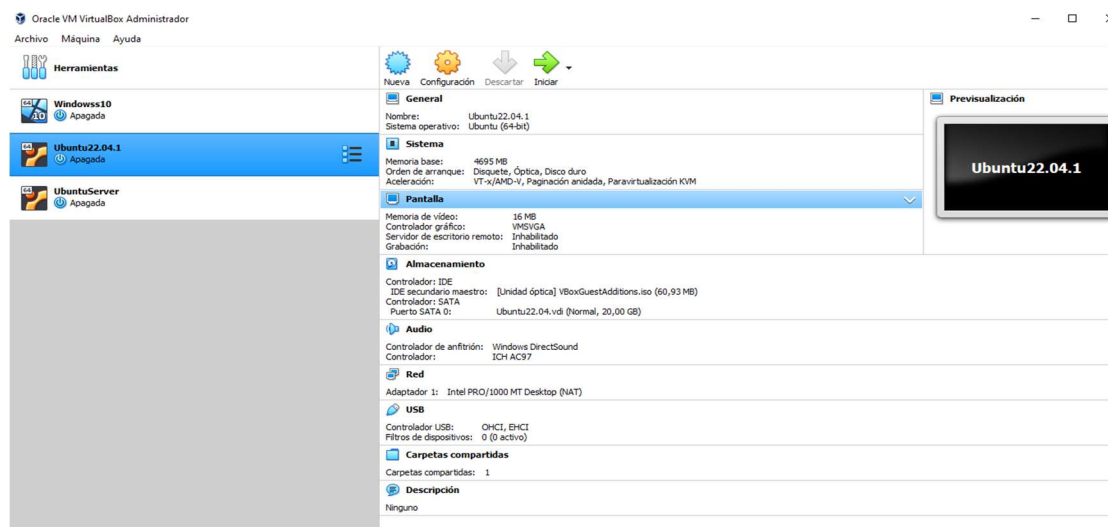


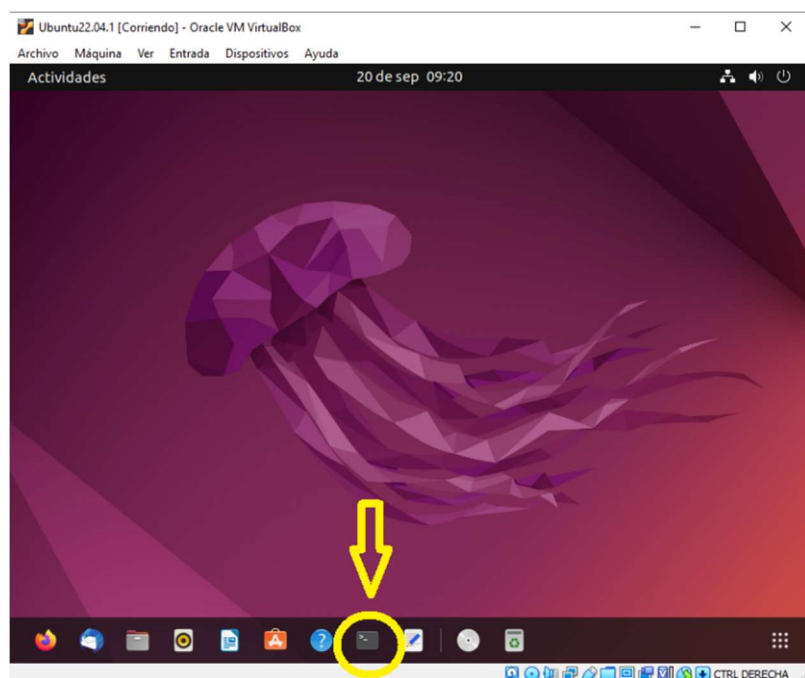
# Trabajo final de Administración de servidores web (Apache)

