

# Módulo de Servidores - Memoria Técnica

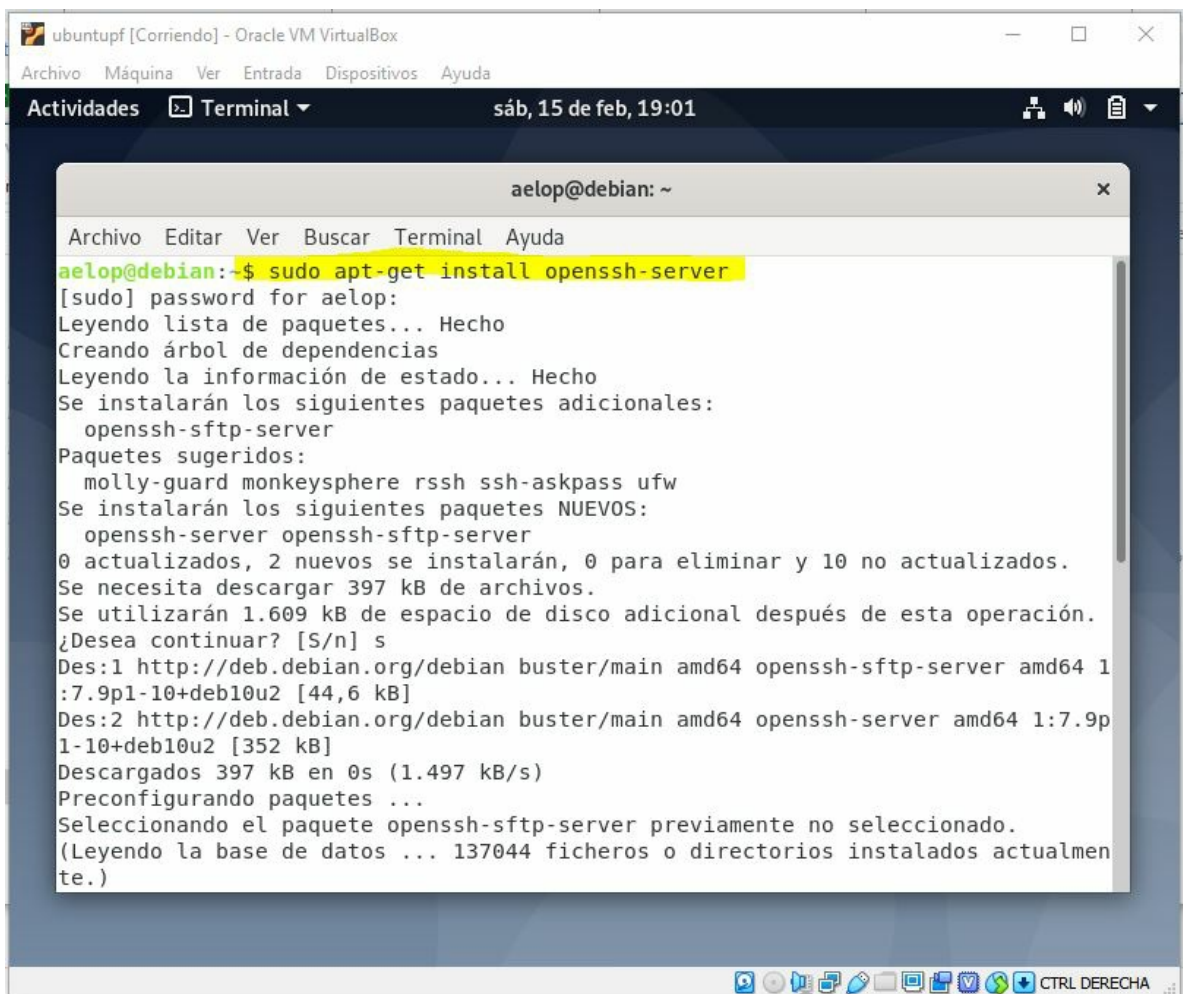
## Consideraciones previas

Para la realización de esta práctica he utilizado un sistema operativo Debian GNU/Linux versión 10.

