ContenIDOS

Instrucciones para entregar la práctica

Tenéis que demostrar que habéis realizado las prácticas de la 4 a la 15.

Corrección

- Se dará 1 punto por apartado demostrado (del 5 al 15).

- No es necesario indicar cada captura, únicamente las que sean necesarias para demostrar que se ha hecho el ejercicio.
- El apartado 16 completo da 1 punto extra en el examen. Solo se dará el punto si se han hecho los 10 anteriores.

Personalización de la práctica

En las capturas tenéis que demostrar que la configuración es diferente a la de los compañeros. Por ello:

- Siempre que podáis adaptarlas a vuestros nombres
- Cuando os pidan IPs a vuestro rango de IP.
- Si tenéis que crear carpetas o archivos, inventaros nombres que sean diferentes
- Es necesaria una pequeña explicación de lo que habéis hecho.

Formato

- Se debe presentar en formato Google Docs presentación, máximo 2 diapositivas por apartado.

Entrega

- Tendréis 5 horas más: lunes 1, martes 2 y jueves 2 más.
- Plazo máximo para la entrega: Domingo 13

Escenario de prácticas

Para estas prácticas de web, vamos a montar una red de toda la clase para poder compartir nuestro trabajo y hacer pruebas entre nosotros.

Vamos a partir de la red 172.16.254.0/24, y cada uno de vosotros tendrá 5 IPs asignadas:

- 2 IP para servidores
- 3 IP para clientes

Las IP las tendréis que sacar de un Excel que yo os facilitaré. Todos los equipos se configurarán con IPs estáticas.

Configuración de servidor Ubuntu Server

Para ello, seguiremos una serie de pasos:

1. Abrimos el Excel de asignación de IPs y nos anotamos en algún sitio nuestro rango de IPs
2. Abrimos nuestra MV ubserver01 y configuramos el archivo interfaces

a. Solo debe estar la interfaz enp0s3 que va a tener asignación estática

b. La configuración IP se coloca a partir de la información del Excel

c. Apagamos la MV

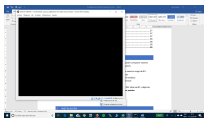
```
auto enp0s3
#iface enp0s3 inet dhcp

iface enp0s3 inet static
address 172.16.254.1
netmask 255.255.0.0
gateway 172.16.0.1
dns-nameservers 127.0.0.1 8.8.8.8
```

Ojo cambiar: dns-nameservers 172.16.254.1 8.8.8.8

3. Desde el menú de VBox, accedemos a las propiedades de red de la MV ubserver01 y dejamos solo activo el primer adaptador de red configurado como **adaptador puente**.

4. Arrancamos la MV

5. Comprobamos que la interfaz está habilitada desde VBox



6. Lanzamos ping contra el servidor DNS (172.16.254.1)

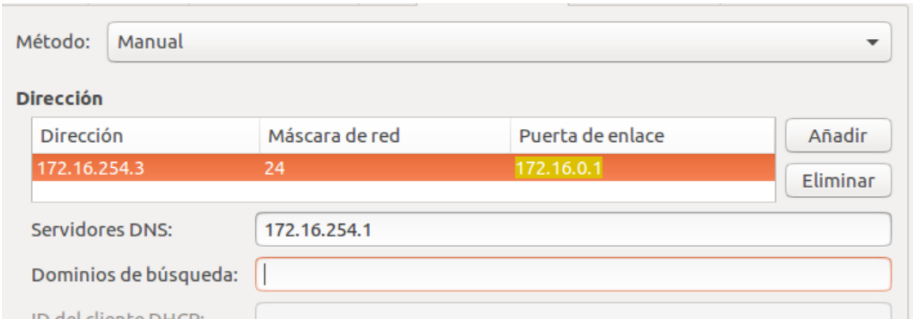
```
root@ubserver01:/etc/bind/cache [130] -> ping 172.16.254.1
PING 172.16.254.1 (172.16.254.1) 56(84) bytes of data.
64 bytes from 172.16.254.1: icmp_seq=1 ttl=64 time=0.053 ms
64 bytes from 172.16.254.1: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 172.16.254.1: icmp_seq=3 ttl=64 time=0.044 ms
64 bytes from 172.16.254.1: icmp_seq=4 ttl=64 time=0.043 ms
```

7. Intentamos resolver la IP de un nombre de dominio

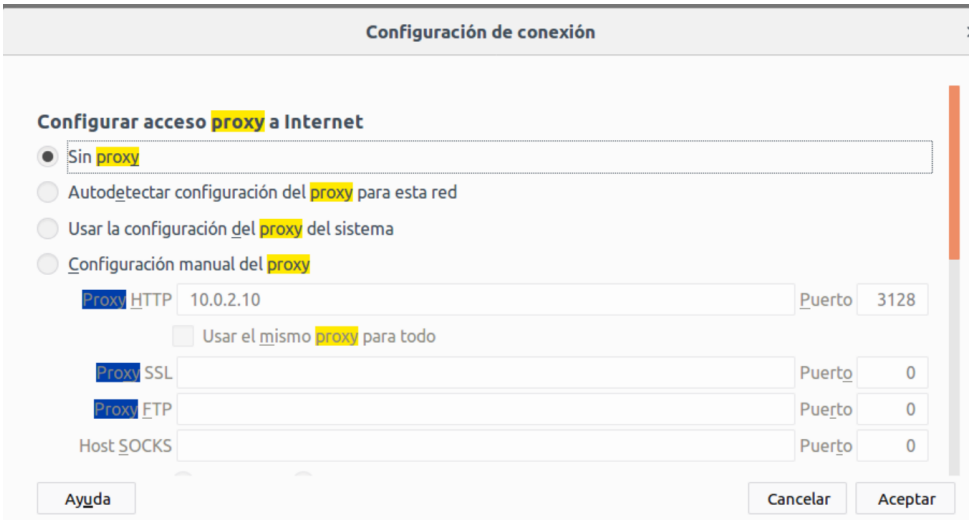
```
root@ubserver01:/etc/bind/cache -> nslookup jalzamora.smx2.org
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   jalzamora.smx2.org
Address: 172.16.254.6
```

Configuración del cliente Ubuntu Desktop

Repetimos los pasos con un cliente Ubuntu Desktop, y configuramos sus parámetros IP:



No está de más comprobar que no tenemos un proxy configurado en Firefox que nos esté impidiendo que todo funcione.



Reiniciamos el servidor: sudo systemctl restart apache2

Dominios y subdominios

Cada uno de vosotros tendrá un subdominio creado que apuntará a vuestra MV ubserver01. Deberéis acceder remotamente al servidor DNS y editarlo para agregar vuestro subdominio. Yo os iré avisando por turnos.

- El dominio de la clase será smx2.org.
- Cada máquina tendrá un nombre de subdominio con vuestra inicial de nombre y el primer apellido. Por ejemplo: jalzamora.smx2.org.
- Si todo va bien, con vuestro nombre de dominio particular se podrá acceder a vuestro servidor web.

Instalación

Disponer de un servidor web en el centro nos permitirá alojar nuestras propias páginas y aplicaciones web de forma que den servicio tanto desde dentro de la intranet como desde Internet.

Instalar apache

Comprobar si apache está instalado

```
pepe@ubserver02:~$ dpkg -s apache2
dpkg-query: el paquete `apache2' no está instalado y no hay ninguna información disponible.
Utilice dpkg --info (= dpkg-deb --info) para examinar archivos,
y dpkg --contents (= dpkg-deb --contents) para listar su contenido.
pepe@ubserver02:~$ _
```

Actualizamos la información de los paquetes para comprobar las últimas versiones

```
pepe@ubserver02:~$ sudo apt update
Des:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [83,2 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Des:5 http://security.ubuntu.com/ubuntu bionic-security/main i386 Packages [164 kB]
Des:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [208 kB]
Des:7 http://security.ubuntu.com/ubuntu bionic-security/universe i386 Packages [97,4 kB]
Des:8 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [97,6 kB]
Des:9 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [55,1 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [442 kB]
Des:11 http://es.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [393 kB]
Des:12 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [579 kB]
Des:13 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe i386 Packages [574 kB]
Des:14 http://es.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [157 kB]
Des:15 http://es.archive.ubuntu.com/ubuntu bionic-updates/multiverse i386 Packages [6.548 B]
Des:16 http://es.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [6.392 B]
Descargados 3.027 kB en 2s (1.678 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 43 paquetes. Ejecute «apt list --upgradable» para verlos.
```

Instalamos apache

```
pepe@ubserver02:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
Paquetes sugeridos:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-blacklist
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.2-0 ssl-cert
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 43 no actualizados.
Se necesita descargar 1.730 kB de archivos.
Se utilizarán 6.985 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Podemos comprobar versión de apache que tenemos instalada

```
pepe@ubserver02:~$ apache2 -v
Server version: Apache/2.4.29 (Ubuntu)
Server built:   2018-10-10T18:59:25
pepe@ubserver02:~$
```

Instalamos opcionalmente paquete de documentación

```
pepe@ubserver02:~$ sudo apt-get install apache2-doc
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  apache2-doc
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 43 no actualizados.
Se necesita descargar 3.697 kB de archivos.
Se utilizarán 24,0 MB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu bionic-updates/main amd64 apache2-doc all 2.4.29-1ubuntu4.
5 [3.697 kB]
Descargados 3.697 kB en 1s (4.076 kB/s)
Seleccionando el paquete apache2-doc previamente no seleccionado.
(Leyendo la base de datos ... 103254 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../apache2-doc_2.4.29-1ubuntu4.5_all.deb ...
Desempaquetando apache2-doc (2.4.29-1ubuntu4.5) ...
Configurando apache2-doc (2.4.29-1ubuntu4.5) ...
apache2_invoke: Enable configuration apache2-doc
pepe@ubserver02:~$ _
```

usuario www-data y el grupo www-data

Por seguridad se recomienda no utilizar el usuario root como usuario propietario del proceso apache2.

Hay que disponer de un usuario y grupo para ello. En Ubuntu se dispone del usuario y grupo www-data.

- cat /etc/passwrd | grep www-data para comprobar la creación del usuario www-data
- cat /etc/group | grep www-data para comprobar la creación del grupo www-data

```
pepe@ubserver02:~$ cat /etc/passwd | grep www-data
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
pepe@ubserver02:~$ cat /etc/group | grep www-data
www-data:x:33:
```

Directorio contenedor de páginas

Por defecto las páginas web se encuentran en /var/www/html/.

Cualquier carpeta y archivo dentro de esta carpeta será visible desde fuera. Por ejemplo:

- La carpeta /var/www/html/hola/quetal.html

- Se vería accediendo a: http://localhost/ hola/quetal.html

```
pepe@ubserver02:/var/www$ tree
.
└── html
    └── index.html

1 directory, 1 file
pepe@ubserver02:/var/www$
```

Configuración

El servidor HTTP Apache se configura mediante archivos de texto simples. Estos archivos pueden ubicarse en cualquiera de una variedad de lugares, dependiendo de la forma exacta en que instaló el servidor.

El archivo de configuración predeterminado generalmente se llama httpd.conf. Esto también puede variar en las distribuciones de terceros del servidor.

La configuración se divide con frecuencia en varios archivos más pequeños, para facilitar la administración. Estos archivos se cargan a través de la directiva Include. Los nombres o ubicaciones de estos subarchivos no son mágicos y pueden variar mucho de una instalación a otra.

Directorios de configuración de módulos

Mods disponibles

/etc/apache2/mods-available/

```
pepe@ubserver02:/var/www$ ll /etc/apache2/mods-available/
total 564
drwxr-xr-x 2 root root 12288 nov 28 10:08 ./
drwxr-xr-x 8 root root  4096 nov 28 10:08 ../
-rw-r--r-- 1 root root   100 oct 10 20:59 access_compat.load
-rw-r--r-- 1 root root   377 oct 10 20:59 actions.conf
-rw-r--r-- 1 root root    66 oct 10 20:59 actions.load
-rw-r--r-- 1 root root   843 oct 10 20:59 alias.conf
-rw-r--r-- 1 root root    62 oct 10 20:59 alias.load
-rw-r--r-- 1 root root    76 oct 10 20:59 allowmethods.load
-rw-r--r-- 1 root root    76 oct 10 20:59 asis.load
-rw-r--r-- 1 root root    94 oct 10 20:59 auth_basic.load
-rw-r--r-- 1 root root    96 oct 10 20:59 auth_digest.load
-rw-r--r-- 1 root root   100 oct 10 20:59 auth_form.load
-rw-r--r-- 1 root root    72 oct 10 20:59 authn_anon.load
-rw-r--r-- 1 root root    72 oct 10 20:59 authn_core.load
-rw-r--r-- 1 root root    85 oct 10 20:59 authn_dbd.load
-rw-r--r-- 1 root root    70 oct 10 20:59 authn_dbm.load
-rw-r--r-- 1 root root    72 oct 10 20:59 authn_file.load
```

Mods habilitados

/etc/apache2/mods-enable/