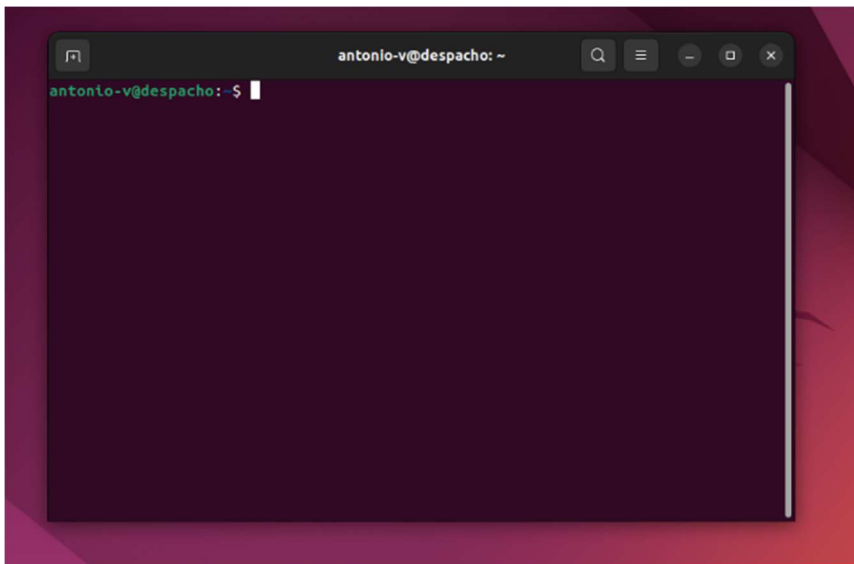
## Creación de un sitio web llamado [www.miweb.com](http://www.miweb.com)

Para la realización de este trabajo creamos una máquina virtual con VirtualBox e instalamos Ubuntu.



Una vez instalada la máquina virtual e instalado y configurado Ubuntu, accedemos al escritorio de Ubuntu y seleccionamos el terminal, este es el icono de la barra de tareas que aparece marcado de amarillo según la siguiente imagen.





Una vez dentro del terminal y antes de cualquier instalación, debemos actualizar la lista de repositorios escribiendo el código “sudo apt update”, le damos a intro y nos solicita la contraseña del usuario “antonio-v”, que es el nuestro.

A terminal window showing the command 'sudo apt update' being executed. The output lists updates for various repositories, including 'jammy InRelease', 'jammy-updates InRelease', 'jammy-security InRelease', 'jammy-backports InRelease', and 'jammy-updates/main amd64 DEP-11 Metadata'. The command 'sudo apt update' is circled in yellow.

```
antonio-v@despacho: ~$ sudo apt update
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Des:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease [99,8 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [92,8 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [240 kB]
Des:7 http://es.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Des:8 http://es.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [12,3 kB]
Des:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [332
```

Esperamos que termine y después bajamos e instalamos los paquetes escribiendo el código “sudo apt upgrade” y le damos a intro.

A terminal window showing the command 'sudo apt upgrade' being executed. The output shows the list of packages to be upgraded, including 'linux-headers-5.15.0-48', 'linux-headers-5.15.0-48-generic', and 'linux-image-5.15.0-48-generic'. The command 'sudo apt upgrade' is circled in yellow.

```
antonio-v@despacho: ~$ sudo apt upgrade
Leyendo la información de estado... Hecho
Se pueden actualizar 34 paquetes. Ejecute «apt list --upgradable» para verlos.
antonio-v@despacho: ~$ sudo apt upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Se instalarán los siguientes paquetes NUEVOS:
 linux-headers-5.15.0-48 linux-headers-5.15.0-48-generic
 linux-image-5.15.0-48-generic linux-modules-5.15.0-48-generic
```

Durante la instalación nos pregunta si deseamos continuar con la instalación, informándonos del espacio que van a ocupar los archivos que vamos a instalar, presionamos “s” y continua la instalación hasta el final.

```
antonio-v@despacho: ~  
5 actualizados, 5 nuevos se instalarán, 0 para eliminar y 29 no actualizados.  
5 standard security updates  
Se necesita descargar 114 MB/115 MB de archivos.  
Se utilizarán 584 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar [S/n] s  
Des:1 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 linux-modules  
-5.15.0-48-generic amd64 5.15.0-48.54 [22,7 MB]  
Found memtest86+ image: /boot/memtest86+.bin  
Warning: os-prober will not be executed to detect other bootable partitions.  
Systems on them will not be added to the GRUB boot configuration.  
Check GRUB_DISABLE_OS_PROBER documentation entry.  
done  
antonio-v@despacho:~$
```

Una vez terminado, podemos reiniciar Linux con el código “sudo reboot”.

