x
1 <VirtualHost *:80>
2     ServerAdmin admin@rubi.com
3     DocumentRoot /var/www/rubi.com
4     ServerName www.rubi.com
5 </VirtualHost>
```



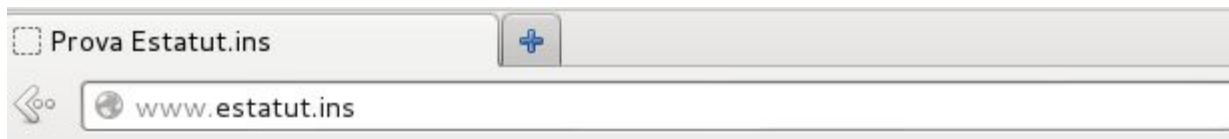
```
index.html x index.htm x
1 <html>
2   <head>
3     <title>Prova Rubi.com
4   </title>
5   </head>
6   <body>
7     Aquesta web és de prova. Rubi.com
8     Provo accents oberts Mòbil
9   </body>
10 </html>
```

2. Hosts Virtuals “sites”

Ara activem els sites. Fitxat que no funciona (a vegades si, però no sé per què) si no hi sou a dins del directori sites-available

```
root@debianServer:/home/wheezy# a2ensite /etc/apache2/sites-available/www.rubi.com
ERROR: No site found matching /etc/apache2/sites-available/www.rubi.com!
root@debianServer:/home/wheezy# cd /etc/apache2/sites-available/
root@debianServer:/etc/apache2/sites-available# a2ensite www.rubi.com
Enabling site www.rubi.com.
To activate the new configuration, you need to run:
    service apache2 reload
root@debianServer:/etc/apache2/sites-available#
```

Aquesta comanda fa un link a dins del directori sites-enable, reiniciem el servei i ja podem connectar-nos des del client

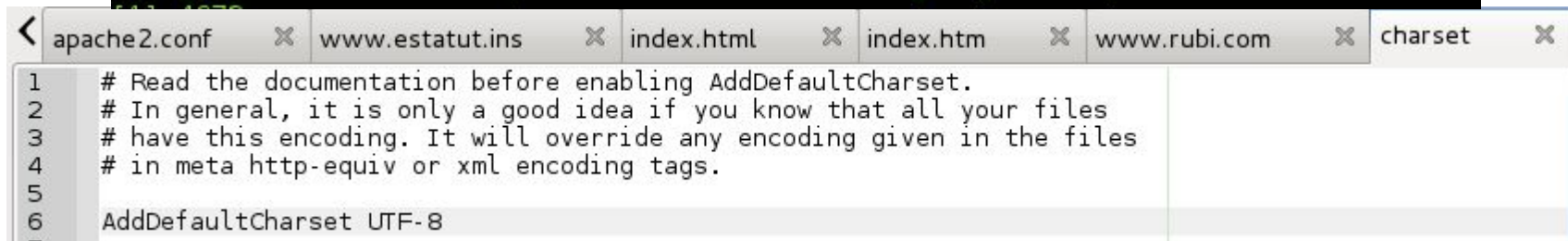


Aquesta web Ã©s de prova. Estatut.ins Provo accents oberts MÃ²bil

2. Hosts Virtuals “sites”

Com veiem els accents no funcionen, per què funcionin podem fer dues coses, escriure'ls en html o activar la variable dins /etc/apache2/conf.d/charset

```
root@debianServer:/etc/apache2/sites-available# geany /etc/apache2/conf.d/charset&
```

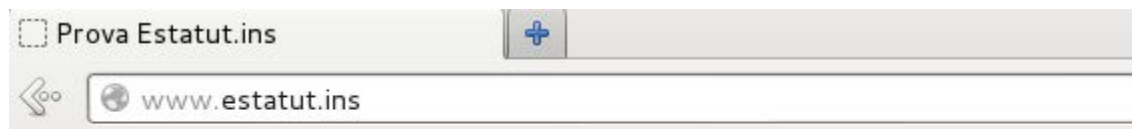