```
pepe@ubserver02:/var/www$ ll /etc/apache2/mods-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov 28 10:08 ./
drwxr-xr-x 8 root root 4096 nov 28 10:08 ../
lrwxrwxrwx 1 root root   36 nov 28 10:08 access_compat.load -> ../mods-available/access_compat.load
lrwxrwxrwx 1 root root   28 nov 28 10:08 alias.conf -> ../mods-available/alias.conf
lrwxrwxrwx 1 root root   28 nov 28 10:08 alias.load -> ../mods-available/alias.load
lrwxrwxrwx 1 root root   33 nov 28 10:08 auth_basic.load -> ../mods-available/auth_basic.load
lrwxrwxrwx 1 root root   33 nov 28 10:08 authn_core.load -> ../mods-available/authn_core.load
lrwxrwxrwx 1 root root   33 nov 28 10:08 authn_file.load -> ../mods-available/authn_file.load
lrwxrwxrwx 1 root root   33 nov 28 10:08 authz_core.load -> ../mods-available/authz_core.load
lrwxrwxrwx 1 root root   33 nov 28 10:08 authz_host.load -> ../mods-available/authz_host.load
lrwxrwxrwx 1 root root   33 nov 28 10:08 authz_user.load -> ../mods-available/authz_user.load
lrwxrwxrwx 1 root root   32 nov 28 10:08 autoindex.conf -> ../mods-available/autoindex.conf
lrwxrwxrwx 1 root root   32 nov 28 10:08 autoindex.load -> ../mods-available/autoindex.load
lrwxrwxrwx 1 root root   30 nov 28 10:08 deflate.conf -> ../mods-available/deflate.conf
lrwxrwxrwx 1 root root   30 nov 28 10:08 deflate.load -> ../mods-available/deflate.load
lrwxrwxrwx 1 root root   26 nov 28 10:08 dir.conf -> ../mods-available/dir.conf
lrwxrwxrwx 1 root root   26 nov 28 10:08 dir.load -> ../mods-available/dir.load
lrwxrwxrwx 1 root root   26 nov 28 10:08 env.load -> ../mods-available/env.load
lrwxrwxrwx 1 root root   29 nov 28 10:08 filter.load -> ../mods-available/filter.load
lrwxrwxrwx 1 root root   27 nov 28 10:08 mime.conf -> ../mods-available/mime.conf
lrwxrwxrwx 1 root root   27 nov 28 10:08 mime.load -> ../mods-available/mime.load
lrwxrwxrwx 1 root root   32 nov 28 10:08 mpm_event.conf -> ../mods-available/mpm_event.conf
lrwxrwxrwx 1 root root   32 nov 28 10:08 mpm_event.load -> ../mods-available/mpm_event.load
lrwxrwxrwx 1 root root   34 nov 28 10:08 negotiation.conf -> ../mods-available/negotiation.conf
lrwxrwxrwx 1 root root   34 nov 28 10:08 negotiation.load -> ../mods-available/negotiation.load
lrwxrwxrwx 1 root root   33 nov 28 10:08 reqtimeout.conf -> ../mods-available/reqtimeout.conf
lrwxrwxrwx 1 root root   33 nov 28 10:08 reqtimeout.load -> ../mods-available/reqtimeout.load
lrwxrwxrwx 1 root root   31 nov 28 10:08 setenvif.conf -> ../mods-available/setenvif.conf
lrwxrwxrwx 1 root root   31 nov 28 10:08 setenvif.load -> ../mods-available/setenvif.load
lrwxrwxrwx 1 root root   29 nov 28 10:08 status.conf -> ../mods-available/status.conf
lrwxrwxrwx 1 root root   29 nov 28 10:08 status.load -> ../mods-available/status.load
pepe@ubserver02:/var/www$
```

Directorios de configuración de sitios virtuales

Sites available

```
pepe@ubserver02:~$ ll /etc/apache2/sites-available/
total 20
drwxr-xr-x 2 root root 4096 nov 28 10:08 ./
drwxr-xr-x 8 root root 4096 nov 28 10:08 ../
-rw-r--r-- 1 root root 1332 oct 10 20:59 000-default.conf
-rw-r--r-- 1 root root 6338 oct 10 20:59 default-ssl.conf
pepe@ubserver02:~$
```

Sites enabled

```
pepe@ubserver02:~$ ll /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov 28 10:08 ./
drwxr-xr-x 8 root root 4096 nov 28 10:08 ../
lrwxrwxrwx 1 root root   35 nov 28 10:08 000-default.conf -> ../sites-available/000-default.conf
pepe@ubserver02:~$ _
```

Directorio de configuraciones locales

/etc/apache2/conf-available/

```
pepe@ubserver02:~$ ll /etc/apache2/conf-available/
total 32
drwxr-xr-x 2 root root 4096 nov 28 10:10 ./
drwxr-xr-x 8 root root 4096 nov 28 10:08 ../
-rw-r--r-- 1 root root  221 oct 10 20:59 apache2-doc.conf
-rw-r--r-- 1 root root  315 oct 10 20:59 charset.conf
-rw-r--r-- 1 root root 3224 oct 10 20:59 localized-error-pages.conf
-rw-r--r-- 1 root root  189 oct 10 20:59 other-vhosts-access-log.conf
-rw-r--r-- 1 root root 2174 oct 10 20:59 security.conf
-rw-r--r-- 1 root root  455 oct 10 20:59 serve-cgi-bin.conf
pepe@ubserver02:~$ _
```

/etc/apache2/conf-enabled/

```
pepe@ubserver02:~$ ll /etc/apache2/conf-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov 28 10:10 ./
drwxr-xr-x 8 root root 4096 nov 28 10:08 ../
lrwxrwxrwx 1 root root   34 nov 28 10:10 apache2-doc.conf -> ../conf-available/apache2-doc.conf
lrwxrwxrwx 1 root root   30 nov 28 10:08 charset.conf -> ../conf-available/charset.conf
lrwxrwxrwx 1 root root   44 nov 28 10:08 localized-error-pages.conf -> ../conf-available/localized-e
rror-pages.conf
lrwxrwxrwx 1 root root   46 nov 28 10:08 other-vhosts-access-log.conf -> ../conf-available/other-vho
sts-access-log.conf
lrwxrwxrwx 1 root root   31 nov 28 10:08 security.conf -> ../conf-available/security.conf
lrwxrwxrwx 1 root root   36 nov 28 10:08 serve-cgi-bin.conf -> ../conf-available/serve-cgi-bin.conf
pepe@ubserver02:~$ _
```

/etc/apache2/envars

```
 GNU nano 2.9.3                          /etc/apache2/envvars

# envvars - default environment variables for apache2ctl

# this won't be correct after changing uid
unset HOME

# for supporting multiple apache2 instances
if [ "${APACHE_CONFDIR##/etc/apache2-}" != "${APACHE_CONFDIR}" ] ; then
        SUFFIX="-${APACHE_CONFDIR##/etc/apache2-}"
else
        SUFFIX=
fi

# Since there is no sane way to get the parsed apache2 config in scripts, some
# settings are defined via environment variables and then used in apache2ctl,
# /etc/init.d/apache2, /etc/logrotate.d/apache2, etc.
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
# temporary state file location. This might be changed to /run in Wheezy+1
export APACHE_PID_FILE=/var/run/apache2$SUFFIX/apache2.pid
export APACHE_RUN_DIR=/var/run/apache2$SUFFIX
export APACHE_LOCK_DIR=/var/lock/apache2$SUFFIX
# Only /var/log/apache2 is handled by /etc/logrotate.d/apache2.
export APACHE_LOG_DIR=/var/log/apache2$SUFFIX

## The locale used by some modules like mod_dav
export LANG=C
## Uncomment the following line to use the system default locale instead:
#. /etc/default/locale

export LANG

## The command to get the status for 'apache2ctl status'.
```

Estado del servicio

Comprobación de que el servidor está INICIADO

Podemos comprobar listando los procesos en ejecución y filtrando con pipes, que existen varios procesos Apache abiertos, síntoma de que está en marcha el servidor.

```
pepe@ubserver02:~$ ps aux | grep apache2
root      2445  0.0  0.4  73960  4464 ?        Ss   10:08   0:00 /usr/sbin/apache2 -k start
www-data  2447  0.0  0.4 826256  4524 ?        Sl   10:08   0:00 /usr/sbin/apache2 -k start
www-data  2448  0.0  0.4 826256  4524 ?        Sl   10:08   0:00 /usr/sbin/apache2 -k start
pepe      2629  0.0  0.0  16948   968 tty1     S+   10:09   0:00 grep --color=auto apache2
pepe@ubserver02:~$
```

Arrancar, parar y estado del servicio

Existen scripts en /etc/init.d/ que nos permiten arrancar, parar y reiniciar el servicio apache2, entre otras cosas.

```
pepe@ubserver02:~$ sudo /etc/init.d/apache2 stop
[ ok ] Stopping apache2 (via systemctl): apache2.service.
pepe@ubserver02:~$ sudo /etc/init.d/apache2 start
[ ok ] Starting apache2 (via systemctl): apache2.service.
pepe@ubserver02:~$ sudo /etc/init.d/apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2018-11-28 10:17:31 CET; 5s ago
  Process: 2877 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
  Process: 2710 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
  Process: 2921 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 2938 (apache2)
    Tasks: 55 (limit: 1110)
   CGroup: /system.slice/apache2.service
           ├─2938 /usr/sbin/apache2 -k start
           ├─2951 /usr/sbin/apache2 -k start
           └─2952 /usr/sbin/apache2 -k start

nov 28 10:17:30 ubserver02 systemd[1]: Starting The Apache HTTP Server...
nov 28 10:17:31 ubserver02 apachectl[2921]: AH00558: apache2: Could not reliably determine the…ssage
nov 28 10:17:31 ubserver02 systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
pepe@ubserver02:~$
```

Reiniciar y recargar servicio

Diferencia:

- Restart: para e inicia el servicio
- Reload: el servicio sigue funcionando y se vuelven a leer los archivos de configuración

```
pepe@ubserver02:~$ sudo /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
pepe@ubserver02:~$ sudo /etc/init.d/apache2 reload
[ ok ] Reloading apache2 configuration (via systemctl): apache2.service.
pepe@ubserver02:~$ sudo /etc/init.d/apache2 force-reload
[ ok ] Reloading apache2 configuration (via systemctl): apache2.service.
pepe@ubserver02:~$
```

Reinicio del servidor con apachectl

Apachectl es una interfaz para el servidor del Protocolo de transferencia de hipertexto de Apache (HTTP). Está diseñado para ayudar al administrador a controlar el funcionamiento del demonio httpd de Apache.

El programa /usr/sbin/apache2ctl es el programa de línea de comandos para manejar Apache en Ubuntu. La sintaxis del comando es bastante sencilla. También se puede lograr la misma funcionalidad con el script de inicio apache2. Por lo tanto, los tres comandos que hacen la misma cosa son los siguientes:

- sudo apache2ctl restart
- sudo /etc/init.d/apache2 restart

- sudo service apache2 restart

Parar servicio

```
pepe@ubserver02:~$ sudo apachectl stop
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0
.1.1. Set the 'ServerName' directive globally to suppress this message
pepe@ubserver02:~$ _
```

Arrancar servicio

```
pepe@ubserver02:/$ sudo apachectl start
Invoking 'systemctl start apache2'.
Use 'systemctl status apache2' for more info.
pepe@ubserver02:/$ _
```

Puertos

Comprobación de puertos

- El protocolo HTTP tiene asociado el puerto 80 por defecto
- Podemos comprobar con netstat que el servidor está escuchando peticiones en ese puerto

```
pepe@ubserver02:~$ sudo netstat -atunp | grep apache2
tcp6       0      0 :::80                   :::*                    ESCUCHAR    3635/apache2
pepe@ubserver02:~$ sudo netstat -atunp
Conexiones activas de Internet (servidores y establecidos)
Proto  Recib Enviad Dirección local         Dirección remota       Estado       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               ESCUCHAR    499/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               ESCUCHAR    756/sshd
tcp6       0      0 :::22                   :::*                    ESCUCHAR    756/sshd
tcp6       0      0 :::80                   :::*                    ESCUCHAR    3635/apache2
udp     4608      0 127.0.0.53:53           0.0.0.0:*                           499/systemd-resolve
udp        0      0 192.168.1.18:68         0.0.0.0:*                           465/systemd-network
pepe@ubserver02:~$ _
```

Configuración de puertos

El archivo **ports.conf** contiene la información necesaria para que Apache2 escuche en direcciones IP y puertos específicos.

Cuando Apache2 se inicia comienza a esperar peticiones entrantes en determinados puertos y direcciones de la máquina en la que se está ejecutando.

Sin embargo, si se quiere limitar la escucha a:

- Un determinado puerto (o varios)
- En unas determinadas direcciones
- En una combinación de ambos

Hay que especificarlo en los archivos de configuración

Utilización de puertos para crear hosts virtuales

Esto se puede, además, combinar con la posibilidad de usar hosts virtuales, funcionalidad con la que un servidor Apache puede responder a peticiones en diferentes direcciones IP, diferentes nombres de hosts y diferentes puertos.

Consultar el fichero /etc/apache2/ports.conf

- Comprobar que están habilitados los servidores virtuales por nombre en todas
- las direcciones IP y en el puerto 80

Directiva Listen

La directiva Listen indica al servidor que acepte peticiones entrantes solamente en los puertos y en las combinaciones de puertos y direcciones IP que se especifiquen.

Si sólo se especifica un número de puerto en la directiva Listen el servidor escuchará en ese puerto, a través de todas las interfaces de red de la máquina.

Si se especifica una dirección IP y un puerto, el servidor escuchará solamente en la interfaz de red a la que pertenezca esa dirección IP y solamente en el puerto indicado.

Se pueden usar varias directivas Listen para especificar varias direcciones IP y puertos de escucha. El servidor responderá a las peticiones de todas las direcciones y puertos que se incluyan.

Ejemplos

Por ejemplo, para hacer que el servidor acepte conexiones tanto en el puerto 80 como en el puerto 8000, desde cualquier interfaz, se puede usar:

- Listen 80
- Listen 8000

Para hacer que el servidor acepte conexiones en dos interfaces de red y puertos específicos, usar

- Listen 192.168.1.1:80
- Listen 192.168.1.5:8000

```
 GNU nano 2.9.3                          /etc/apache2/ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

En este caso, Apache escuchará en todas las IP de sus adaptadores:

- Abre el puerto 80
- Si el módulo ssl_module está cargado, también abre el 443
- Si el módulo mod_gnutls.c está cargado, también abre el 443

Configuración de cortafuegos

Recordemos que Ubuntu Server trae firewall por defecto. Conviene tenerlo en cuenta, puesto que nuestro servidor va a escuchar en un puerto concreto, y tiene que poder atravesar el Firewall.