Con Linux reiniciado, nos conectamos vía “ssh” para instalar el Apache2, si no tenemos “ssh” tendremos que instalarlo mediante el código: “sudo apt install openssh-server”.

El servidor “ssh” nos sirve para conectarnos a nuestra máquina de forma remota.

```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo apt install openssh-server
```

Le presionamos intro y nos solicitará la contraseña, posteriormente decimos sí presionando la tecla “s” y esperamos que termine la instalación.

Podemos modificar el fichero “ssh\_config” que se encuentre en la ruta /etc/ssh/ssh\_config y modificar ciertos parámetros como el puerto de escucha, que por defecto es el 22, el tiempo que la sesión va a durar abierta “LoginGraceTime” y muchos más parámetros.

El editor que utilizamos para la modificación de dicho archivo, es el “nano”.



Con el código “sudo service ssh status”, vemos el estado del servicio “ssh”.

```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo service ssh status  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en>  
   Active: active (running) since Tue 2022-09-20 19:33:40 CEST; 24min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
  Process: 828 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
 Main PID: 879 (sshd)  
    Tasks: 1 (limit: 5325)  
  Memory: 3.6M  
     CPU: 29ms  
   CGroup: /system.slice/ssh.service  
           └─879 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
sep 20 19:33:38 despacho systemd[1]: Starting OpenBSD Secure Shell server...  
sep 20 19:33:40 despacho sshd[879]: Server listening on 0.0.0.0 port 22.  
sep 20 19:33:40 despacho sshd[879]: Server listening on :: port 22.  
sep 20 19:33:40 despacho systemd[1]: Started OpenBSD Secure Shell server.  
lines 1-17/17 (END)
```

Salimos de esta pantalla, presionando ctrl + c.

Procedemos a la instalación de apache2, con el código “sudo apt install apache2” presionamos intro y luego nos piden la contraseña, y la introducimos y volvemos a dar intro, esperamos que termine...

```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo apt install apache2
```

Una vez terminada la instalación, podemos ver si funciona Apache haciendo un “sudo service apache2 status”.

```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo service apache2 status  
[sudo] contraseña para antonio-v:  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>  
   Active: active (running) since Tue 2022-09-20 19:33:45 CEST; 12h ago  
     Docs: https://httpd.apache.org/docs/2.4/  
 Main PID: 1057 (apache2)  
    Tasks: 6 (limit: 5325)  
  Memory: 30.0M  
     CPU: 2.941s  
   CGroup: /system.slice/apache2.service  
           └─1057 /usr/sbin/apache2 -k start  
             └─4154 /usr/sbin/apache2 -k start  
               └─4155 /usr/sbin/apache2 -k start
```

Presionamos ctrl + c para salir de la pantalla anterior y reiniciamos apache con “sudo service apache2 restart”

```
sep 21 00:00:03 despacho systemd[1]: Reloading The Apache HTTP Server...
sep 21 00:00:03 despacho apachectl[4125]: AH00558: apache2: Could not reliably
sep 21 00:00:03 despacho systemd[1]: Reloaded The Apache HTTP Server.
lines 1-22/22 (END)
antonio-v@despacho:~$ sudo service apache2 restart
antonio-v@despacho:~$
```

Una vez instalado y reiniciado apache creamos un directorio en “home” para guardar nuestras distintas páginas web que hagamos, creamos el directorio o carpeta con “sudo mkdir /home/webs”

```
antonio-v@despacho: ~
antonio-v@despacho:~$ sudo mkdir /home/webs
antonio-v@despacho:~$
```

Tras la creación del directorio webs, creamos otro dentro, que le llamaremos “miweb” y donde alojaremos nuestro index.html.

```
antonio-v@despacho: ~
antonio-v@despacho:~$ sudo mkdir /home/webs/miweb
antonio-v@despacho:~$
```