```
<
apache2.conf x www.estatut.ins x index.html x index.htm x www.rubi.com x charset x
1 # Read the documentation before enabling AddDefaultCharset.
2 # In general, it is only a good idea if you know that all your files
3 # have this encoding. It will override any encoding given in the files
4 # in meta http-equiv or xml encoding tags.
5
6 AddDefaultCharset UTF-8
```

```
<BODY>
  P&agrave;gina d'inici
</BODY>
```

```
root@debianServer:/etc/apache2/sites-available# service apache2 reload
[ ok ] Reloading web server config: apache2.
```

I ja podem connectar-nos des del client, si no funciona eliminem la caché del navegador



Aquesta web és de prova. Estatut.ins Provo accents oberts Mòbil

2. Hosts Virtuals “sites”

En cas de fer servir DNS quedaria més o menys així però substituint `srvweb1.com` i `srvweb2.com` per www.cicles.ins i www.ciutat.com (en aquest cas no hem de fer res als clients)

Editem `named.conf.local` creant les zones que toquin

```
//  
// Do any local configuration here  
//  
  
// Consider adding the 1918 zones here, if they are not used  
// by your organization  
//include "/etc/bind/zones.rfc1918";  
zone "iesrubi.org" {  
    type master;  
    allow-query {127.0.0.1;192.168.18.0/24;};  
    file "/etc/bind/db.iesrubi.org";  
};  
zone "srvweb1.com" {  
    type master;  
    file "/etc/bind/db.srvweb1.com";  
};  
zone "srvweb2.com" {
```

2. Hosts Virtuals “sites”

Després editem els fitxers de configuració associats a cada zona

```
$TTL 604800
@ IN SOA DEBIAN-JSR.srvweb1.com. root.DEBIAN-JSR.srvweb1.com. (
    2013011001 ; Numero de serie
    86400      ; Refresco
    7200       ; Reintento
    2592000    ; Expiracion
    172800 )   ; TTL
srvweb1.com. IN NS DEBIAN-JSR.srvweb1.com.
DEBIAN-JSR IN A 192.168.18.10
www IN CNAME DEBIAN-JSR.srvweb1.com.
```

Fem check de la zona (recordem també **#named-checkconf** **/etc/bind/named.conf** i **#/usr/sbin/named-checkconf -z**) i reiniciem el servei

```
root@DEBIAN-JSR:/etc/bind# named-checkzone srvweb2.com db.srvweb2.com
zone srvweb2.com/IN: loaded serial 2013011001
OK
```