- Al instalar algunos paquetes, se crean perfiles de aplicación con reglas predefinidas para el cortafuegos UFW.
- Comprobamos que se crean 3 apps para Apache

```
pepe@ubserver02:~$ sudo ufw app list
Aplicaciones disponibles:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
```

Información de la configuración de app

```
pepe@ubserver02:~$ sudo ufw app info Apache
Perfil: Apache
Título: Web Server
Descripción: Apache v2 is the next generation of the omnipresent Apache web
server.

Puerto:
  80/tcp
```

Sitios web

Configuración de sites

Un mismo servidor puede alojar más de un sitio web diferente. Para cada uno de ellos, podemos crear un site para, entre otras cosas:

- Tener una configuración específica diferente para cada sitio web
- Habilitar y deshabilitarlos de forma separada.

Sites disponibles

En Apache podemos crear tantos sitios web como queramos. Todos los sitios (sites) están definidos por archivos en la carpeta de sitios disponibles. Podemos ver el archivo de configuración (**.conf**) de cada site en la ruta /etc/apache2/sites-available/. Por defecto vienen dos sites de ejemplo creados:

Mostrar carpeta de sitios disponibles

```
pepe@ubserver02:/var/www$ ll /etc/apache2/sites-available/
total 20
drwxr-xr-x 2 root root 4096 nov 28 10:08 ./
drwxr-xr-x 8 root root 4096 nov 28 10:08 ../
-rw-r--r-- 1 root root 1332 oct 10 20:59 000-default.conf
-rw-r--r-- 1 root root 6338 oct 10 20:59 default-ssl.conf
pepe@ubserver02:/var/www$
```

Estos sites que aparecen en esta carpeta están creados y disponibles, pero no significa que se estén ejecutando. Para ello hay que hacer algunos cambios. Como podemos comprobar, vienen 2 sites creados a modo de ejemplo.

Ver el contenido del sitio por defecto

Abrimos el primer site y podemos ver la configuración del mismo.

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName www.dmorenoweb.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Sites habilitados

Para activar (habilitar) un site y que se pueda ver:

- Se ejecuta el comando a2ensite mysite (donde el archivo se llama mysite.conf)
- El sistema crea un enlace simbólico en el sistema de archivos en la carpeta de sitios habilitados (sites-enabled).
- Los sites que aparezcan en esta carpeta estarán activados y son **visibles** a priori.

Podemos comprobar que en /etc/apache2/sites-enabled existe el fichero **000-default.conf** y es un enlace simbólico al que se encuentra en **sites-available**

```
pepe@ubserver02:/var/www$ ll /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov 28 10:08 ./
drwxr-xr-x 8 root root 4096 nov 28 10:08 ../
lrwxrwxrwx 1 root root   35 nov 28 10:08 000-default.conf -> ../sites-available/000-default.conf
pepe@ubserver02:/var/www$ _
```

Habilitar y deshabilitar sites

Para habilitar o deshabilitar un sitio alojado con Apache, puede usar los comandos 'a2ensite' y 'a2dissite', respectivamente. Ambos comandos usan esencialmente la misma sintaxis:

- a2ensite <site>
- a2dissite <site>

donde '<site>' es el nombre del archivo de configuración del host virtual de su sitio, ubicado en /etc/apache2/sites-available/, menos la extensión '.conf'.

Crear nuestro propio site

Vamos a copiar el site de ejemplo y vamos a crear uno nuestro, que llevará como nombre el nombre de nuestro subdominio.

```
root@ubserver01:/etc/apache2/sites-available -> ll
total 16K
-rw-r--r-- 1 root 1,4K jun 11  2018 000-default.conf
-rw-r--r-- 1 root 6,2K jun 11  2018 default-ssl.conf
-rw-r--r-- 1 root  230 dic 12 13:14 smx2.org.conf
root@ubserver01:/etc/apache2/sites-available ->
```

Lo vamos a modificar para que coincida con nuestra configuración.

```
<VirtualHost *:80>
        ServerAdmin dmoreno@iesfbmoll.org
        ServerName smx2.org
        DocumentRoot /var/www/smx2.org/

        ErrorLog ${APACHE_LOG_DIR}/error_smx2.org.log
        CustomLog ${APACHE_LOG_DIR}/access_smx2.org.log combined
</VirtualHost>
```

Necesitaremos crear la carpeta /var/www/xxx.smx2.org/ y copiar dentro un archivo HTML

Habilitar el nuevo site

Deshabilitamos el site por defecto y recargamos

```
root@ubserver01:/etc/apache2 -> cd sites-enabled/
root@ubserver01:/etc/apache2/sites-enabled -> ll
total 0
lrwxrwxrwx 1 root 35 oct 15 11:47 000-default.conf -> ../sites-available/000-default.conf
root@ubserver01:/etc/apache2/sites-enabled -> a2dissite 000-default
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@ubserver01:/etc/apache2/sites-enabled -> systemctl reload apache2
root@ubserver01:/etc/apache2/sites-enabled ->
```

Habilitamos el site nuevo que hemos creado:

```
root@ubserver01:/etc/apache2/sites-enabled -> a2ensite smx2.org
Enabling site smx2.org.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@ubserver01:/etc/apache2/sites-enabled -> systemctl reload apache2
root@ubserver01:/etc/apache2/sites-enabled -> ll
total 0
lrwxrwxrwx 1 root 32 dic 12 13:19 smx2.org.conf -> ../sites-available/smx2.org.conf
root@ubserver01:/etc/apache2/sites-enabled ->
```

Accedemos desde un navegador para ver que funciona todo correctamente.

Comprobación de log

En el archivo de log veremos que la petición se ha guardado correctamente dos peticiones. Una para el archivo html, y otra para enviar la imagen que necesita el html para mostrarse correctamente.

Por otro lado, podemos ver los códigos de estado y la IP del ordenador que ha hecho la solicitud

```
root@ubserver01:/var/log/apache2 -> cat access_smx2.org.log
192.168.1.2 - - [12/Dec/2018:13:20:36 +0100] "GET / HTTP/1.1" 200 2671 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0"
192.168.1.2 - - [12/Dec/2018:13:20:36 +0100] "GET /favicon.ico HTTP/1.1" 404 504 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0"
root@ubserver01:/var/log/apache2 ->
```

Acceder desde el navegador al servidor

Comprobamos desde un navegador de un equipo que tenga acceso al servidor que se abre la página por defecto. Esto lo podemos hacer de varias formas:

Utilizando la interfaz de red local

Si ejecutamos el navegador en la misma máquina en la que está corriendo el servidor podemos hacer lo siguiente:

- http://localhost
- http://127.0.0.1

Utilizando la IP del servidor

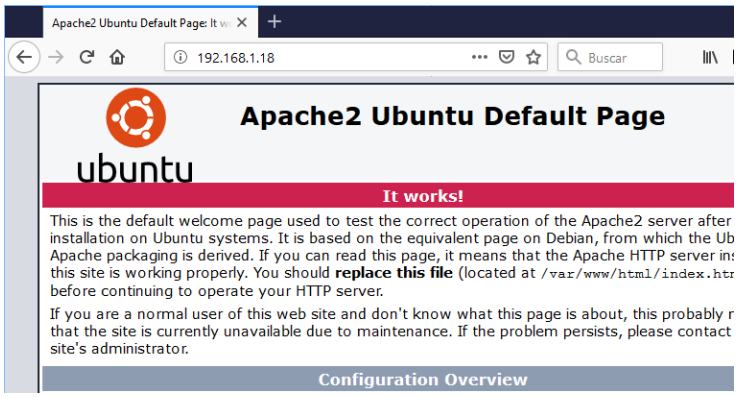
- Si estamos en otra máquina diferente a la del servidor, cambiando la IP por la del servidor donde se encuentra instalado Apache
- Si el servidor se encuentra tras una red NAT, deberemos redireccionar los puertos.
- **Ejemplo:** http://172.6.254.1

Utilizando un nombre de dominio

Si existe un servidor DNS o tenemos configurado el archivo de host local. Si no existe ninguna de estas dos configuraciones, deberemos conectar utilizando la IP

- http://smx2.org

En cualquier caso, al principio y hasta que lo cambiemos, el navegador nos mostrará la página que trae por defecto Apache, la cual podremos cambiar posteriormente.



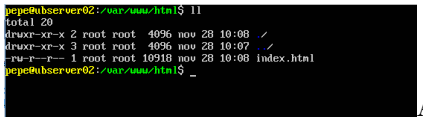
Configurar un hostname para el servidor

Editar /etc/hostname y /etc/hosts configurar el nombre FQDN del equipo.

```
pepe@ubserver02:~$ cat /etc/hostname /etc/hosts
ubserver02
127.0.0.1       localhost
127.0.1.1       ubserver02

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
pepe@ubserver02:~$
```

Crear estructura de páginas web

Comprobar archivo de índice



Acceder a /var/www/xxx.smx2.org/ y comprobar de que existe index.html. Este es el archivo que se sirve por defecto si no se especifica en el navegador ninguno.

Crear más archivos y carpetas

A continuación, vamos a crear más archivos y directorios para hacer pruebas.

Crear archivo red.html

Crear el fichero de texto /var/www/xxx.smx2.org/red.html

```
<html>
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
</head>
<body>
    <div align="center">
        <hl>ESTE ES EL SERVIDOR WEB DE LA RED<br>
        smx2.org</hl>
    </div>)
</body>
</html>
```

Crear un nuevo directorio

Crear el directorio /var/www/xxx.smx2.org/datos/ y dentro colocar el archivo datos1.html

Crear el archivo datos.html

```
<html>
  <head>
    <meta http-equiv="content-type' content="text/html; charset=utf-8'/>
  </head>
  <body>
    <hl><b>MIS DATOS SON:</b></hl>
    <ol>
      <li><b>Nombre:</b> Daniel Moreno Rosselló</li>
      <li><b>Curso:</b> 2º SMX </li>
      <li><b>Asignatura:</b> Servicios de Red</li>
    </ol>
  </body>
</html>
```

Comprobar funcionamiento desde cliente

Podemos comprobar si la web está funcionando correctamente desde un cliente con interfaz gráfica, que disponga de un navegador. No importa el SO del que se trate, siempre que tenga conexión con el servidor y se pueda comunicar con el

Acceder con IP

Podemos acceder a un servidor HTTP tanto a través de su IP como de su nombre de dominio. En este primer caso, podemos visitar http://172.168.254.1. Tendremos que poner en nuestro caso la IP del servidor. Si está tras una red NAT, deberemos redireccionar los puertos.

Acceder con nombre de dominio

También podemos comprobar que podemos navegar utilizando el nombre de dominio del servidor: http://smx2.org/red.html

Para qué funcione, en algún sitio se tiene que resolver la IP correspondiente al nombre de dominio. Lo podemos hacer en nuestra máquina localmente, o a través de un servidor DNS.

Para esta opción, deberemos modificar el servidor DNS y agregar la siguiente entrada:

www IN A 192.168.0.10

Esto no hace falta hacerlo porque ya está configurada la zona en el servidor

Abrir páginas desde el shell

Comprobar con el navegador las URLs

- http://localhost
- http://172.6.254.1/red.html
-

http://smx2.org/datos/datos.htmlSi no disponemos de navegador, podemos utilizar WGET. Wget es un paquete de software gratuito para recuperar archivos usando HTTP, HTTPS, FTP y FTPS.

```
usuario@ubserver02:~$ wget http://localhost
--2018-12-02 13:05:13--  http://localhost/
Resolviendo localhost (localhost)... ::1, 127.0.0.1
Conectando con localhost (localhost)[::1]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 10918 (11K) [text/html]
Guardando como: “index.html”

index.html              100%[===================================>]  10,66K  --.-KB/s    en 0s

2018-12-02 13:05:13 (383 MB/s) - “index.html” guardado [10918/10918]
```

Navegador en shell

Podemos también utilizar un navegador para shell, como Lynx, si queremos ver de una manera aproximada como queda la página.

Secciones

Las directivas presentes en los ficheros de configuración pueden ser de aplicación para todo el servidor, o puede que su aplicación se limite solamente a determinados directorios, ficheros, hosts, o URLs.

Las secciones de configuración y los ficheros .htaccess para modificar el ámbito de aplicación de las directivas de configuración.

Tipos de secciones

Existen **dos tipos básicos** de secciones de configuración.

- Por un lado, la mayoría de las secciones de configuración **se evalúan para cada petición** que se recibe y se aplican las directivas que se incluyen en las distintas secciones solamente a las peticiones que se adecúan a determinadas características.
- Por otro lado, las secciones de tipo <IfDefine> e <IfModule>, se evalúan **solamente al inicio o reinicio del servidor**.

Si al iniciar el servidor las condiciones son las adecuadas, las directivas que incluyen estas secciones se aplicarán a todas las peticiones que se reciban. Es caso contrario, esas directivas que incluyen se ignoran completamente

Sección directory

La sección directory engloba una o más directivas de configuración que sólo se aplican al directorio y subdirectorios especificados.

Se especifica: <Directory /path/a/directorio> / </Directory>

El argumento /path/a/directorio puede ser:

- Un nombre de directorio (ruta absoluta)
- Una expresión regular

Dentro de una sección podemos incluir tantas directivas como queramos.