## Instalación del servidor SSH y APACHE 2

1. Para permitir conexión remota al terminal habilitamos un servidor SSH. Lo instalaremos desde los repositorios de paquetes oficiales de Debian con el comando:

```
sudo apt-get install openssh-server
```



The screenshot shows a terminal window titled 'aelop@debian: ~' within an Oracle VM VirtualBox environment. The terminal output shows the execution of the command 'sudo apt-get install openssh-server'. The user 'aelop' provides the password. The system then lists the packages to be installed, including 'openssh-sftp-server' and 'openssh-server'. It shows the disk space requirements and the download progress of the packages. The installation is successful, and the terminal displays the final status of the packages.

```
aelop@debian:~$ sudo apt-get install openssh-server
[sudo] password for aelop:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  openssh-sftp-server
Paquetes sugeridos:
  molly-guard monkeysphere rssh ssh-askpass ufw
Se instalarán los siguientes paquetes NUEVOS:
  openssh-server openssh-sftp-server
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 10 no actualizados.
Se necesita descargar 397 kB de archivos.
Se utilizarán 1.609 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://deb.debian.org/debian buster/main amd64 openssh-sftp-server amd64 1:7.9p1-10+deb10u2 [44,6 kB]
Des:2 http://deb.debian.org/debian buster/main amd64 openssh-server amd64 1:7.9p1-10+deb10u2 [352 kB]
Descargados 397 kB en 0s (1.497 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete openssh-sftp-server previamente no seleccionado.
(Leyendo la base de datos ... 137044 ficheros o directorios instalados actualmente.)
```

2. Una vez instalado el servidor de SSH pasaremos a establecer una conexión remota. En este caso con una máquina Windows y el cliente SSH PuTTY.

```
aelop@debian: ~  
login as: aelop  
aelop@192.168.0.114's password:  
Linux debian 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1 (2020-01-26) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
aelop@debian:~$ cat /etc/apt/sources.list  
#  
  
# deb cdrom:[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 NETINST 20200208-  
12:07]/ buster main  
  
#deb cdrom:[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 NETINST 20200208-1  
2:07]/ buster main  
  
deb http://deb.debian.org/debian/ buster main  
deb-src http://deb.debian.org/debian/ buster main  
  
deb http://security.debian.org/debian-security buster/updates main  
deb-src http://security.debian.org/debian-security buster/updates main  
  
# buster-updates, previously known as 'volatile'  
deb http://deb.debian.org/debian/ buster-updates main  
deb-src http://deb.debian.org/debian/ buster-updates main  
  
# This system was installed using small removable media  
# (e.g. netinst, live or single CD). The matching "deb cdrom"  
# entries were disabled at the end of the installation process.  
# For information about how to configure apt package sources,  
# see the sources.list(5) manual.  
aelop@debian:~$
```

Conectamos con el usuario aelop a la IP 192.168.0.114 de la red local. Aprovechamos para comprobar qué repositorios de paquetes están disponibles en el sistema para saber si podremos instalar Apache 2 en el siguiente paso. Para ello sacamos el contenido del fichero `/etc/apt/sources.list` por salida estándar:

```
cat /etc/apt/sources.list
```

3. A continuación instalamos el servidor web Apache 2:

```
sudo apt-get install apache2
```

```
aelop@debian: ~  
aelop@debian:~$ su  
Contraseña:  
root@debian:/home/aelop# apt-get install apache2  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  apache2-data apache2-utils  
Paquetes sugeridos:  
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom  
Se instalarán los siguientes paquetes NUEVOS:  
  apache2 apache2-data apache2-utils  
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 10 no actualizados.  
Se necesita descargar 653 kB de archivos.  
Se utilizarán 1.988 kB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] s  
Des:1 http://deb.debian.org/debian buster/main amd64 apache2-data all 2.4.38-3+deb10u3 [165 kB]  
Des:2 http://deb.debian.org/debian buster/main amd64 apache2-utils amd64 2.4.38-3+deb10u3 [236 kB]  
Des:3 http://deb.debian.org/debian buster/main amd64 apache2 amd64 2.4.38-3+deb10u3 [251 kB]  
Descargados 653 kB en 1s (700 kB/s)  
Seleccionando el paquete apache2-data previamente no seleccionado.  
(Leyendo la base de datos ... 137070 ficheros o directorios instalados actualmente.)  
Preparando para desempaquetar .../apache2-data_2.4.38-3+deb10u3_all.deb ...  
Desempaquetando apache2-data (2.4.38-3+deb10u3) ...  
Seleccionando el paquete apache2-utils previamente no seleccionado.  
Preparando para desempaquetar .../apache2-utils_2.4.38-3+deb10u3_amd64.deb ...  
Desempaquetando apache2-utils (2.4.38-3+deb10u3) ...  
Seleccionando el paquete apache2 previamente no seleccionado.  
Preparando para desempaquetar .../apache2_2.4.38-3+deb10u3_amd64.deb ...  
Desempaquetando apache2 (2.4.38-3+deb10u3) ...  
Configurando apache2-data (2.4.38-3+deb10u3) ...  
Configurando apache2-utils (2.4.38-3+deb10u3) ...  
Configurando apache2 (2.4.38-3+deb10u3) ...
```

Para verificar que el software se ha instalado correctamente levantaremos el servicio con el comando:

```
sudo systemctl apache2 start
```

Una vez arrancado debería de ser accesible la página de bienvenida de Apache 2 donde se muestra un texto que pone IT WORKS!. Podemos verla entrando desde cualquier navegador al puerto 80 de la máquina.