Con el editor nano y el código “sudo nano /home/webs/miweb/index.html” creamos y editamos a la vez el fichero index.html y que vamos alojar en el directorio “miweb”.

```
antonio-v@despacho: ~
antonio-v@despacho:~$ sudo nano /home/webs/miweb/index.html
```

Este es el contenido html que hemos puesto en el index.html de “miweb”.

```
GNU nano 6.2 /home/webs/miweb/index.html *
<html>
  <head>
    <title>Bienvenido!</title>
  </head>
  <body>
    <h1>Bienvenido a miweb.com!</h2>
  </body>
</html>
```

También creamos otra carpeta para los log, con el código: “sudo mkdir /var/log/apache2/miweb”.

En esta carpeta guardamos los errores producidos en el servidor.

```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo mkdir /var/log/apache2/miweb
```

Tras la creación de los directorios “log” y los directorios para el alojamiento de nuestro index.html correspondiente a “miweb” creamos el host virtual que será accesible con [www.miweb.com](http://www.miweb.com) y [miweb.com](http://miweb.com)

Primeramente, vamos a la carpeta “sites-available” de apache para crear el archivo de configuración de nuestro host virtual, la ruta de dicha carpeta se encuentra en /etc/apache2/, en apache2 nos encontramos varios directorios y archivos, los que nos interesa son sites-available y sites-enable, el primero es donde tenemos los host virtuales de los distintos sitios web creados, y el segundo es donde tenemos los sitios que están activados.

Creamos y editamos el fichero de configuración de nuestro host virtual, con extensión “.conf”, utilizando el código:

“sudo nano /etc/apache2/sites-available/miweb.com.conf”.

```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo nano /etc/apache2/sites-available/miweb.com.conf
```

Accedemos a nano y añadimos el siguiente código:

```
GNU nano 6.2 /etc/apache2/sites-available/miweb.com.conf *  
<VirtualHost *:80>  
  
    ServerName miweb.com  
    ServerAlias www.miweb.com  
    ServerAdmin admin@localhost  
    DocumentRoot /home/webs/miweb  
  
    <Directory /home/webs/miweb>  
        Options Indexes FollowSymLinks  
        AllowOverride All  
        Require all granted  
    </Directory>  
  
    ErrorLog ${APACHE_LOG_DIR}/miweb/error.log  
    CustomLog ${APACHE_LOG_DIR}/miweb/access.log combined  
  
</VirtualHost>
```

Guardamos el código con ctrl + o y salimos con ctrl + x.

Ahora podemos hacer una comprobación de si el archivo tiene un error sintáctico, con el código “apachectl configtest” y nos devuelve “Syntax OK”



```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo nano /etc/apache2/sites-available/miweb.com.conf  
antonio-v@despacho:~$ apachectl configtest  
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  
Syntax OK  
antonio-v@despacho:~$
```

Reiniciamos Apache con:

“sudo systemctl reload apache2 | sudo service apache2 restart”

```
antonio-v@despacho:~$ sudo systemctl reload apache2 | sudo service apache2 restart  
antonio-v@despacho:~$
```

Ahora editamos el archivo “hosts”

“Sudo nano /etc/hosts”

```
GNU nano 6.2 /etc/hosts *  
127.0.0.1 localhost  
127.0.1.1 despacho  
  
127.0.0.1 mipagina.com  
127.0.0.1 prueba1.com  
127.0.0.1 unidad3.com  
127.0.0.1 miweb.com  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

Añadimos la línea “127.0.0.1 miweb.com”, guardamos con ctrl + o + intro + ctrl + x, y salimos.

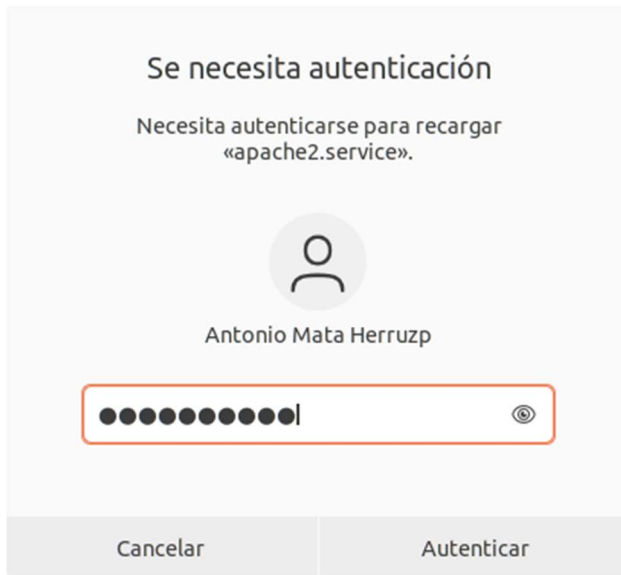
Activamos la configuración del host virtual con:

“sudo a2ensite miweb.com.conf” y posteriormente nos pide que recarguemos apache con el código:

“systemectl reload apache2”