Ejemplo de configuración de directory

Este es un ejemplo de configuración, no hay que hacerlo. Las líneas tomadas del archivo /etc/apache2/apache2.conf. Partes importantes:

- Options especifica Indexes (mostrará el contenido de la carpeta si es necesario)
- AllowOverride indica si se pueden sobrescribir las opciones de esta carpeta a través de un archivo externo (.htaccess). Lo veremos luego.
- Allow from all nos indica que cualquiera puede acceder al contenido de esta carpeta desde el navegador.

<Directory "/usr/share/apache2/icons">

Options Indexes MultiViews

AllowOverride None

Order allow,deny

Allow from all

</Directory>

Permitir solo acceso a directorio a una red concreta

Crear un directorio llamado clase en /var/www/xxx.smx2.org/. En este directorio:

- Denegar la entrada a todo el mundo
- Permitir a los ordenadores de nuestra red.

Si hacemos esto, todos los equipos desde 172.16.254.1 hasta 172.16.254.255 podrán ver la carpeta. El testo, no. Para ello editamos el archivo **apache2.conf**

```
<Directory /var/www/smx2.org/clase>
        Order deny,allow
        Deny from all
        Allow from 172.16.254.0/24
</Directory>
```

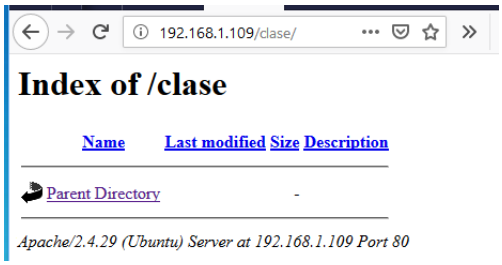
Si probamos a acceder desde el navegador desde un ordenador fuera de esta red:

Permitir solo acceso a directorio a un equipo concreto

Puedo hacer que solo yo pueda ver esta carpeta desde un equipo en concreto:

```
<Directory /var/www/smx2.org/clase>
        Order deny,allow
        Deny from all
        Allow from 192.168.1.2
</Directory>
```

En este caso, el equipo 192.168.1.2 será el único que tenga acceso a la carpeta.



Sección files

Las directivas incluidas en una sección Files se aplican al archivo especificado sin tener en cuenta en qué directorio se encuentra. Es decir, la directiva Files limita el ámbito de aplicación de las directivas que incluye según el nombre de los archivos.

Denegar acceso a un archivo concreto

En el ejemplo siguiente las directivas de configuración, cuando se colocan en la sección principal del archivo de configuración, deniegan el acceso:

A cualquier archivo llamado **privado.html** de cualquier carpeta

```
<Files privado.html>
        Order allow,deny
        Deny from all
</Files>
```

Podéis crear varios archivos privado.html en diferentes carpetas y comprobar que no podemos acceder a través del navegador a ninguno de ellos.

Utilización conjunta de directory y files

También se pueden utilizar de forma conjunta las secciones <Directory> y <Files> para referirse a archivos que se encuentran en un determinado lugar del sistema de archivos. Por ejemplo:

- Denegará el acceso a /var/www/xxx.smx2.org/clase/privado.html
- Cualquier otra aparición de privado.html que se encuentre en /var/www/xxx.smx2.org/o cualquiera de sus subdirectorios si que se mostrará.

```
<Directory /var/www/smx2.org/clase>
        <Files privado.html>
                Order allow,deny
                Deny from all
        </Files>
</Directory>
```

Especificar un conjunto de archivos al que aplicará una configuración

El argumento de <Files> puede ser un nombre de archivo, o una expresión regular. Por ejemplo, para indicar que el argumento es cualquier archivo con formato gif, jpg, jpeg o png utilizaríamos la **siguiente** expresión regular:

<Files ~ "\.(gif|jp?g|png)$">

donde

- '?' sustituye 0 o una ocurrencia de la expresión regular anterior (1 carácter)
- '*' equivale a cualquier secuencia de caracteres
- '$' indica final de línea
- '\' se utiliza para escapar el carácter '.'

Se puede utilizar la sección FilesMatch para trabajar con expresiones regulares.

Sección location

La sección Location está relacionada con el espacio web y cambia la configuración para el contenido del espacio web. Por ejemplo:

<Location /privado>

Order Allow,Deny

Deny from all

</Location>

Esta configuración evita que se acceda a cualquier URL que empiece por /privado, utilizándose la palabra privado como expresión regular. Por ejemplo, se aplicaría a peticiones que comiencen o contengan /privado en su URL (http://localhost/privado123).

Sección ifmodule

La sección IfModule engloba directivas que se procesarán si el módulo dado como argumento se compila con Apache2.

Esta sección solo se comprueba al iniciar el servidor

Una de las directivas permitidas en IfModule es UserDir.

<IfModule mod_userdir.c>

Userdir directorio

</IfModule>

En este caso, si el módulo **usedir** está activado, se aplica la directiva de dentro. En cambio, si no está activado, se ignora.

Sección ifdefine

IfDefine engloba directivas que serán procesadas sólo si se cumple una determinada condición (test) al iniciar el servidor.

<IfDefine [!]nombre_parametro>

</IfDefine>

IfDefine se usa para marcar directivas que son condicionales.

Las directivas que hay dentro de una sección IfDefine se procesan sólo si el test devuelve un resultado positivo.

Si el test produce un resultado negativo todo lo que haya entre los marcadores de comienzo y final será ignorado.

El test puede ser:

- nombre-parámetro: las directivas se procesan sólo si el parámetro llamado nombre-parámetro está definido.
- !nombre-parámetro: hace lo contrario, y procesa las directivas sólo si nombre-parámetro no está definido.

Paso por parámetro

Podemos pasar parámetros por línea de comandos al ejecutar Apache, en lugar de incluirlos en un archivo de configuración. De esta forma, sólo se utilizarán esos parámetros cuando así se utilice en la llamada al servicio. El argumento nombre-parámetro se define cuando se ejecuta Apache2 por la línea de órdenes con la opción -Dparametro, al iniciar el servidor.

Ejemplo: apache2 -DReverseProxy ...

en el archivo apache2.conf

Como hemos pasado el argumento ReverseProxy, en la sección Ifdefine lo verá, y el servidor utilizará las directivas que hay dentro.

<IfDefine ReverseProxy>

LoadModule rewrite_module modules/mod_rewrite.so

LoadModule proxy_module modules/libproxy.so

</IfDefine>

Indica que se cargan los módulos mod_rewrite y libproxy, ya que, en el archivo de configuración existe la directiva IfDefine relativa al parámetro ReverseProxy con el que se ha lanzado la ejecución de Apache (-D).

Directivas

Los archivos de configuración de apache2 se encuentran en la carpeta /etc/apache2. El archivo principal de configuración es /etc/apache2/apache2.conf. Antes de realizar cualquier cambio en este archivo, es conveniente realizar una copia de seguridad del mismo ya que si apache encuentra algún error en el archivo de configuración, no arrancará.

Directivas

El servidor se configura colocando directivas de configuración en estos archivos de configuración. Una directiva es una palabra clave seguida de uno o más argumentos que establecen su valor.

Las directivas se pueden colocar en muchos sitios, generalmente se responde al considerar dónde desea que una directiva sea efectiva.

- Si se trata de una configuración global, debe aparecer en el archivo de configuración, fuera de cualquier <Directory>, <Location>, <VirtualHost> u otra sección.
- Si se aplica solo a un directorio en particular, debe ir dentro de una sección <Directory> que se refiera a ese directorio, y así sucesivamente.

Directiva ServerRoot

La directiva ServerRoot es el directorio raíz donde se almacenan los distintos ficheros que utiliza el servidor Apache salvo que se indiquen rutas absolutas como se indicaba con anterioridad.

Podemos comprobar la directiva en /etc/apache2/apache2.conf

```
#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# NOTE!  If you intend to place this on an NFS (or otherwise network)
# mounted filesystem then please read the Mutex documentation (available
# at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);
# you will save yourself a lot of trouble.
#
# Do NOT add a slash at the end of the directory path.
#
#ServerRoot "/etc/apache2"
```

Conexiones persistentes

KeepAlive indica si se permiten o no las conexiones persistentes, es decir más de una petición por conexión. Puede tomar los valores On u Off. El valor predeterminado es On.

```
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100

#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
```

Ficheros a servir por defecto (Directory Index)

Comprobar página por defecto

Acceder a http://IPdelServidor y mirar qué página se está sirviendo por defecto. Podemos utilizar el comando wget para descargar un contenido web a través de HTTP

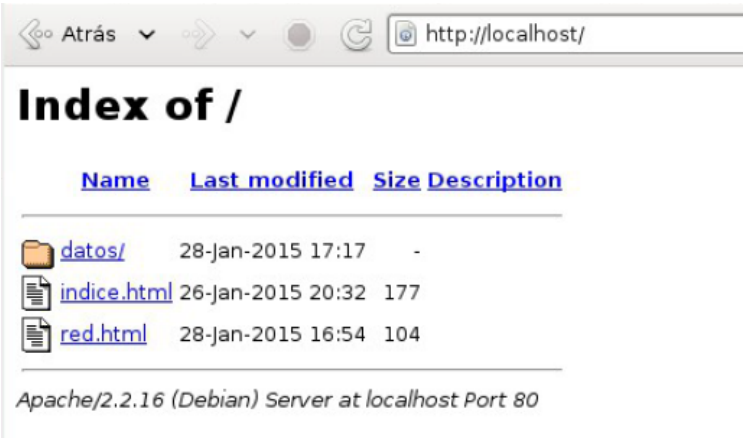
```
usuario@ubserver02:~$ wget localhost
--2018-12-02 14:58:02--  http://localhost/
Resolviendo localhost (localhost)... ::1, 127.0.0.1
Conectando con localhost (localhost)[::1]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 751 [text/html]
Guardando como: “index.html.1”

index.html.1            100%[=================================>]     751  --.-KB/s    en 0s

2018-12-02 14:58:02 (88,0 MB/s) - “index.html.1” guardado [751/751]
```

Renombrar fichero de índice

Renombrar el fichero index.html a indice.html

```
usuario@ubserver02:~$ sudo mv /var/www/html/index.html /var/www/html/indice.html
```

Acceder a http://IPdelServidor. Como no se encuentra index.html, muestra una lista con el contenido del directorio.



Cambiar directiva DirectoryIndex

Podemos configurar un índice de directorio para llamar a varios archivos usando el siguiente texto:

- Ejemplo: **DirectoryIndex** index.html index.cgi index.php

Las líneas anteriores le indican al servidor web Apache que:

1. muestre el archivo 'index.html' como el índice de directorio.
2. Si este archivo no está disponible, entonces muestre 'index.cgi'
3. Si no está disponible, entonces muestre 'index.php'.

Si no está disponible uno de los archivos especificados, el servidor web Apache volverá a su configuración predeterminada, ya sea mostrando un mensaje de error, un mensaje de lista de directorios no disponible o mostrando las listas de directorios de archivos y directorios

Editar /etc/apache2/sites-available/xxx.smx2.org.conf e incluir la directiva DirectoryIndex para que muestre como índice el archivo índice.html

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port tha
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName www.dmorenoweb.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        DirectoryIndex indice.html_
        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn
```

Reiniciamos el servidor: sudo systemctl restart apache2

Accediendo a la página web en http://IPdelServidor/, aplicando la directiva **DirectoryIndex**, podemos hacer que se muestra la página **red.html.**

Opciones sobre directorios

Editar /etc/apache2/sites-available/xxx.smx2.org.conf y modificar la siguiente sección directory, para hacer que el archivo a servir por defecto sea red.html

Aquí debajo cambiar /var/www/ por /var/www/xxx.smx2.org/

```
<Directory /var/www/>
        DirectoryIndex red.html
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
```

Tal como indica la sección, a no ser que otra diga lo contrario, todas las carpetas que cuelgan de /var/www/xxx.smx2.org

- Van a buscar un archivo red.html para servirlo como página por defecto (DirectoryIndex)
- De no encontrarlo, mostrarán el contenido del directorio (Options Indexes)

Comprobación

Cuando accedamos a http://IPdelServidor sin especificar ningún archivo, se nos mostrará el documento **red.html**.

Al acceder a la carpeta datos, http://IPdelServidor/datos/, al no existir ningún documento dentro llamado **red.html**, muestra el contenido del directorio.

Configurar la carpeta DATOS para mostrar index.html y no listar contenido de la carpeta

Imaginemos que no queremos que para la carpeta datos se aplique la configuración de /var/www. Deberíamos crear una nueva sección Directory para /var/www/datos. Todo lo que pongamos aquí, afectará ahora a la carpeta datos y todas sus subcarpetas. Si alguna directiva no la ponemos, aplicará las opciones de la carpeta padre (/var/www)

```
<Directory /var/www/datos>
        DirectoryIndex index.html
        Options FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
```

En este caso:

1. Hemos definido que el archivo por defecto en esta carpeta será index.html. Por lo tanto, este es el archivo que se servirá cuando accedamos a http://IPdelServidor/datos/
2. En la directiva Options ya no aparece Indexes. Por lo que cuando el servidor no encuentre los ficheros definidos en DirectoryIndex, no listará el contenido del directorio.

Reiniciamos el servidor: sudo systemctl restart apache2

Comprobaciones

Acceder a http://IPdelServidor/ muestra el archivo red.html.

Acceder a http://IPdelServidor/datos/ Al no existir el fichero red.html en /var/www/datos y no se permite el listado del directorio, el servidor retorna el código 403 Forbbiden. Nos muestra un mensaje de prohibido porque la carpeta datos SI existe, pero no nos la quiere enseñar.

Acceder a http://IPdelServidor/datos/**meloinvento.html.**En este caso lo que pasa es que no existe la página que pedimos, nos dará un mensaje de NOT FOUND

Logs (ErrorLog, CustomLog, LogFormat)

Se nos plantea la cuestión ¿y qué hacer cuando algo no funciona? En primer lugar, se debe consultar el registro de errores o logs: es posible que el problema pueda deducirse de un mensaje de error.

Los logs son archivos que guardan los datos de todos los acontecimientos que tienen que ver con un programa en concreto, anotando el momento en que ocurrió cada acción.

Consultar logs

Por defecto Apache crea sus propios logs, pese a que se pueden crear otros que sean más convenientes. Por defecto el archivo de registro de errores se encuentra en /var/log/apache2/error.log.

Se recomienda mostrar los archivos de registro en una consola mientras se accede al servidor para ver cómo reacciona éste en cada momento. Con este fin, ejecute en una consola el siguiente comando como root.

Ejecutar la orden: tail -f /var/log/apache2/*.log

Mientras mantengamos abierto el shell, se irán mostrando por pantalla solo las nuevas líneas que se añadan al log a partir del momento en que ejecutemos el comando, hasta que decidamos cancelarlo.

Consultar en internet

Otra opción que tenemos, partiendo de la información recogida en los logs, es la de consultar en diferentes sitios en Internet en los que podemos encontrar soluciones o recomendaciones, como por ejemplo:

- Consultar la base de datos de fallos en la página web http://httpd.apache.org/bug_report.html.
- Consultar en las listas de correo y los foros de noticias.
- La lista de correo para los usuarios de Apache: http://httpd.apache.org/userslist.html.
- Los foros de noticias, se recomienda comp.infosystems.www.servers.unix.

Conocer donde se guardan los archivos de log

El lugar donde se guardan los archivos de log está configurado dentro del fichero /etc/apache2/sites-available/xxx.smx2.org.conf

```
        DirectoryIndex indice.html
        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
```

En general, los ficheros de log se encuentran por defecto en /var/log/apache2/ (la localización habitual). $(APACHE_LOG_DIR) hace referencia a esta carpeta.