```
root@DEBIAN-JSR:/etc/bind# /etc/init.d/bind9 restart
```

2. Hosts Virtuals “sites”

Exemple configuració /var/www/html

Tot el que col·loquem a aquesta carpeta per exemple /var/www/glpi, /var/www/html/mediawiki ... i podrem accedir fent <http://localhost/glpi> o <http://localhost/mediawiki>

Seguretat

Per defecte, Apache administra totes les peticions entrants sota la identitat de l'**usuari www-data**. Això vol dir que, **en cas d'una vulnerabilitat** de seguretat en un script CGI executat per Apache (per a una pàgina dinàmica), **no es comprometrà tot el sistema sinó només els arxius que són propietat d'aquest usuari** en particular.

Els mòduls **suexec** permeten **evitar aquesta limitació** perquè **alguns scripts CGI executin sota la identitat d'altres usuaris**. Pot configurar amb la **directiva SuexecUserGroup usuari grup** en la configuració d'Apache.

2. Hosts Virtuals “sites”

Seguretat

Una **altra possibilitat** és utilitzar un **MPM dedicat**, com el que proveeix el **paquet apache2-mpm-itk**. Aquest MPM en particular té un comportament lleugerament diferent: **permet «aïllar» els servidors virtuals («virtual hosts»)** perquè **cadascú executi com un usuari diferent**. Per tant, una vulnerabilitat en un lloc web no pot comprometre els arxius que pertanyen al propietari d'un altre lloc web.

Llista completa dels mòduls estàndard d'**Apache** a la seva web.

<https://httpd.apache.org/docs/2.4/mod/index.html>

Apache 2.4, així com ve, **inclou el mòdul SSL** necessari **per** a HTTP segur (**HTTPS**). Només **necessita activar amb a2enmod SSL** i després **afegir les directives necessàries als fitxers de configuració**. Podeu trobar un arxiu de configuració d'exemple a /
usr/share/doc/apache2.4-common/examples/apache2/extra/httpd-ssl.conf.gz.

2. Hosts Virtuals “sites”

Seguretat

Ha de tenir certes **cures addicionals** si prefereix connexions **SSL amb «Perfect Forward Secrecy»** (secret perfecte a futur, on **les connexions utilitzen claus efímeres en cada sessió** assegurant que si es compromet la clau privada del servidor no signifiqui que s'hagi compromès tot el trànsit antic que pot haver emmagatzemat des de la xarxa). En particular, **revisi les recomanacions de Mozilla:**

https://wiki.mozilla.org/Security/Server_Side_TLS#Apache

L'opció per defecte només activa els mòduls més comuns, però activar nous mòduls és tan simple com executar `a2enmod mòdul`; podrà desactivar un mòdul executant `a2dismod mòdul`. En realitat, aquests programes només creen (o eliminen) enllaços simbòlics a `/etc/apache2/mods-enabled/` que apunten als arxius en si (emmagatzemats a `/etc/apache2/mods-available/`). **Manera anàloga a `a2ensite` i `a2dissite`.**

Recordem **de manera predeterminada escoltem pel port 80** → configurat a `/etc/apache2/ports.conf` i **servim webs situades a `/var/www`** → configurat a `/etc/apache2/sites-enabled/default`

2. Hosts Virtuals “sites”

Seguretat → El paquet Apache proveït per **Debian** és compilat amb la **compatibilitat per SNI** «Server Name Indication», no **necessita** cap configuració particular a més d'**activar els servidors virtuals basats en nom al port 443 (SSL) a més de l'usual al port 80**. Per això editem el fitxer /etc/apache2/ports.conf perquè inclogui el següent :

aquestes línies no venen per defecte

```
NameVirtualHost *:80
Listen 80

<IfModule mod_ssl.c>
    # If you add NameVirtualHost *:443 here, you will also have to change
    # the VirtualHost statement in /etc/apache2/sites-available/default-ssl
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hosts is currently not
    # supported by MSIE on Windows XP.
    NameVirtualHost *:443
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    NameVirtualHost *:443
    Listen 443
</IfModule>
```

També ha de tenir cura d'**assegurar** que la configuració del **primer servidor virtual** (l'utilitzat per defecte) **tingui TLSv1 actiu**, ja que Apache utilitza els paràmetres d'aquest primer servidor virtual per establir connexions segures i ¡hauria de permetre!

2. Hosts Virtuals “sites”

Seguretat amb certificats

Es podria tenir un **servidor virtual principal** per al lloc principal i que atén al **port 80**, i un **servidor virtual** per al lloc segur que atén al **port 443** i que utilitza SSL.

Per oferir pàgines de forma segura en un lloc web, el servidor necessita un certificat que aprovi el seu autenticidad. Per **generar un certificat autosignat cal executar la instrucció make-ssl-cert**, i se'ns sol·licitarà informació per completar aquest certificat. Haurem d'indicar com a paràmetre el **fitxer amb extensió ". pem"**, que es generarà al final de l'execució i que **contindrà les claus per efectuar el xifrat asimètric pel servidor**.

Es necessari habilitar el port de escolta mitjançant Listen 443