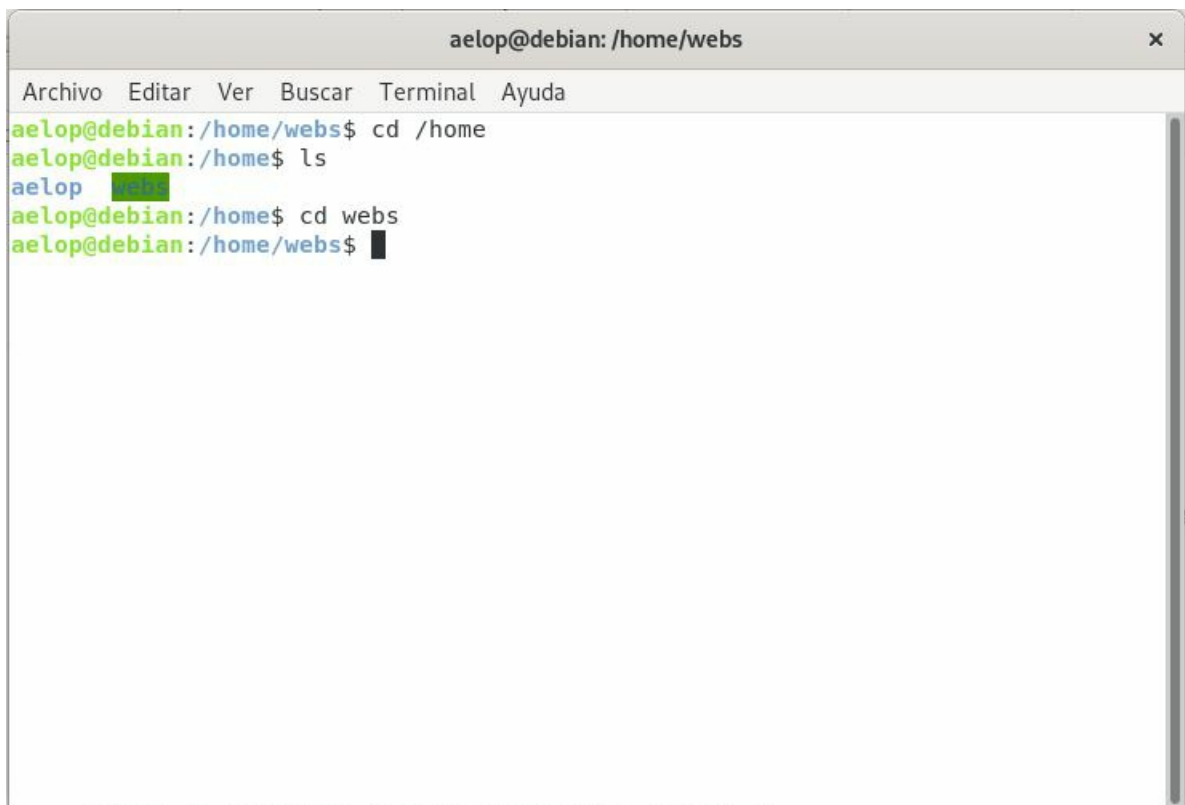
- `http://localhost`
- `http://127.0.0.1`
- `http://192.168.0.114` (Desde una máquina remota)

## Creación de los VirtualHosts

1. Creamos una carpeta donde se alojará el contenido de los distintos sitios virtuales que configuraremos. Para ello usamos permisos de administrador.

```
sudo mkdir /home/webs
```

En la captura puede verse el resultado:



```
aelop@debian: /home/webs
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
aelop@debian:/home/webs$ cd /home
aelop@debian:/home$ ls
aelop
aelop@debian:/home$ cd webs
aelop@debian:/home/webs$
```

VirtualHost: [www.miweb.com](http://www.miweb.com)

1. Como ya hemos dicho, los distintos sitios estarán alojados dentro de la carpeta `/home/webs`, por lo tanto crearemos un subdirectorio llamado `/miweb`.

```
sudo mkdir /home/webs/miweb
```

2. Dentro de esta carpeta creamos un fichero `index.html` que será la página de inicio del sitio.

```
echo '<html><body><h1>Hola mundo</h1><p>Página de inicio de miweb</body></html>' >> index.html
```

Puede verse en la siguiente captura:

```
aelop@debian:/home/webs$ ls
miweb
aelop@debian:/home/webs$ cd miweb/
aelop@debian:/home/webs/miweb$ ls
index.html
aelop@debian:/home/webs/miweb$ cat index.html
<html>
    <body>
        <h1>Hola mundo</h1>
        <p>P&aacute;gina de inicio de miweb</p>
    </body>
</html>
aelop@debian:/home/webs/miweb$
```

3. A continuación vamos a configurar un VirtualHost nuevo. Aquí hay que distinguir dos carpetas importantes a nivel de configuración de Apache:

- /etc/apache2/sites-available: Contiene los ficheros de configuración (.conf) de los distintos sitios virtuales disponibles para el servidor.
- /etc/apache2/sites-enabled: Aquí debe existir un enlace simbólico al fichero .conf de cada uno de los sitios virtuales que queramos tener habilitados.