```
usuario@ubserver02:~$ ll /var/log/apache2/
total 52
drwxr-x---  2 root adm     4096 dic  4 18:32 ./
drwxrwxr-x 10 root syslog  4096 dic  7 11:03 ../
-rw-r-----  1 root adm    12197 dic  4 17:41 access.log
-rw-r-----  1 root adm    17425 dic  7 11:03 error.log
-rw-r-----  1 root adm        0 nov 28 10:08 other_vhosts_access.log
-rw-r--r--  1 root root     737 dic  4 18:45 seguro_access.log
-rw-r--r--  1 root root    1476 dic  7 11:03 seguro.error.log
usuario@ubserver02:~$ _
```

Consulta del fichero log de errores

Los archivos de log de apache tienen líneas muy largas y conviene visualizarlos con la pantalla completa. Podemos utilizar **grep** para hilar más fino y filtrar los logs, o bien utilizar **more** y **less** para navegar por el resultado.

El log de errores general se suele llamar error.log

```
root@ubserver02:~# cat /var/log/apache2/error.log | grep -i client
[Sun Dec 02 15:51:24.051112 2018] [auth_basic:error] [pid 2678:tid 140283515254528] [client 192.168.1.2:17663] AH01618: user dsds not found: /privado/
[Sun Dec 02 15:51:26.438252 2018] [auth_basic:error] [pid 2678:tid 140283371030272] [client 192.168.1.2:17663] AH01618: user dsds not found: /privado/
[Sun Dec 02 15:51:27.068623 2018] [auth_basic:error] [pid 2678:tid 140283362637568] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
[Sun Dec 02 15:51:27.558903 2018] [auth_basic:error] [pid 2678:tid 140283523647232] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
[Sun Dec 02 15:51:27.732699 2018] [auth_basic:error] [pid 2678:tid 140283354244864] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
[Sun Dec 02 15:51:27.899326 2018] [auth_basic:error] [pid 2678:tid 140283345852160] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
[Sun Dec 02 15:51:28.058966 2018] [auth_basic:error] [pid 2678:tid 140283532039936] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
[Sun Dec 02 15:51:28.229091 2018] [auth_basic:error] [pid 2678:tid 140283540432640] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
[Sun Dec 02 15:51:28.390953 2018] [auth_basic:error] [pid 2678:tid 140283312281344] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
[Sun Dec 02 15:51:28.561169 2018] [auth_basic:error] [pid 2678:tid 140282825799424] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
[Sun Dec 02 15:51:28.735459 2018] [auth_basic:error] [pid 2678:tid 140282817406720] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
[Sun Dec 02 15:51:28.909501 2018] [auth_basic:error] [pid 2678:tid 140282809014016] [client 192.168.1.2:17663] AH01618: user  not found: /privado/
```

Consulta del fichero log de accesos

En este archivo se guardan los accesos que se han realizado al servidor junto con las peticiones y respuestas.

El log de accesos se suele llamar accesos.log

```
192.168.1.2 - "" [02/Dec/2018:15:51:27 +0100] "GET /privado/ HTTP/1.1" 401 726 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36"
192.168.1.2 - "" [02/Dec/2018:15:51:27 +0100] "GET /privado/ HTTP/1.1" 401 726 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36"
192.168.1.2 - "" [02/Dec/2018:15:51:27 +0100] "GET /privado/ HTTP/1.1" 401 726 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36"
192.168.1.2 - "" [02/Dec/2018:15:51:27 +0100] "GET /privado/ HTTP/1.1" 401 726 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36"
192.168.1.2 - "" [02/Dec/2018:15:51:28 +0100] "GET /privado/ HTTP/1.1" 401 726 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36"
192.168.1.2 - "" [02/Dec/2018:15:51:28 +0100] "GET /privado/ HTTP/1.1" 401 726 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36"
```

Como podemos ver en este caso, las últimas solicitudes han sido peticiones GET a la URL /privado/. Al no tener permiso, el servidor nos ha contestado con un "Error HTTP **401** No autorizado"

Códigos de error (ErrorDocument)

En el caso que suceda un error porque el documento solicitado no existe, se nos mostrará un aviso en el navegador ligado al error HTTP 404.

Configurar directiva ErrorDocument

Configurar mensaje

Podemos configurar el servidor virtual por defecto cambiando la directiva ErrorDocument para que presente un texto de aviso.

Para ello se debe modificar /etc/apache2/sites-available/xxx.smx2.org.conf para que cuando retorne el código de error 404 (página no encontrada) envíe el texto "Página no encontrada en el servidor de la red".

```
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

ErrorDocument 404 "Página no encontrada en servidor de la red iesdmoreno.org"

Alias /wiki /home/alumno/wiki
<Directory /home/alumno/wiki>
        AllowOverride All
</Directory>
```

Reiniciamos el servidor: sudo systemctl restart apache2

Podemos comprobar que funciona accediendo a una página cualquiera inexistente, por ejemplo http://IPdelServidor/noesta.html. Así podemos forzar un error *not found 404* que desencadene la configuración que hemos hecho.

Nos debería mostrar el mensaje personalizado para el error.

Crear archivo para error no encontrado

En lugar de una frase, quedaría mejor mostrar un documento HTML cada vez que no se encuentre un archivo. Para ello, podemos configurar el servidor virtual por defecto cambiando la directiva ErrorDocument para que presente una página de aviso en un archivo html que tengamos en el servidor.

Crear fichero personalizado

Crearemos el archivo en /var/www/xxx.smx2.org/no_encontrada.html y escribiremos su contenido en lenguaje HTML

```
 GNU nano 2.9.3            /var/www/html/no_encontrada.html            Modificado

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"/> </head>
<body>
<h1><b>error 404</b></h1>
<ol>
<li><b>Causa:</b> Página no encontrada.</li>
<li><b>Red</b> iesdmoreno.org</li>
</ol>
</body>
</html>
```

Modificar archivo de configuración del site

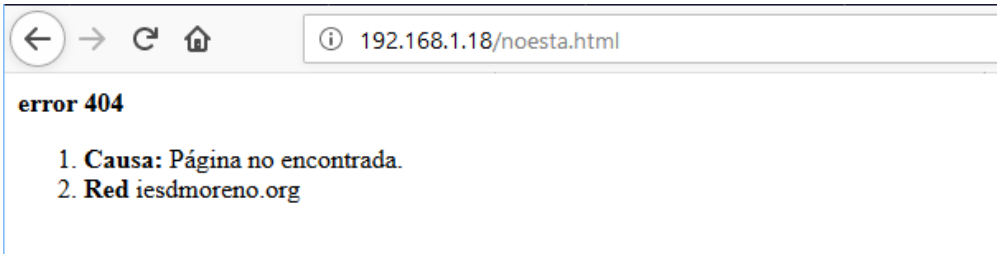
Deberemos de modificar el archivo de configuración /etc/apache2/sites-available/xxx.smx2.org.conf para que cuando retorne el código de error 404 (página no encontrada) presente el contenido del archivo /var/www/no_encontrada.html

```
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        ErrorDocument 404 /no_encontrada.html
```

Reiniciamos el servidor: sudo systemctl restart apache2

Probar que se muestra el archivo creado

Podemos acceder a una página que no exista, como http://IPdelServidor/noesta.html y comprobar que se muestra el archivo creado para este fin.



Directorios virtuales

Directorios Virtuales (Directiva Alias)

Podemos hacer que se pueda acceder a otras carpetas fuera de la carpeta /var/www utilizando alias. Estos alias, redirigirán una ruta de la URL a una carpeta concreta en el servidor.

- Por defecto, http://IPdelServidor/wiki lleva a la carpeta /var/www/wiki.
- También podríamos alojar nuestros documentos en otra carpeta, como por ejemplo /home/alumno/wiki, y podríamos conseguir, por ejemplo, que

http://IPdelServidor/wiki/ redirigiera las peticiones a esta carpeta.

Para poder conseguir esto, una de las estrategias posibles es la de utilizar alias. Viene a ser un acceso directo que nos lleva de una carpeta a otra.

Iniciar sesión como usuario alumno y crear directorio

Pasos

- Crear una cuenta llamada **alumno** e iniciar sesión con este usuario
- Crear la carpeta /home/alumno/wiki
- Dentro crear una página **wiki1.html** que contenga "PÁGINA WIKI 1"

```
alumno@ubserver02:/$ mkdir /home/alumno/wiki
alumno@ubserver02:/$ touch /home/alumno/wiki/wiki1.html
```

Documento /home/alumno/wiki/wiki1.html

```
<html>
        <head>
                <meta http-equiv="content-type" content="text/html; charset=utf-8">
        </head>
        <body>
                <div align="center">
                        <h1>PAGINA WIKI 1</h1>
                </div>
        </body>
</html>
```

Iniciar sesión como administrador y crear el Alias /wiki

Gracias al alias, cuando visitemos http://IPdelServidor/wiki, nos redirigirá al contenido de la carpeta /home/alumno/wiki.

```
Alias /wiki /home/alumno/wiki
<Directory /home/alumno/wiki>
        DirectoryIndex index.html
        Options Indexes FollowSymLinks Multiviews
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
```

Además, deberemos buscar y cambiar en la sección directory / (raíz) de apache2.conf para que ponga require from all. Sino no funcionará

Reiniciamos el servidor: sudo systemctl restart apache2

Acceder a la página wiki

http://IPdelServidor/wiki

Nos daremos cuenta de que aparece el listado del directorio /home/alumno/wiki ya que la directiva **DirectoryIndex** está en **index.html** y al no existir el archivo index.html, siguiendo la directiva Indexes lista el directorio.

Directorios Virtuales (usando enlaces simbólicos)

Un **enlace simbólico** es un tipo de archivo especial que contiene una ruta a otro archivo. De esta forma, cuando accedemos a un enlace simbólico desde un editor de texto u otro programa, lo redirige al archivo indicado por la ruta del enlace.

Sintaxis:

- **Para crearlo:** ln -s <destination file or directory> <name of the symlink>
- **Para borrarlo:** rm <name of the symlink>

Nota: Al borrar el enlace simbólico no se borra el archivo destino

Pasos a realizar:

1. Iniciar sesión como usuario alumno.
2. Crear directorio /home/alumno/blog/
3. Dentro crear el archivo blog1.html con el texto "PÁGINA BLOG 1"

```
alumno@ubserver02:/$ mkdir /home/alumno/blog
alumno@ubserver02:/$ nano /home/alumno/blog1.html_
```

4. Iniciar sesión como administrador
5. Si no tenemos a **alumno** como *sudoer*, desde una cuenta que lo permita
   a. Ejecutar sudo adduser alumno sudo
6. Crear el enlace simbólico /var/www/blog que apunte a /home/alumno/blog
   b. Ejecutar sudo ln -s /home/alumno/blog /var/www/blog
7. A continuación, podemos ver el enlace creado

```
alumno@ubserver02:~/blog$ ll /var/www/
total 12
drwxr-xr-x  3 root root 4096 dic  8 11:18 ./
drwxr-xr-x 14 root root 4096 nov 28 10:07 ../
lrwxrwxrwx  1 root root   18 dic  8 11:18 blog -> /home/alumno/blog/
drwxr-xr-x  4 root root 4096 dic  7 11:27 html/
```

Modificar /etc/apache2/sites-available/xxx.smx2.org.conf y configurar el sitio para el directorio /home/alumno/blog

```
<Directory /var/www/blog>
        DirectoryIndex index.html
        Options Indexes FollowSymLinks Multiviews
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
```

Opción followsymlinks

Verificar que está definida la opción FollowSymLinks en la configuración del directorio /var/www

```
<Directory /var/www/>
        DirectoryIndex red.html
        Options Indexes FollowSymLinks Multiviews
        AllowOverride None
        Order allow,deny
        allow from all
```

Reiniciamos el servidor: sudo systemctl restart apache2

Acceder a http://IPdelServidor/blog

Módulos

Apache es un servidor **modular**. Esto implica que en el servidor básico se incluyen únicamente las **funcionalidades más básicas**. Otras funcionalidades se encuentran disponibles a través de módulos que pueden ser cargados por Apache.

Por defecto, durante la compilación se incluye en el servidor un juego de módulos base.

- Si el servidor se compila para usar **carga dinámica** de módulos, entonces los módulos pueden ser compilados por separado, e incluidos en cualquier momento usando la directiva LoadModule.
- En caso contrario, Apache deberá ser recompilado para agregar o eliminar módulos.

Las directivas de configuración se pueden incluir de forma condicional dependiendo de la presencia de un módulo particular, poniéndolas dentro de un bloque <IfModule> como veremos más adelante

Instalación y carga de módulos

Comprobar los módulos que se han cargado estáticamente

Iniciar sesión como Administrador y comprobar los módulos estáticos que se han cargado

Comprobar los módulos que se han cargado dinámicamente

Los módulos dinámicos cargados se comprueban consultando el directorio /etc/apache2/mods-enabled, que son enlaces simbólicos a ficheros de /etc/apache2/mods-available.

Carpetas de módulos

mods-available: Este directorio contiene una serie de archivos .load y .conf.

- Los archivos. load contienen directivas de configuración de Apache necesarias para la carga del módulo en cuestión.
- El correspondiente archivo .conf contiene directivas de configuración necesarias para la utilización del módulo en cuestión.

mods-enabled: para activar un módulo para Apache2 es necesario crear un enlace simbólico en este directorio a los

Módulos activados por defecto

Por defecto la instalación de Apache2 deja 'activados' un grupo de módulos.