```
# Editamos el fichero /etc/apache2/ports.conf:
nano /etc/apache2/ports.conf

# Añadimos los servidores virtuales:
Listen 80
Listen 443

NameVirtualHost *:80
NameVirtualHost *:443
```

2. "sites"

Seguretat amb certificats

Si llegim la documentació per a Debian a la part de SSL veiem els mòduls i sites que hem d'habilitar, i les comandes per generar els certificats o modificar-lo si canviem el nom de la màquina.

#a2enmod ssl

#a2ensite default-ssl

#!/etc/init.d/apache2 restart

#apt-get install ssl-cert

IMPORTANT: Tingueu en compte que el navegador us donarà un error SSL al tractar-se d'un certificat SSL autosignat

```
Enabling SSL
-----

To enable SSL, type (as user root):

    a2ensite default-ssl
    a2enmod ssl

If you want to use self-signed certificates, you should install the ssl-cert
package (see below). Otherwise, just adjust the SSLCertificateFile and
SSLCertificateKeyFile directives in /etc/apache2/sites-available/default-ssl to
point to your SSL certificate. Then restart apache:

    /etc/init.d/apache2 restart

The SSL key file should only be readable by root, the certificate file may be
globally readable. These files are read by the Apache parent process which runs
as root. Therefore it is not necessary to make the files readable by the
www-data user.

Creating self-signed certificates
-----

If you install the ssl-cert package, a self-signed certificate will be
automatically created using the hostname currently configured on your computer.
You can recreate that certificate (e.g. after you have changed /etc/hosts or
DNS to give the correct hostname) as user root with:

    make-ssl-cert generate-default-snakeoil --force-overwrite

To create more certificates with different host names, you can use

    make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /path/to/cert-file.crt

This will ask you for the hostname and place both SSL key and certificate in
the file /path/to/cert-file.crt . Use this file with the SSLCertificateFile
directive in the Apache config (you don't need the SSLCertificateKeyFile in
this case as it also contains the key). The file /path/to/cert-file.crt should
only be readable by root. A good directory to use for the additional
certificates/keys is /etc/ssl/private .
```

2. Hosts Virtuals “sites”

Seguretat amb certificats

Per crear un certificat autosignat que guardarem al directori /etc/apache2/certificats/certificat1.pem executarem la comanda:

```
# Creamos el directorio dónde almacenaremos los certificados:
mkdir -p /etc/apache2/certificados

# Generamos el fichero .pem que contendrá el certificado:
# Con esta instrucción o bien con:
make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/certificados/miweb.local.pem

openssl req @$ -new -x509 -days 365 -nodes -out /etc/apache2/certificados/miweb.local.pem -keyout
/etc/apache2/certificados/miweb.local.pem

# Respondemos a las preguntas que se nos hacen en la generación del certificado.

# Damos permisos de lectura al certificado
chmod 600 /etc/apache2/certificados/miweb.local.pem

# Habilitamos el soporte SSL
a2enmod ssl

# Reiniciamos el servicio
service apache2 restart
```

2. Hosts Virtuals “sites”

Seguretat amb certificats

A més **seran necessàries les següents directives en els contenidors dels llocs** al directori `/etc/apache2/sites-available`:

→ **SSLRequireSSL**: s'usa en el contenidor `<Directory>` i força la utilització de SSL, evitant la seva desactivació.

→ **SSLEngine**: habilita SSL (valor on) en un servidor virtual, si es va deshabilitar al servidor principal.

→ **SSLCertificateFile**: correspon al camí del fitxer (. pem) que conté el certificat

Com sempre un cop afegida tota la informació en d'assegurar-nos que el lloc està habilitat i reiniciar apache.

Mes info → <http://httpd.apache.org/docs/2.0/vhosts/examples.html>

Ara podem veure l'exemple d'aquesta configuració.