Crearemos el fichero de configuración miweb.conf en la carpeta /sites-available y añadiremos las directivas necesarias:

```
sudo nano miweb.conf
```



```
aelop@debian: /etc/apache2/sites-available
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 3.2      miweb.conf

<VirtualHost www.miweb.com:80>
    #ServerName www.miweb.com
    ServerAlias www.miweb.com miweb.com

    ServerAdmin webmaster@localhost
    DocumentRoot /home/webs/miweb

    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /home/webs/miweb>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        #Order allow,deny
        #allow from all
        Require all granted
    </Directory>

[ 23 líneas escritas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Algunas consideraciones:

- En la primera línea resaltada de amarillo se define el hostname del sitio y el puerto de conexión: `www.miweb.com:80`
  - Además, también queremos que se resuelva la dirección escribiendo `miweb.com`. Así pues, definimos un `ServerAlias`: `miweb.com`
  - En la tercera línea resaltada en amarillo estamos definiendo la ruta donde se aloja el contenido del sitio, en nuestro caso `/home/webs/miweb`. Aprovechamos para permitir seguir enlaces simbólicos y multiviews: `FollowSymLinks MultiViews`.
  - La directiva `Require all granted` da acceso público a todos los recursos del directorio.
4. El siguiente paso es habilitar como página de inicio el `index.html`. Esto se consigue a través del módulo `mod_dir`.

```
aelop@debian: /etc/apache2/mods-available$ cat dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

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

5. Para habilitar el host que acabamos de configurar hay dos opciones. Por un lado, como comentamos previamente, se puede crear un enlace simbólico a `miweb.conf` en la carpeta `/sites-enabled`:

```
cd /etc/apache2/sites-enabled && sudo ln -s miweb.conf ../sites-
```

```
available/miweb.conf
```

De manera alternativa, también puede hacerse uso del script `a2ensite` para habilitarlo (`a2dissite` para deshabilitar):

```
sudo a2ensite miweb
```

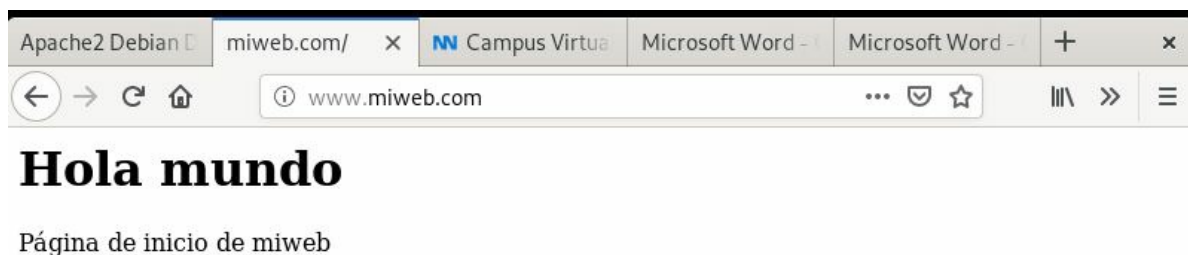
6. Para terminar, es necesario reiniciar el servidor Apache con el siguiente comando:

```
sudo systemctl apache2 restart
```

7. Como las pruebas las estamos realizando en un entorno de pruebas local los nombres de dominio no van a resolverse, ya que no están propagados en ningún servidor DNS todavía. Para simular esto, añadiremos la siguiente línea al fichero `/etc/hosts` local:

```
127.0.0.1 www.miweb.com miweb.com
```