```
usuario@ubserver02:~$ ls -l /etc/apache2/mods-enabled/
total 0
lrwxrwxrwx 1 root root 36 nov 28 10:08 access_compat.load -> ../mods-available/access_compat.load
lrwxrwxrwx 1 root root 28 nov 28 10:08 alias.conf -> ../mods-available/alias.conf
lrwxrwxrwx 1 root root 28 nov 28 10:08 alias.load -> ../mods-available/alias.load
lrwxrwxrwx 1 root root 33 nov 28 10:08 auth_basic.load -> ../mods-available/auth_basic.load
lrwxrwxrwx 1 root root 33 nov 28 10:08 authn_core.load -> ../mods-available/authn_core.load
lrwxrwxrwx 1 root root 33 nov 28 10:08 authn_file.load -> ../mods-available/authn_file.load
lrwxrwxrwx 1 root root 33 nov 28 10:08 authz_core.load -> ../mods-available/authz_core.load
lrwxrwxrwx 1 root root 33 nov 28 10:08 authz_host.load -> ../mods-available/authz_host.load
lrwxrwxrwx 1 root root 33 nov 28 10:08 authz_user.load -> ../mods-available/authz_user.load
lrwxrwxrwx 1 root root 32 nov 28 10:08 autoindex.conf -> ../mods-available/autoindex.conf
lrwxrwxrwx 1 root root 32 nov 28 10:08 autoindex.load -> ../mods-available/autoindex.load
lrwxrwxrwx 1 root root 30 nov 28 10:08 deflate.conf -> ../mods-available/deflate.conf
lrwxrwxrwx 1 root root 30 nov 28 10:08 deflate.load -> ../mods-available/deflate.load
lrwxrwxrwx 1 root root 26 nov 28 10:08 dir.conf -> ../mods-available/dir.conf
lrwxrwxrwx 1 root root 26 nov 28 10:08 dir.load -> ../mods-available/dir.load
lrwxrwxrwx 1 root root 26 nov 28 10:08 env.load -> ../mods-available/env.load
lrwxrwxrwx 1 root root 29 nov 28 10:08 filter.load -> ../mods-available/filter.load
lrwxrwxrwx 1 root root 27 nov 28 10:08 mime.conf -> ../mods-available/mime.conf
lrwxrwxrwx 1 root root 27 nov 28 10:08 mime.load -> ../mods-available/mime.load
lrwxrwxrwx 1 root root 32 nov 28 10:08 mpm_event.conf -> ../mods-available/mpm_event.conf
lrwxrwxrwx 1 root root 32 nov 28 10:08 mpm_event.load -> ../mods-available/mpm_event.load
lrwxrwxrwx 1 root root 34 nov 28 10:08 negotiation.conf -> ../mods-available/negotiation.conf
lrwxrwxrwx 1 root root 34 nov 28 10:08 negotiation.load -> ../mods-available/negotiation.load
lrwxrwxrwx 1 root root 33 nov 28 10:08 reqtimeout.conf -> ../mods-available/reqtimeout.conf
lrwxrwxrwx 1 root root 33 nov 28 10:08 reqtimeout.load -> ../mods-available/reqtimeout.load
lrwxrwxrwx 1 root root 31 nov 28 10:08 setenvif.conf -> ../mods-available/setenvif.conf
lrwxrwxrwx 1 root root 31 nov 28 10:08 setenvif.load -> ../mods-available/setenvif.load
lrwxrwxrwx 1 root root 29 nov 28 10:08 status.conf -> ../mods-available/status.conf
lrwxrwxrwx 1 root root 29 nov 28 10:08 status.load -> ../mods-available/status.load
usuario@ubserver02:~$ _
```

Directiva LoadModule

Editar un archivo. load. Observar cómo se utiliza la directiva LoadModule

```
usuario@ubserver02:~$ cat /etc/apache2/mods-enabled/alias.load
LoadModule alias_module /usr/lib/apache2/modules/mod_alias.so
usuario@ubserver02:~$ _
```

Archivos de configuración de módulos

Editar un fichero .conf y observar cómo se añaden directivas dentro de una declaración <IfModule nombre modulo>...</IfModule> que se ejecutarán si se carga el módulo

```
<IfModule alias_module>
        # Aliases: Add here as many aliases as you need (with no limit). The format is
        # Alias fakename realname
        #
        # Note that if you include a trailing / on fakename then the server will
        # require it to be present in the URL.  So "/icons" isn't aliased in this
        # example, only "/icons/".  If the fakename is slash-terminated, then the
        # realname must also be slash terminated, and if the fakename omits the
        # trailing slash, the realname must also omit it.
        #
        # We include the /icons/ alias for FancyIndexed directory listings.  If
        # you do not use FancyIndexing, you may comment this out.

        Alias /icons/ "/usr/share/apache2/icons/"

        <Directory "/usr/share/apache2/icons">
                Options FollowSymlinks
                AllowOverride None
                Require all granted
        </Directory>

</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
~
```

Este es el módulo gracias al cual hemos podido crear alias en ejercicios anteriores.

Consultar los módulos disponibles

Consultar el directorio /usr/lib/apache2/modules para ver los módulos disponibles para cargar. Estos módulos están instalados, pero no tienen por qué estar habilitados.

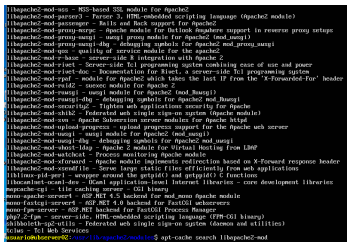
```
usuario@ubserver02:/usr/lib/apache2/modules$ ls
httpd.exp               mod_cache_socache.so   mod_info.so               mod_proxy_wstunnel.so
mod_access_compat.so    mod_case_filter_in.so  mod_lbmethod_bybusyness.so  mod_ratelimit.so
mod_actions.so          mod_case_filter.so     mod_lbmethod_byrequests.so  mod_reflector.so
mod_alias.so            mod_cern_meta.so       mod_lbmethod_bytraffic.so   mod_remoteip.so
mod_allowmethods.so     mod_cgid.so            mod_lbmethod_heartbeat.so   mod_reqtimeout.so
mod_asis.so             mod_cgi.so             mod_ldap.so               mod_request.so
mod_auth_basic.so       mod_charset_lite.so    mod_log_debug.so          mod_rewrite.so
mod_auth_digest.so      mod_data.so            mod_log_forensic.so       mod_sed.so
mod_auth_form.so        mod_dav_fs.so          mod_lua.so                mod_session_cookie.so
mod_authn_anon.so       mod_dav_lock.so        mod_macro.so              mod_session_crypto.so
mod_authn_core.so       mod_dav.so             mod_mime_magic.so         mod_session_dbd.so
mod_authn_dbd.so        mod_dbd.so             mod_mime.so               mod_session.so
mod_authn_dbm.so        mod_deflate.so         mod_mpm_event.so          mod_setenvif.so
```

No son los únicos mods disponibles, puesto que se pueden descargar e instalar muchos más a través del gestor de paquetes (apt).

Listar mods para Apache

Comando para mostrar los paquetes disponibles en los repositorios para cargar módulos adicionales en Apache.

Apt-cache search libapache2-mod



Mod USEDIR

Directorios personales de usuarios (módulo userdir)

El objetivo de esta actividad es configurar en nuestro Web Server para que pueda ofrecer un espacio web para usuarios de nuestro sistema y que estos puedan tener un espacio web.

Cada uno de estos usuarios disfrutará de un espacio web que se almacenará en la carpeta /public_html, dentro de su carpeta home (/home/nombreusuario/public_html)

Para que Apache procese los espacios web de los usuarios es necesario activar el módulo **userdir**

Comprobar que el módulo no está habilitado

Lo podemos hacer consultando el directorio /etc/apache2/mods-enabled que el módulo userdir no está habilitado.

En esta carpeta nunca tocaremos nada, puesto que todo lo que hay en ella lo crea automáticamente Apache.

Habilitar el módulo ejecutando el comando: a2enmod userdir

Para habilitar o deshabilitar cualquier módulo de los que dispone Apache debemos utilizar los comandos:

- **a2enmod** para habilitar un módulo que nos interese
- **a2dismod** para deshabilitar el módulo.

Los módulos disponibles se encuentran en el directorio /etc/apache2/mods-available

```
usuario@ubserver02:/etc/apache2/mods-enabled$ sudo a2enmod usedir
[sudo] contraseña para usuario:
ERROR: Module usedir does not exist!
```

Vamos a buscarlo

```
usuario@ubserver02:/etc/apache2/mods-enabled$ apt-cache search libapache2-mod | grep userdir
libapache2-mod-ldap-userdir - Apache module that provides UserDir lookups via LDAP
libapache2-mod-ldap-userdir-dbg - Debugging symbols for mod_ldap_userdir
usuario@ubserver02:/etc/apache2/mods-enabled$
```

Y lo instalamos: sudo apt install libapache2-mod-ldap-usedir

```
usuario@ubserver02:/$ sudo apt install libapache2-mod-ldap-userdir
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-ldap-userdir
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 44 no actualizados.
Se necesita descargar 19,0 kB de archivos.
Se utilizarán 87,0 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu bionic/universe amd64 libapache2-mod-ldap-userdir amd64 1.
1.19-2.1 [19,0 kB]
Descargados 19,0 kB en 0s (102 kB/s)
Seleccionando el paquete libapache2-mod-ldap-userdir previamente no seleccionado.
(Leyendo la base de datos ... 105886 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libapache2-mod-ldap-userdir_1.1.19-2.1_amd64.deb ...
Desempaquetando libapache2-mod-ldap-userdir (1.1.19-2.1) ...
Configurando libapache2-mod-ldap-userdir (1.1.19-2.1) ...
apache2_invoke: Enable module ldap_userdir
usuario@ubserver02:/$ _
```

Vemos que me los ha creado:

```
usuario@ubserver02:/$ ll /etc/apache2/mods-available/ | grep userdir
-rw-r--r-- 1 root root    76 jul 13  2013 ldap_userdir.load
-rw-r--r-- 1 root root   324 oct 10 20:59 userdir.conf
-rw-r--r-- 1 root root    66 oct 10 20:59 userdir.load
```

Lo cargamos

```
usuario@ubserver02:/$ sudo a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
  systemctl restart apache2
usuario@ubserver02:/$ _
```

Verificar que se ha cargado el módulo

Verificar dentro del directorio /etc/apache2/mods-enabled que se han creado enlaces simbólicos del módulo userdir (ficheros .conf y. load) hacia /etc/apache2/mods-available

```
usuario@ubserver02:/$ ll /etc/apache2/mods-enabled/u*
lrwxrwxrwx 1 root root 30 dic  2 16:48 /etc/apache2/mods-enabled/userdir.conf -> ../mods-available/u
serdir.conf
lrwxrwxrwx 1 root root 30 dic  2 16:48 /etc/apache2/mods-enabled/userdir.load -> ../mods-available/u
serdir.load
usuario@ubserver02:/$
```

Reiniciamos el servidor: sudo systemctl restart apache2

Consultar fichero de configuración de userdir

Consultar el fichero /etc/apache2/mod_enabled/userdir.conf

```
<IfModule mod_userdir.c>
        UserDir public_html
        UserDir disabled root

        <Directory /home/*/public_html>
                AllowOverride FileInfo AuthConfig Limit Indexes
                Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
                Require method GET POST OPTIONS
        </Directory>
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Crear directorio personal

Iniciar sesión como usuario y crear el directorio /home/pepe/public_html

Por defecto la carpeta public_html no está creada en ninguna carpeta home de ningún usuario. Por lo que debemos crear y asignar los permisos necesarios.

Asignar permisos a la carpeta

Es necesario asignar permisos 755 para que el grupo de usuarios y el resto de usuarios tengan acceso de lectura y puedan visualizar las páginas allí almacenadas.

Para cambiar permisos: chmod -R 755 /home/pepe/public_html

Dentro crear el archivo personal1.html con el texto "PÁGINA PERSONAL DEL USUARIO ALUMNO"

Acceder a http://IPdelServidor/~pepe/personal1.html

Modulo modsecurity

Opcional por si queréis investigar qué es y cómo funciona. Si vais mal de tiempo, ignorarlo.