```
# Editamos el fichero /etc/apache2/sites-available/miweb.local
nano /etc/apache2/sites-available/miweb.local
```

```
# Comprobamos los contenedores y activamos las opciones correspondientes SSL:
# Comenzamos con el sitio SSL:
```

```
<VirtualHost *:443>
    ServerName      miweb.local
    DocumentRoot    /var/www/miweb.local/htdocs
    DirectoryIndex  index.html
    SSLEngine on
    SSLCertificateFile /etc/apache2/certificados/miweb.local.pem

    <Directory /var/www/miweb.local/htdocs>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride none
        Order allow,deny
        Allow from all
        SSLRequireSSL
    </Directory>
</VirtualHost>

<VirtualHost *:80>
    ServerName      miweb.local
    DocumentRoot    /var/www/miweb.local/htdocs
    DirectoryIndex  index.html
</VirtualHost>
```

Segons la web Acacha hem de posar això, hauríem de provar que ens funciona



Afegim les següents línies (per exemple després de la línia **ServerAdmin**):

```
-----
LoadModule ssl_module /usr/lib/apache2/modules/mod_ssl.so
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/certificat.pem
```

```
# Si tenemos un sitio default-ssl y lo habilitamos, revisaremos en ese fichero
# las rutas del certificado y activamos estas opciones:
```

```
SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
```

```
SSLCertificateFile    /etc/ssl/certs/server.crt
SSLCertificateKeyFile  /etc/ssl/private/server.key
```

```
# En el ejemplo anterior se permite el acceso a miweb.local en modo seguro y modo normal.
# Si quisiéramos crear dos webs diferentes para el acceso seguro o normal,
# simplemente modificaríamos los directorios DocumentRoot dónde se alojan los ficheros y ya está.
```

hem d'editar /etc/apache2/apache2.conf a l'apartat
<Directory /var/www>...AllowOverride All...</Directory>

3. Autenticació

En **algunes circumstàncies** necessitarà **restringir l'accés a parts d'un lloc web**, de manera que només usuaris legítims que proveeixin un nom d'**usuari i una contrasenya** tinguin accés al contingut. Es pot fer de **dues maneres**:

→ mitjançant el fitxer .htaccess

→ definint els usuaris i passwords a un fitxer i dient al Virtual Host (arxiu www.cicles.ins creat a /etc/apache2/sites-available) que les demani.

Arxiu .htaccess

Conté directives de configuració d'Apache que **s'apliquen cada vegada que una comanda involucra un element del directori en què es troba aquest arxiu**. L'abast d'aquestes directives també **inclou als seus subdirectoris**. **Aquests fitxers permeten configurar per directoris. Si un directori té un fitxer .htaccess les directives que trobem en aquest fitxers sobrescriuen la configuració general**. Normalment els fitxers .htaccess **s'utilitzen per tal que els usuaris que no tenen accés als fitxers de configuració principals puguin variar la seva configuració** (per exemple accedint via FTP). La directiva **AllowOverride** permet configurar quan els usuaris tenen privilegis per sobrescriure la configuració.

3. Autenticació

La majoria de les directives que poden ocórrer en un bloc **Directory** també són **vàlides en un arxiu .htaccess**. **Des d'un .htaccess podem:**

- Especificar els nostres propis documents d'error
- Afegir comportaments específics segons tipus MIME
- Definir variables d'entorn
- Redirigir URLs http://acacha.org/mediawiki/index.php/Redireccions_amb_Apache
- Re-escriure URLs
- Restringir accessos

La directiva **AllowOverride** enumera totes les opcions que poden ser **activades o desactivades en un arxiu .htaccess**. Un ús comú d'aquesta opció és **restringir ExecCGI** perquè els **administradors** puguin **triar els usuaris** que **podran executar programes sota** la identitat del servidor web (l'usuari **www-data**). **Ejemplo 11.19. Archivo .htaccess para autenticación obligatoria**