8. Verificamos que todo funciona correctamente accediendo al sitio web desde el navegador:

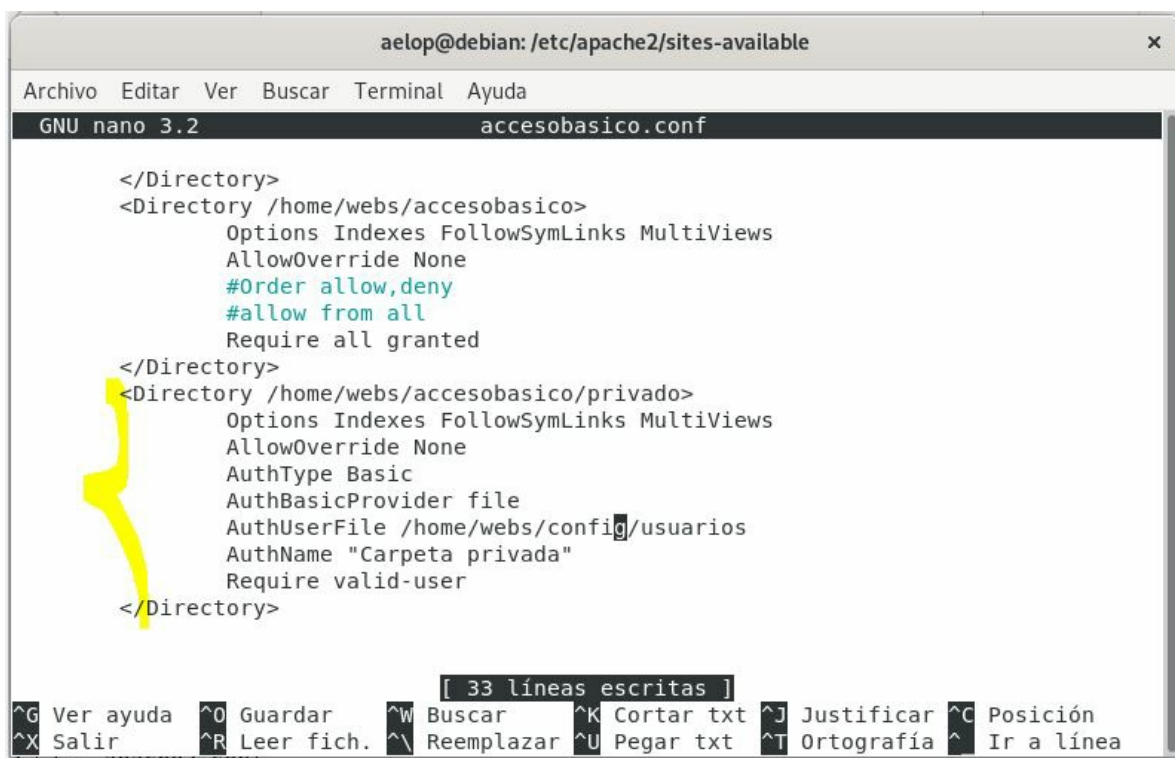


VirtualHost: [www.accesobasico.com](http://www.accesobasico.com)

1. El contenido de este sitio se alojará en una nueva carpeta dentro de `/home/webs` llamada `/accesobasico`. A su vez tendremos un subdirectorio con recursos de acceso privado al que llamaremos `/privado`.

```
sudo mkdir -p /home/webs/accesobasico/privado
```

2. A continuación configuramos el VirtualHost en un fichero `accesobasico.conf` del mismo modo que [www.miweb.com](http://www.miweb.com), con la salvedad de que aquí es necesario incluir una directiva para definir el nivel de seguridad del directorio `/home/webs/accesobasico/privado`.



```
aelop@debian: /etc/apache2/sites-available
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 3.2 accesobasico.conf

</Directory>
<Directory /home/webs/accesobasico>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    #Order allow,deny
    #allow from all
    Require all granted
</Directory>
<Directory /home/webs/accesobasico/privado>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    AuthType Basic
    AuthBasicProvider file
    AuthUserFile /home/webs/config/usuarios
    AuthName "Carpeta privada"
    Require valid-user
</Directory>

[ 33 líneas escritas ]
^G Ver ayuda  ^O Guardar    ^W Buscar     ^K Cortar txt ^J Justificar ^C Posición
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar txt   ^T Ortografía ^_ Ir a línea
```

En la captura anterior está marcado en amarillo el fragmento donde se define el tipo de autenticación necesaria para acceder al recurso privado. Esta consiste en un login básico de usuario/contraseña. La información de usuarios se encuentra en un fichero al que hemos llamado `usuarios`. Existe una utilidad para generar este tipo de ficheros, y se utiliza del siguiente modo:

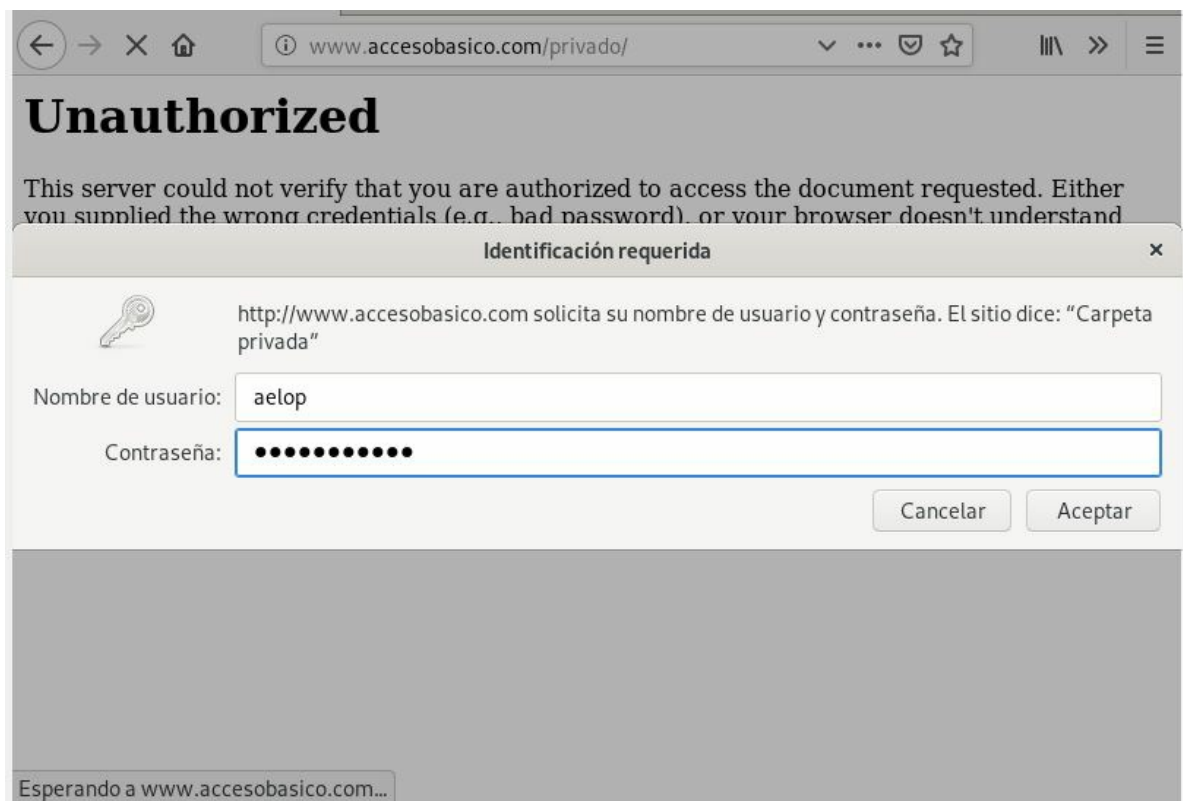
```
aelop@debian:/usr/sbin$ sudo htpasswd -c /home/webs/config/usuarios aelop
[sudo] password for aelop:
New password:
Re-type new password:
Adding password for user aelop
aelop@debian:/usr/sbin$ sudo htpasswd /home/webs/config/usuarios usuario2
New password:
Re-type new password:
Adding password for user usuario2
```

Aquí hemos generado dos usuarios, *aelop* y *usuario2*.

3. Ya solo quedaría habilitar el nuevo sitio con `a2ensite`, reiniciar el servidor y añadir el hostname [www.accesobasico.com](http://www.accesobasico.com) al fichero `/etc/hosts`. Una vez hecho esto, podemos



comprobar que para acceder al recurso privado nos pide identificación de usuario:



Ahí va eso fiera, titán, mastodonte, tiburón!!

## VirtualHost: [www.missl.com](http://www.missl.com)

Para el tercer ejercicio vamos a configurar un sitio web con cifrado SSL. Esto quiere decir que configuraremos el VirtualHost para que sea accesible tanto por el puerto 80 (http) como por el 443 (https).

1. Antes de nada hay que verificar que tenemos instalado los *wrappers* de soporte para SSL en Debian. En nuestro caso estaban ya instalados:

```
aelop@debian:/usr/sbin$ sudo apt-get install ssl-cert
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
ssl-cert ya está en su versión más reciente (1.0.39).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 16 no actualizados.
```

2. Seguidamente lo que haremos será generar el certificado RSA que identificará al servidor. Para ello utilizaremos las herramientas de OpenSSL:

```
openssl genrsa -out server.key 2048
```

Este comando genera una clave privada RSA de 2048 bits cifrada con triple DES (PKCS#12) y la guarda en un fichero llamado *server.key*

```
aelop@debian: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
aelop@debian:~$ openssl genrsa -des3 -out server.key 2048  
Generating RSA private key, 2048 bit long modulus (2 primes)  
.+++++  
.....+++++  
e is 65537 (0x010001)  
Enter pass phrase for server.key:  
Verifying - Enter pass phrase for server.key:  
aelop@debian:~$ openssl req -new -key server.key -out server.csr  
Enter pass phrase for server.key:  
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]:cordoba  
Locality Name (eg, city) []:cordoba  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:masterd  
Organizational Unit Name (eg, section) []:masterd aelop  
Common Name (e.g. server FQDN or YOUR name) []:aelop  
Email Address []:alopez@gmail.com  
  
Please enter the following 'extra' attributes  
to be sent with your certificate request
```