Instalar el mod y configurar:

- Sudo apt-get install libapache2-mod-security2
- Sudo apachectl -M | grep –color security2

- sudo mv /etc/modsecurity/modsecurity.conf-recommended modsecurity.conf
- sudo service apache2 reload

Control de acceso

Control de acceso por IP y nombre de dominio

1. Iniciar sesión como administrador
2. Crear el directorio /var/www/xxx.smx2.org/privado
3. Crear dentro la página privado1.html con el texto "PÁGINA PRIVADA"

Editar /etc/apache2/sites-available/xxx.smx2.org.conf y utilizar la sentencia <Directory> para denegar el acceso al directorio a todos los equipos excepto al local y a la MV Ubuntu Desktop

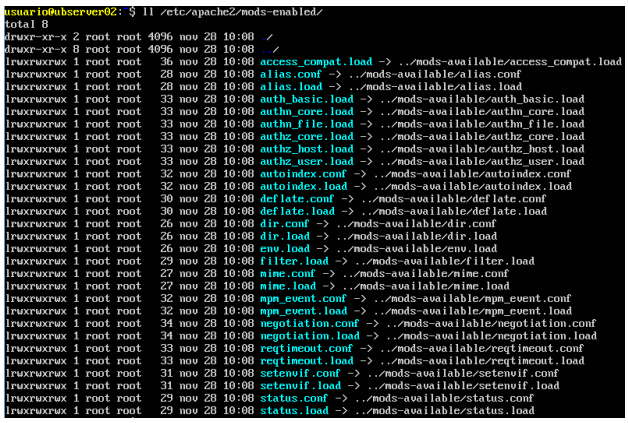
Reiniciamos el servidor: sudo systemctl restart apache2

Comprobar el acceso a http://IPdelServdor/privado desde varias ubicaciones

Autenticación HTTP Basic

Comprobar módulo auth_basic habilitado

Miramos en /etc/apache2/mods-enabled, que el módulo **auth_basic** está habilitado



Usar autenticación básica en Apache

Crear el fichero /etc/apache2/passwd y añadir el usuario mortadelo.

Ejecutando el comando htpasswd -c mortadelo. La opción -c creará el fichero.

```
usuario@ubserver02:/etc/apache2$ sudo htpasswd -c /etc/apache2/passwd mortadelo
[sudo] contraseña para usuario:
New password:
Re-type new password:
Adding password for user mortadelo
usuario@ubserver02:/etc/apache2$ _
```

Fichero passwd creado en /etc/apache2

```
usuario@ubserver02:/etc/apache2$ ll
total 92
drwxr-xr-x  8 root root  4096 dic  2 15:37 ./
drwxr-xr-x 89 root root  4096 dic  2 13:04 ../
-rw-r--r--  1 root root  7224 oct 10 20:59 apache2.conf
drwxr-xr-x  2 root root  4096 nov 28 10:10 conf-available/
drwxr-xr-x  2 root root  4096 nov 28 10:10 conf-enabled/
-rw-r--r--  1 root root  1782 oct 10 20:59 envvars
-rw-r--r--  1 root root 31063 oct 10 20:59 magic
drwxr-xr-x  2 root root 12288 nov 28 10:08 mods-available/
drwxr-xr-x  2 root root  4096 nov 28 10:08 mods-enabled/
-rw-r--r--  1 root root    48 dic  2 15:37 passwd
-rw-r--r--  1 root root   320 oct 10 20:59 ports.conf
drwxr-xr-x  2 root root  4096 dic  2 15:02 sites-available/
drwxr-xr-x  2 root root  4096 nov 28 10:08 sites-enabled/
usuario@ubserver02:/etc/apache2$ cat passwd
mortadelo:$apr1$rDUzD8oC$5WO.jjgALBw2WS.83zst40
usuario@ubserver02:/etc/apache2$ _
```

Añadir el usuario filemon

Ejecutando el comando htpasswd. La opción c ya no es necesaria para añadir nuevos usuarios al fichero passwd

```
usuario@ubserver02:/etc/apache2$ sudo htpasswd /etc/apache2/passwd filemon
New password:
Re-type new password:
Adding password for user filemon
usuario@ubserver02:/etc/apache2$ cat passwd
mortadelo:$apr1$rDUzD8oC$5WO.jjgALBw2WS.83zst40
filemon:$apr1$/nsyZc4T$oCG97wOXIENvjnut8iCh40
usuario@ubserver02:/etc/apache2$
```

Permitir accesos a mortadelo y filemon

- Editar /etc/apache2/sites-available/xxx.smx2.org.conf
- Permitir el acceso al directorio /var/www/privado a los usuarios mortadelo y filemon

```
<Directory /var/www/html/privado>
        Options Indexes FollowSymlinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from 127.0.0.1 localhost
        allow from 192.168.1.2
        AuthName "Acceso Privado"
        Authtype Basic
        AuthUserFile /etc/apache2/passwd
        Require user mortadelo filemon
</Directory>
```

Reiniciamos el servidor: sudo systemctl restart apache2

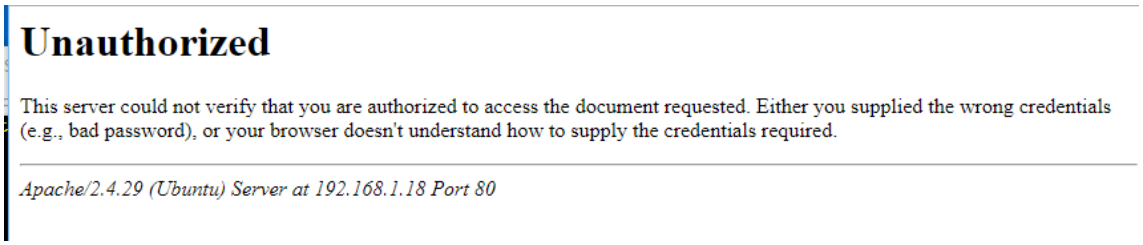
Comprobación de los cambios realizados

Accediendo a la página web, veremos que no nos dejará acceder a ella, y nos pedirá un usuario y contraseña válidos para dejarnos continuar.

Acceso con usuario no autorizado

Conectando a **http://IPdelServidor/privado/**

En caso de que los datos proporcionados no sean correctos, nos mostrará una captura indicando que el acceso no ha sido autorizado.



Habilitar uso de .htaccess

Los ficheros .htaccess (o "ficheros de configuración distribuida") facilitan una forma de realizar cambios en la configuración en contexto directorio.

Se trata de un archivo, que contiene una o más directivas, se coloca en un directorio, y estas directivas aplican a ese directorio y todos sus subdirectorios.

Generalmente, los ficheros .htaccess usan la misma sintaxis que los ficheros de la configuración principal. Lo que puede utilizar en estos ficheros lo determina la directiva AllowOverride. Esta directiva específica, en categorías, qué directivas tendrán efecto si se encuentran en un fichero .htaccess.

Generalmente, solo se debería usar ficheros .htaccess cuando no tiene acceso al fichero principal de configuración del servidor.

- Como administrador, editar el fichero /etc/apache2/sites-available/xxx.smx2.org.conf
- Habilitar el uso de ficheros .htaccess

AllowOverride va a permitir que se puedan sobrescribir parámetros desde un archivo externo

```
Alias /wiki /home/alumno/wiki
<Directory /home/alumno/wiki>
        AllowOverride All
</Directory>
```

Reiniciamos el servidor: sudo systemctl restart apache2

Crear un archivo .htaccess

- Iniciar sesión como usuario alumno
- Crear el fichero /home/alumno/wiki/ .htaccess

Configuración mediante .htaccess

Pasos:

- Crear el fichero /home/alumno/wiki/.htpassw y añadir al usuario wiki
- Editar el fichero /home/alumno/wiki/.htaccess y añadir las directivas para realizar autenticación básica

Acceder a http://IPdelServdor/wiki

Configurar un site seguro con https

Configurar servidor HTTPS

mod_ssl es un módulo opcional para el servidor HTTP Apache. Proporciona criptografía para el servidor a través de los protocolos criptográficos Secure Sockets Layer) y Transport Layer Security.

Habilitar mod_ssl

```
pepe@ubserver02:~$ sudo a2enmod ssl
[sudo] contraseña para pepe:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certifica
tes.
To activate the new configuration, you need to run:
  systemctl restart apache2
pepe@ubserver02:~$ sudo systemctl restart apache2
pepe@ubserver02:~$
```

Reiniciamos el servidor: sudo systemctl restart apache2

Comprobar carga del módulo

```
lrwxrwxrwx 1 root root   31 nov 28 10:08 setenvif.conf -> ../mods-available/setenvif.conf
lrwxrwxrwx 1 root root   31 nov 28 10:08 setenvif.load -> ../mods-available/setenvif.load
lrwxrwxrwx 1 root root   36 dic  4 17:33 socache_shmcb.load -> ../mods-available/socache_shmcb.load
lrwxrwxrwx 1 root root   26 dic  4 17:33 ssl.conf -> ../mods-available/ssl.conf
lrwxrwxrwx 1 root root   26 dic  4 17:33 ssl.load -> ../mods-available/ssl.load
lrwxrwxrwx 1 root root   29 nov 28 10:08 status.conf -> ../mods-available/status.conf
lrwxrwxrwx 1 root root   29 nov 28 10:08 status.load -> ../mods-available/status.load
lrwxrwxrwx 1 root root   30 dic  2 16:48 userdir.conf -> ../mods-available/userdir.conf
lrwxrwxrwx 1 root root   30 dic  2 16:48 userdir.load -> ../mods-available/userdir.load
pepe@ubserver02:~$ ll /etc/apache2/mods-enabled/
```

Ver contenido de port.conf

En este caso, al estar habilitado el módulo SSL_module, el servidor abrirá, además del puerto 80, el puerto 443.

- Las peticiones HTTP llegarán por defecto por el puerto 80
- Las peticiones HTTPS llegarán por defecto por el puerto 443

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Comprobar puertos con netstat

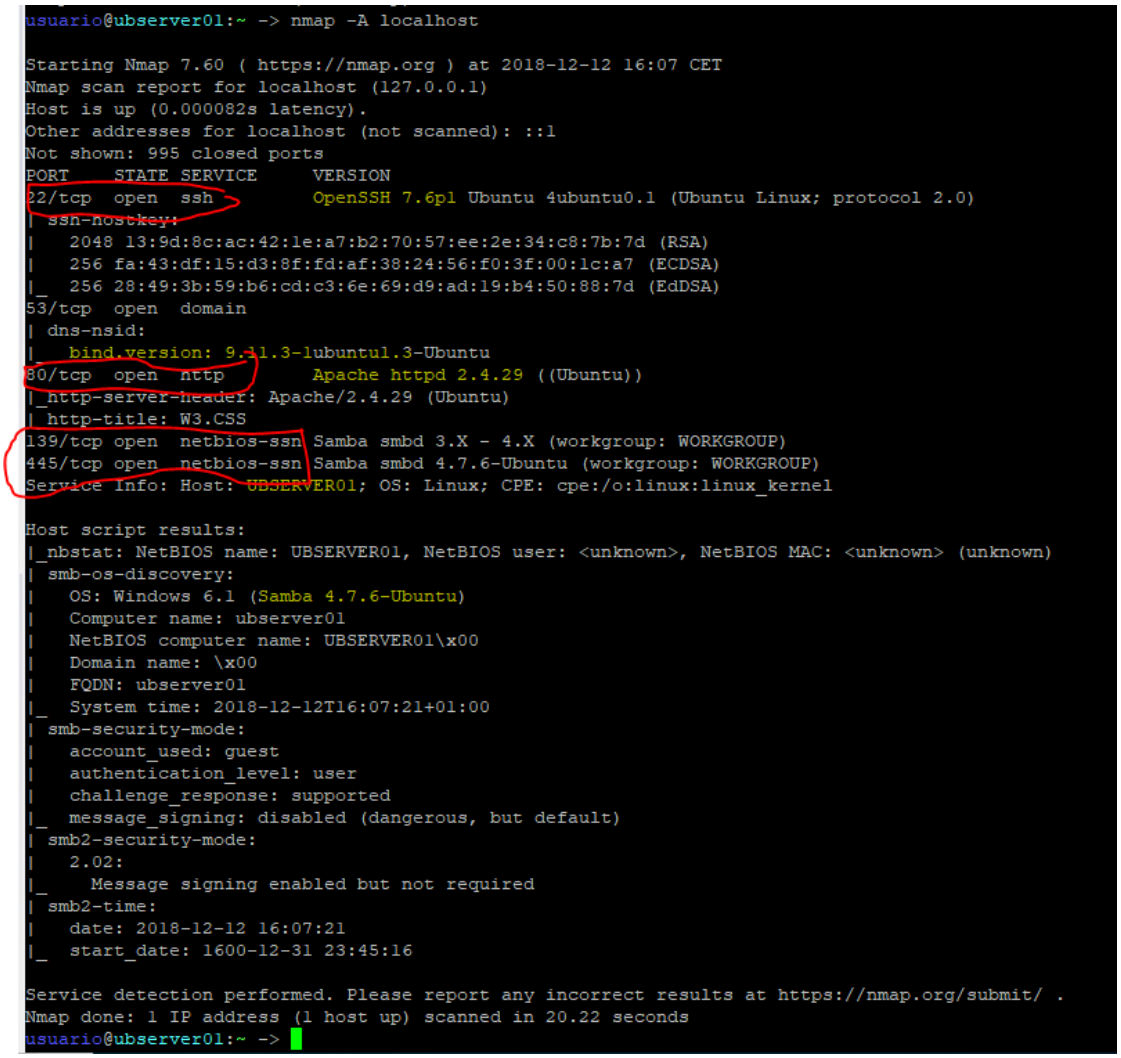
```
pepe@ubserver02:~$ netstat -ntl
Conexiones activas de Internet (solo servidores)
Proto  Recib Enviad Dirección local         Dirección remota       Estado
tcp        0      0 127.0.0.53:53           0.0.0.0:*              ESCUCHAR
tcp        0      0 0.0.0.0:22              0.0.0.0:*              ESCUCHAR
tcp6       0      0 :::22                   :::*                   ESCUCHAR
tcp6       0      0 :::443                  :::*                   ESCUCHAR
tcp6       0      0 :::80                   :::*                   ESCUCHAR
```