```
Require valid-user
AuthName "Private directory"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-private
```

3. Autenticació

(**Basic**) té una seguretat mínima ja que **s'envia la contrasenya en text pla** (codificada només amb base64 que és només una codificació, no un mètode de xifrat). També s'ha de saber que **els documents «protegits» per aquest mecanisme també són enviats sense xifrar** a través de la xarxa. **Si la seguretat és important**, s'ha de xifrar la connexió HTTP completa amb **SSL**.

El fitxer /etc/apache2/authfiles/htpasswd-private (el nom del fitxer pot ser un altre, inclús ocult. .htpasswd per exemple) conté una llista d'usuaris i contrasenyes que manipularem amb con el programa htpasswd. Per exemple, per agregar un usuari o canviar la seva contrasenya fem:

→ si no tenim el directori i el fitxer els fem #mkdir /etc/apache2/authfiles
#touch /etc/apache2/authfiles/htpasswd-private
#htpasswd /etc/apache2/authfiles/htpasswd-private usuario

```
# htpasswd /etc/apache2/authfiles/htpasswd-private usuario
New password:
Re-type new password:
Adding password for user usuario
```



```

root@debianServer:/home/wheezy# ls -l /etc/apache2/
total 76
-rw-r--r-- 1 root root 1473 mar  5 09:33 apache2.conf
-rw-r--r-- 1 root root 9640 mar  5 09:28 apache2.conf_Backup
drwxr-xr-x 2 root root 4096 mar  6 10:03 conf.d
-rw-r--r-- 1 root root 1465 ene 31 19:35 envvars
-rw-r--r-- 1 root root 31063 jul 20 2013 magic
drwxr-xr-x 2 root root 4096 feb 15 09:54 mods-available
drwxr-xr-x 2 root root 4096 feb 15 09:54 mods-enabled
-rw-r--r-- 1 root root  750 ene 26 13:13 ports.conf
drwxr-xr-x 2 root root 4096 mar  8 07:09 sites-available
drwxr-xr-x 2 root root 4096 mar  6 12:04 sites-enabled
root@debianServer:/home/wheezy# mkdir -p /etc/apache2/authfiles/

```

Ara si **mirem el fitxer** que he creat pels usuaris i passwords veur  el **password encriptat amb MD5**, que  s l'encriptaci  que porta per defecte (mirar man htpasswd on tamb  veurem que es pot crear un usuari amb l'opci  -c, per  reescriu l'arxiu, i eliminar-lo amb -D)

```

root@debianServer:/var/www# touch /etc/apache2/authfiles/.htpasswd

```

```

root@debianServer:/var/www# htpasswd /etc/apache2/authfiles/.htpasswd usuario
New password:
Re-type new password:
Adding password for user usuario
root@debianServer:/var/www# geany /etc/apache2/authfiles/.htpasswd&

```

```

< .htpasswd x apache2.conf x www.estatut.ins
1 usuario:$apr1$55I4UG5s$n2JafdrSxhi7XLg.A9rlg0
2

```

```

htpasswd [ -c ] [ -m ] [ -D ] passwdfile username

```

```

-c      Create the passwdfile. If passwdfile already exists, it is
        rewritten and truncated. This option cannot be combined with the
        -n option.

```

```

-D      Delete user. If the username exists in the specified htpasswd
        file, it will be deleted.

```


3. Autenticació

Ara faltaria col·locar l'arxiu **.htaccess** amb el **contingut adient** a **dins** del lloc on tenim la web o directori que volem protegir. Al nostre exemple a **/var/www/cicles.ins**