3. También generamos el certificado o clave pública a partir de la clave privada y lo guardamos en un fichero llamado *server.crt*:

```
openssl req -x509 -days 365 -key server.key -out server.crt
```

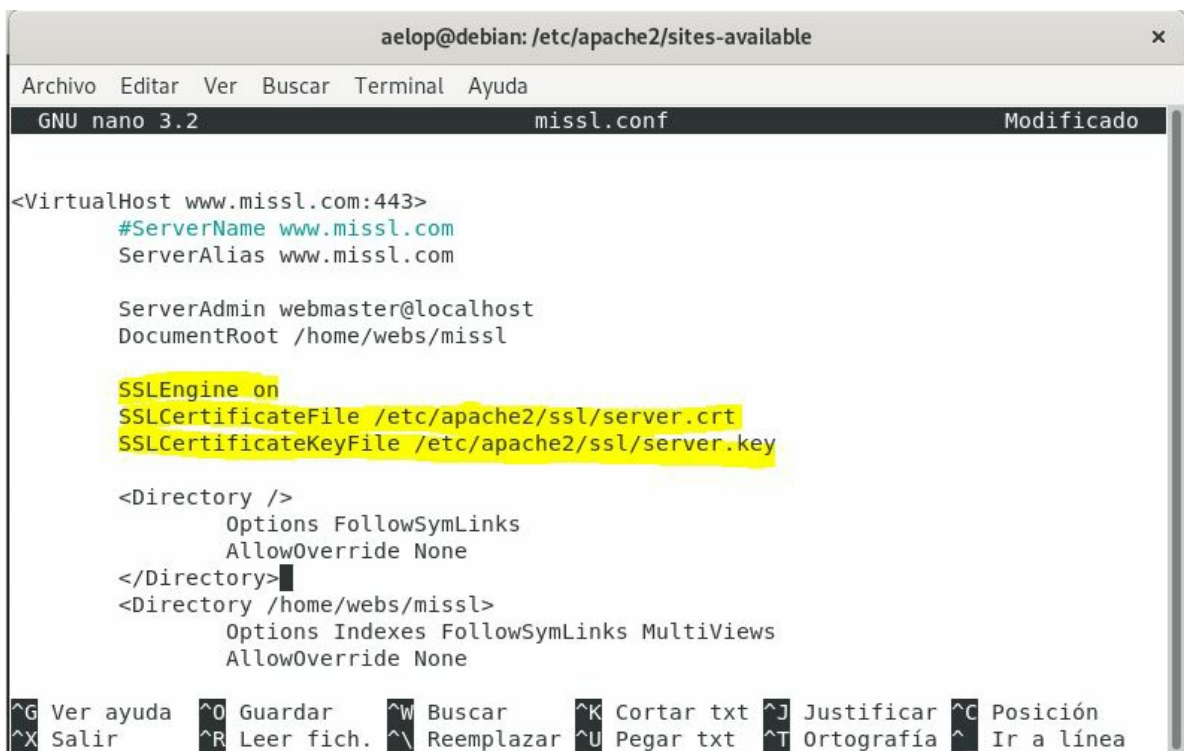
- Con la opción *-x509* pedimos que el certificado sea autofirmado y nos saltamos el paso de generar una solicitud de certificación (ficheros *\*.csr*). Los navegadores mostrarán una alerta de seguridad cuando accedamos al sitio debido a que no ha sido validado por ninguna Entidad de Certificación.
- *-days 365* indica el período de validez del certificado.
- *-key server.key* indica el fichero con la clave privada
- *-out server.crt* indica el fichero de salida donde se guardará el certificado.

Es posible abreviar el paso 2 y 3 en un mismo comando:

```
openssl req -x509 -days 365 -newkey rsa:2048 -keyout server.key -out server.crt
```

4. Ahora configuramos el sitio virtual como hemos hecho en ejercicios anteriores. Así pues, creamos un nuevo directorio en */home/webs/missl/* y el archivo *missl.conf* en el servidor Apache. Éste tendrá dos directivas *VirtualHost*, una para el puerto 80 y otra para el 443.