Comprobar puertos con nmap

Nmap es un programa de código abierto que sirve para efectuar rastreo de puertos. Instalar si no lo está: sudo apt install nmap

```
pepe@ubserver02:~$ nmap localhost

Starting Nmap 7.60 ( https://nmap.org ) at 2018-12-04 17:37 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00012s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
pepe@ubserver02:~$ _
```

Con la opción -A podéis averiguar mucha más información:



Para más información y comandos de nmap: https://hackertarget.com/nmap-cheatsheet-a-quick-reference-guide/

Habilitar servidor virtual por defecto para SSL

Comprobamos que existe un site por defecto para SSL, pero no está habilitado. Se llama default-ssl

```
pepe@ubserver02:~$ ll /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov 28 10:08 ./
drwxr-xr-x 8 root root 4096 dic  2 15:45 ../
lrwxrwxrwx 1 root root   35 nov 28 10:08 000-default.conf -> ../sites-available/000-default.conf
pepe@ubserver02:~$ ll /etc/apache2/sites-available/
total 20
drwxr-xr-x 2 root root 4096 dic  2 15:52 ./
drwxr-xr-x 8 root root 4096 dic  2 15:45 ../
-rw-r--r-- 1 root root 1667 dic  2 15:49 000-default.conf
-rw-r--r-- 1 root root 6338 oct 10 20:59 default-ssl.conf
pepe@ubserver02:~$
```

Habilitamos y comprobamos que se carga el site.

```
pepe@ubserver02:~$ a2ensite default-ssl
Enabling site default-ssl.
Could not create /etc/apache2/sites-enabled/default-ssl.conf: Permission denied
pepe@ubserver02:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
pepe@ubserver02:~$ sudo systemctl reload apache2
pepe@ubserver02:~$ ll /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 dic  4 17:39 ./
drwxr-xr-x 8 root root 4096 dic  2 15:45 ../
lrwxrwxrwx 1 root root   35 nov 28 10:08 000-default.conf -> ../sites-available/000-default.conf
lrwxrwxrwx 1 root root   35 dic  4 17:39 default-ssl.conf -> ../sites-available/default-ssl.conf
pepe@ubserver02:~$
```

Comprobar si puedo cargar la página web por HTTPS

```
pepe@ubserver02:~$ wget https://localhost
--2018-12-04 17:40:40--  https://localhost/
Resolviendo localhost (localhost)... ::1, 127.0.0.1
Conectando con localhost (localhost)[::1]:443... conectado.
ERROR: no se puede verificar el certificado de localhost, emitido por “CN=ubserver02”:
  Se encontró un certificado autofirmado.
ERROR: ningún nombre de sujeto alternativo del certificado encaja con
       el nombre de equipo “localhost” solicitado.
Para conectar inseguramente a localhost, use `--no-check-certificate'.
pepe@ubserver02:~$
```

Comprobaremos que no podemos descargar la página

El servidor apache utiliza por defecto un **certificado digital auto firmado** creado al instalar apache. Un certificado auto firmado no está firmado por una autoridad de certificación (3a parte de confianza) y por tanto no existen mecanismos automáticos que garanticen su autenticidad. Por eso los navegadores pedirán confirmación cuando el servidor se lo envíe.

Conectar al site SSL sin comprobar certificado

```
pepe@ubserver02:~$ wget https://localhost --no-check-certificate
--2018-12-04 17:41:33--  https://localhost/
Resolviendo localhost (localhost)... ::1, 127.0.0.1
Conectando con localhost (localhost)[::1]:443... conectado.
AVISO: no se puede verificar el certificado de localhost, emitido por "CN=ubserver02":
  Se encontró un certificado autofirmado.
AVISO: ningún nombre de sujeto alternativo del certificado encaja con
       el nombre de equipo "localhost" solicitado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 948 [text/html]
Guardando como: "index.html.2"

index.html.2            100%[=================================>]     948  --.-KB/s    en 0s

2018-12-04 17:41:33 (107 MB/s) - "index.html.2" guardado [948/948]

pepe@ubserver02:~$
```

Crear un servidor virtual https

Creo la carpeta "seguro" y dentro la página index.html que lo identifique.

- sudo mkdir /var/www/seguro
- sudo nano /var/www/seguro/index.html

Crear un certificado digital

Creo un certificado digital auto firmado usado openssl. Para ello:

- Me voy al directorio home del usuario
- Creo una clave privada RSA (en este caso de 2048 bits)

```
root@ubserver02:/# openssl genrsa -out seguro.key 2048
Generating RSA private key, 2048 bit long modulus
........+++
.....+++
e is 65537 (0x010001)
root@ubserver02:/# _
```

Una vez generada la clave privada, puedo generar una solicitud de certificado csr (relleno los datos)

```
root@ubserver02:/# openssl req -new -key seguro.key -out seguro.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Illes Balears
Locality Name (eg, city) []:Palma
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES DMORENO
Organizational Unit Name (eg, section) []:DEPARTAMENT INFORMATICA
Common Name (e.g. server FQDN or YOUR name) []:Daniel
Email Address []:dmoreno@iesdmoreno.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:pa$$w0rd
An optional company name []:IES DMORENO
root@ubserver02:/#
```

Esta solicitud de certificado creada, la podría enviar a una autoridad de certificación para generase el certificado crt. En este caso lo firmamos nosotros, es decir creamos un certificado auto firmado usando la clave privada.

```
root@ubserver02:/# openssl x509 -req -days 365 -in seguro.csr -signkey seguro.key -out seguro.crt
Signature ok
subject=C = ES, ST = Illes Balears, L = Palma, O = IES DMORENO, OU = DEPARTAMENT INFORMATICA, CN = D
aniel, emailAddress = dmoreno@iesdmoreno.org
Getting Private key
root@ubserver02:/# _
```

Lo siguiente que haremos es mover la clave y el certificado creados a los directorios que utiliza por defecto apache y configurar los permisos adecuados (se habrán creado en el directorio en el que estuviéramos al ejecutar los comandos anteriores).

```
root@ubserver02:/# mv seguro.key /etc/ssl/private/
root@ubserver02:/# mv seguro.crt /etc/ssl/certs/
root@ubserver02:/# chown root:ssl-cert /etc/ssl/private/seguro.key
root@ubserver02:/# chmod 640 /etc/ssl/private/seguro.key
root@ubserver02:/# chown root:root /etc/ssl/certs/seguro.crt
root@ubserver02:/#
```

Cat seguro.key seguro.crt > nombre-sitio.pem

Copy nombre-sitio.pem /etc/ssl/private/

Crear un nuevo site seguro

Creo el archivo de configuración seguro (hago una copia del default-ssl con el nombre seguro)

```
root@ubserver02:/#
root@ubserver02:/# cd /etc/apache2/sites-available/
root@ubserver02:/etc/apache2/sites-available# cp default-ssl.conf seguro.conf
root@ubserver02:/etc/apache2/sites-available# nano seguro.conf
```

Configuración del nuevo site

Modificar la configuración del site **seguro.** Dado que lleva la sección ifmodule delante, este site no se cargará si el módulo SSL no está cargado, pese a que si aparezca en sites-enabled.

```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost

                ServerName seguro.iesdmoreno.org
                DocumentRoot /var/www/html/seguro

                <Directory /var/www/seguro>
                        DirectoryIndex index.html
                        Options Indexes FollowSymLinks MultiViews
                        AllowOverride None
                        Order allow,deny
                        allow from all
                </Directory>

                # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
                # error, crit, alert, emerg.
                # It is also possible to configure the loglevel for particular
                # modules, e.g.
                #LogLevel info ssl:warn

                ErrorLog ${APACHE_LOG_DIR}/seguro.error.log
                CustomLog ${APACHE_LOG_DIR}/seguro_access.log combined

                # For most configuration files from conf-available/, which are
                # enabled or disabled at a global level, it is possible to
                # include a line for only one particular virtual host. For example the
                # following line enables the CGI configuration for this host only
                # after it has been globally disabled with "a2disconf".
                #Include conf-available/serve-cgi-bin.conf
```

Agregar las directivas dentro de virtualhost:

SSLEngine on

SSLCertificateFile /etc/ssl/private/nombre-sitio.pem

Hay que comentar la opción SSLCertificateKeyFile

Dentro de la carpeta html debe aparecer la siguiente configuración:

<Directory /var/www/html>

SSLRequireSSL

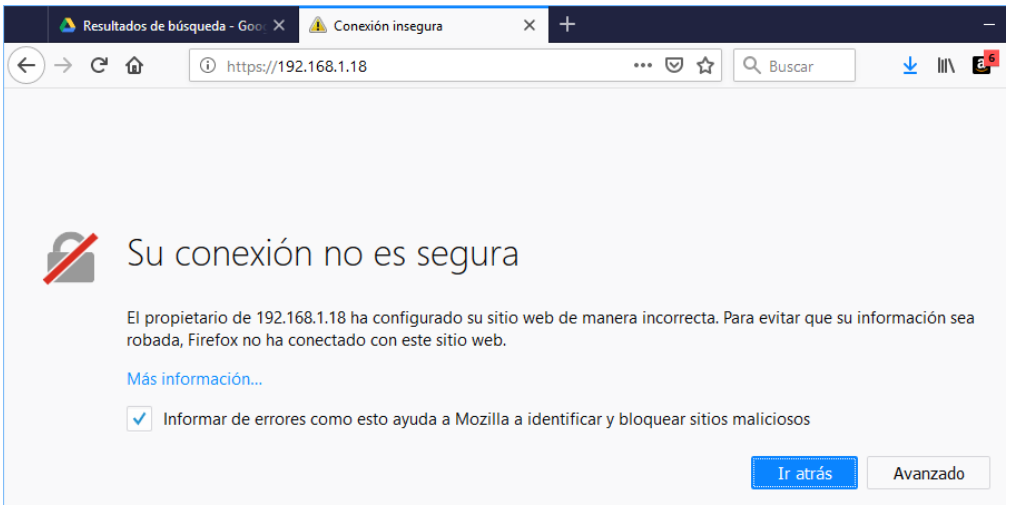
...

</Directory>

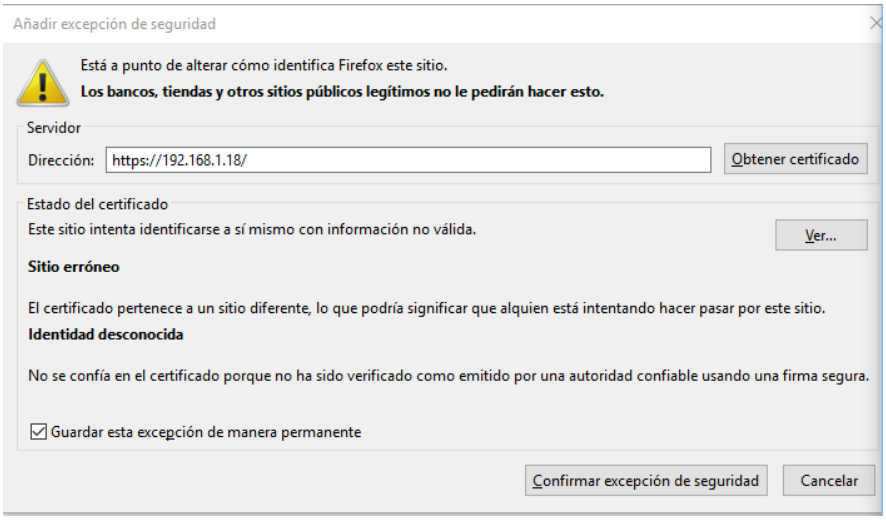
Por defecto tenemos activado el site HTTPS que viene por defecto con apache, por lo que deberemos deshabilitarlo y habilitar el site nuevo:

- Deshabilito el servidor virtual SSL que venía por defecto (default-ssl): a2dissite default-ssl
- Habilitamos el site que hemos creado (seguro): a2ensite seguro

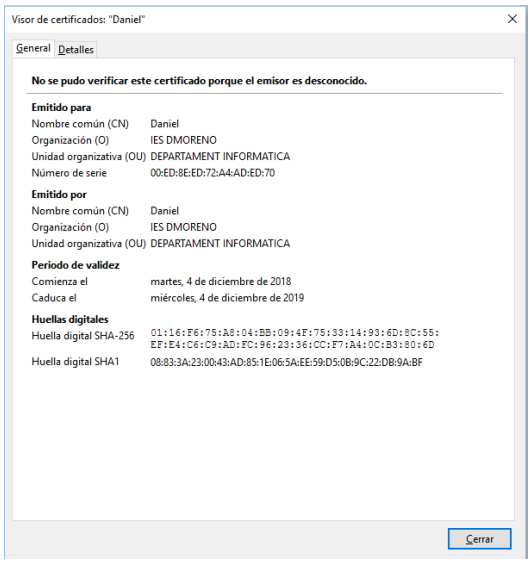
Conexión al servidor HTTPS desde el navegador



Como el certificado es autofirmado nos saltará un aviso, y podremos comprobar el certificado antes de darle validez



Consultar certificado en el cliente



Una vez aceptado el certificado, podemos acceder a la página