```
root@debianServer:/var/www# geany /var/www/estatut.ins/.htaccess
```

```
< .htaccess x apache2.conf x www.estatut.ins x index
1 AuthType Basic
2 AuthName "Restricted Files"
3 AuthUserFile /etc/apache2/authfiles/.htpasswd
4 Require user usuario
```

Identificación requerida



http://www.estatut.ins está solicitando un nombre de usuario y una contraseña. El sitio dice: "Restricted Files"

Nombre de usuario:

Contraseña:

Cancelar

Aceptar

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

Prova Estatut.ins x

Apache - SergiTurWiki x

ebre-escool/apache at maste... x



www.estatut.ins



Google

Aquesta web és de prova. Estatut.ins Provo accents oberts Mòbil. Ara des de .htaccess

3. Autenticació

Si volem definint els **usuaris i passwords a un fitxer i dient al Virtual Host** (arxiu `www.ciutat.com` creat a `/etc/apache2/sites-available`) que les demani.



The image shows two terminal windows side-by-side. The left window displays the configuration for a Virtual Host in `httpd.conf`. The right window shows the output of a web browser accessing `www.rubi.com`.

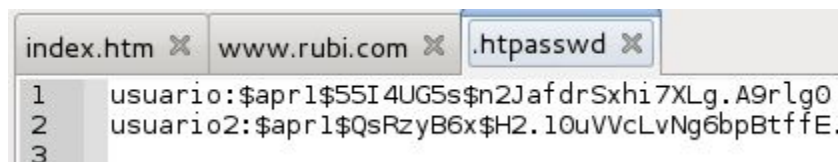
```
index.htm x www.rubi.com x .htpasswd x
1 <VirtualHost *:80>
2   ServerAdmin admin@rubi.com
3   DocumentRoot /var/www/rubi.com
4   ServerName www.rubi.com
5   <Directory /var/www/rubi.com>
6     Options Indexes FollowSymLinks
7     Order allow,deny
8     allow from all
9     AuthType Basic
10    AuthName "Zona amb autenticació requerida"
11    AuthUserFile /etc/apache2/authfiles/.htpasswd
12    Require user usuario usuario2
13  </Directory>
14
15 </VirtualHost>
```

```
index.htm x www.rubi.com x .htpasswd x
1 <html>
2   <head>
3     <title>Prova Rubi.com
4   </title>
5   </head>
6   <body>
7     Aquesta web és de prova. Rubi.com
8     Provo accents oberts Mòbil<br>
9     Ara amb autenticació configurada a Virtual Host.
10  </body>
11 </html>
```

Podríem permetre un rang **Allow from 192.168.8.0/24**

Suposem que fem servir el mateix arxiu que abans (si volem usuaris diferents per a cada lloc, farem un arxiu diferent para cada lloc). Afegim un altre usuari i així anem provant(tenim 2 usuaris usuario i usuario2):

`#htpasswd /etc/apache2/authfiles/.htpasswd usuario2`



The image shows a terminal window with the output of the `htpasswd` command. It lists two users: `usuario` and `usuario2`, each followed by a long, hashed password string.


```
index.htm x www.rubi.com x .htpasswd x
1 usuario:$apr1$55I4UG5s$n2JafdrSxhi7XLg.A9rlg0
2 usuario2:$apr1$QsRzyB6x$H2.10uVVcLvNg6bpBtffE.
3
```

3. Autenticació

Només ens queda reiniciar apache i accedir al web, on ens demanarà l'usuari i el password.

```
#systemctl reload apache2
```

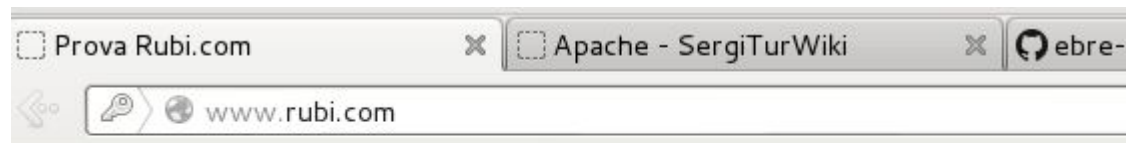
Identificación requerida



http://www.rubi.com está solicitando un nombre de usuario y una contraseña. El sitio dice: "Zona amb autenticació requerida"

Nombre de usuario:

Contraseña:



Aquesta web és de prova. Rubi.com Provo accents oberts Mòbil
Ara amb autenticació configurada a Virtual Host.