Primero vamos a ver la del puerto 443:



```
aelop@debian: /etc/apache2/sites-available
GNU nano 3.2 missl.conf Modificado

<VirtualHost www.missl.com:443>
    #ServerName www.missl.com
    ServerAlias www.missl.com

    ServerAdmin webmaster@localhost
    DocumentRoot /home/webs/missl

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/server.crt
    SSLCertificateKeyFile /etc/apache2/ssl/server.key

    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /home/webs/missl>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
    </Directory>

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Como se puede observar, estamos configurando un VirtualHost en el puerto 443 para el hostname [www.missl.com](http://www.missl.com). El cifrado SSL se habilita mediante la directiva `SSLEngine On`, indicándole las rutas al certificado y la clave del servidor. Ambos ficheros deben encontrarse en la ruta `/etc/apache2/ssl`.

- `/etc/apache2/ssl/server.key`
- `/etc/apache2/ssl/server.crt`

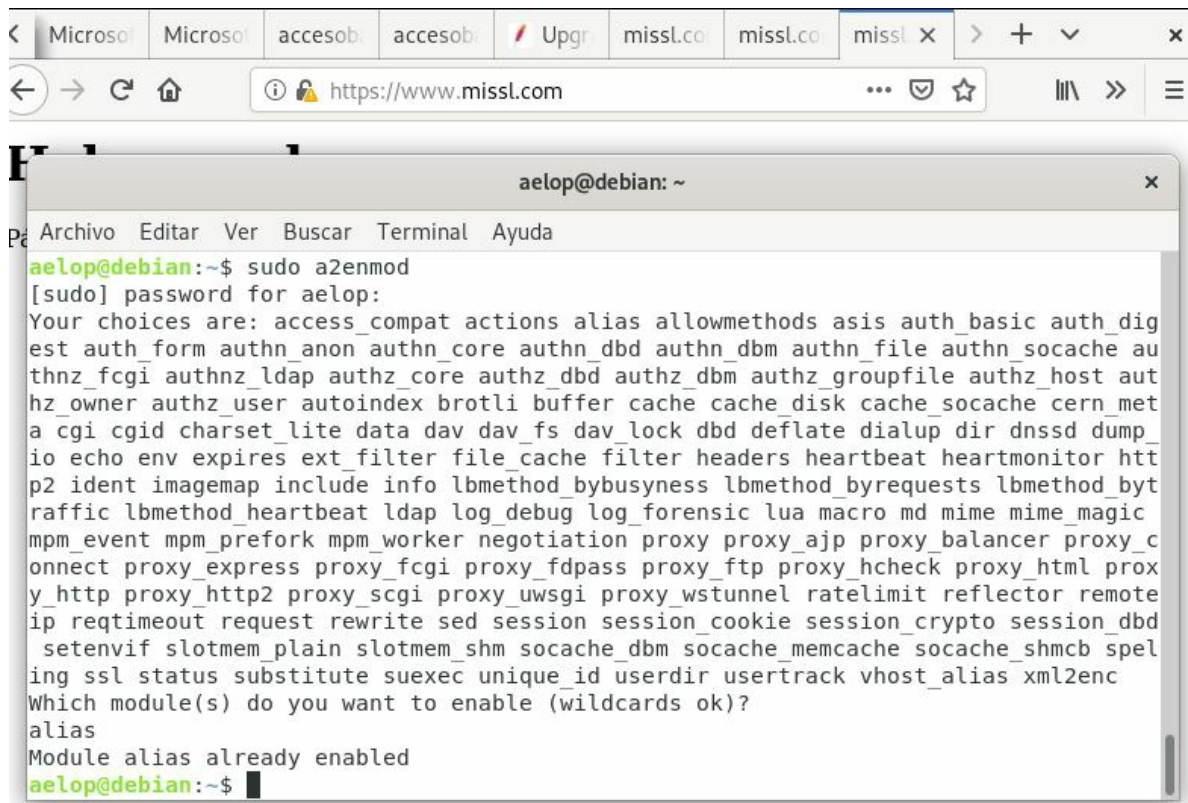
5. Por otro lado, dentro del mismo archivo `missl.conf` configuraremos otro VirtualHost para trabajar con las conexiones del puerto 80. Como queremos que todos los usuarios que accedan a la dirección [www.missl.com](http://www.missl.com) pasen a través de conexión segura necesitamos redirigir las peticiones enviadas al puerto 80 hacia el puerto 443. Para ello incluiremos la directiva `Redirect` al final de éste VirtualHost:

```
<VirtualHost www.missl.com:80>
    [...]
    Redirect / https://www.missl.com
</VirtualHost>
<VirtualHost www.missl.com:443>
    [...]
</VirtualHost>
```

6. La redirección es una capacidad de Apache que puede habilitarse o deshabilitarse a

través del módulo `mod_alias`. Para que funcione deberemos activarlo con el comando:

```
sudo a2enmod alias
```



7. A su vez, el cifrado SSL en Apache también depende de un módulo. Concretamente `mod_ssl`. Nos aseguramos de que también esté habilitado:

```
sudo a2enmod ssl
```

8. Para terminar añadimos [www.missl.com](http://www.missl.com) al fichero `/etc/hosts` y reiniciamos el servidor Apache.

```
127.0.0.1 www.miweb.com miweb.com www.accesobasico.com www.missl.com
```

```
sudo systemctl apache2 restart
```