```
antonio-v@despacho:~$ sudo a2ensite miweb.com.conf  
Enabling site miweb.com.  
To activate the new configuration, you need to run:  
systemctl reload apache2  
antonio-v@despacho:~$ systemctl reload apache2
```

Una vez que hemos mandado recargar apache nos solicita la autenticación del usuario.



Se necesita autenticación

Necesita autenticarse para recargar  
«apache2.service».

  
Antonio Mata Herruzp

Cancelar Autenticar

Y ya está listo para probar nuestra web en un navegador:

Abrimos el navegador que tiene por defecto Ubuntu y en la barra de direcciones ponemos miweb.com y nos mostrará nuestra página web:



# Bienvenido a miweb.com!

## Ejercicio: [www.accesobasico.com](http://www.accesobasico.com)

Creamos un nuevo sitio web llamado accesobasico.com, de la misma forma que realizamos el ejercicio anterior.

Esta web la alojaremos en un directorio llamado “accesobasico” y este a su vez en el directorio webs, la carpeta “accesobasico” será una carpeta privada que solo tendrán acceso tres usuarios, Pepe, Ana y Eva.

Lo primero que tendremos que hacer es crear en un archivo el usuario y contraseña que dará acceso a la carpeta “accesobasico”, este archivo lo vamos a colocar dentro de la carpeta “accesobasico”, (no es necesaria esta ubicación, se puede colocar en cualquier otro lugar o crear otro directorio para ubicar el archivo), nos dirigimos a la carpeta y vemos que es lo que contiene, utilizaremos el “ls -a” para ver si hay archivos ocultos.



```
antonio-v@despacho: /home/webs/accesobasico
antonio-v@despacho:/home/webs/accesobasico$ cd /home/webs/accesobasico/
antonio-v@despacho:/home/webs/accesobasico$ ls -a
.  ..  index.html
antonio-v@despacho:/home/webs/accesobasico$
```

La “a” de “ls -a” es para que muestre también los archivos ocultos y en esta ocasión vemos que no hay ninguno en la carpeta “accesobasico”.

Ahora creamos el archivo que va a contener los usuarios y las contraseñas de acceso a la carpeta con la instrucción:

“htpasswd -c .htpasswd nombre\_user”, el código “-c” se utiliza con el primer usuario que metamos, en los sucesivos sería la misma instrucción menos con la “-c” quitada “htpasswd .htpasswd nombre\_user”, el “.htpasswd” es el nombre del archivo donde se encuentran las contraseña y usuario, lleva un punto delante el cual indica que es un archivo oculto.

Una vez puesto la instrucción con el nombre del usuario, nos pedirá que pongamos una contraseña, tendremos que repetirla nuevamente.

```
antonio-v@despacho: /home/webs/accesobasico
antonio-v@despacho:/home/webs/accesobasico$ cd /home/webs/accesobasico/
antonio-v@despacho:/home/webs/accesobasico$ ls -a
.  ..  index.html
antonio-v@despacho:/home/webs/accesobasico$ sudo htpasswd -c .htpasswd pepe
[sudo] contraseña para antonio-v:
New password:
Re-type new password:
Adding password for user pepe
antonio-v@despacho:/home/webs/accesobasico$ sudo htpasswd .htpasswd ana
New password:
Re-type new password:
Adding password for user ana
antonio-v@despacho:/home/webs/accesobasico$ sudo htpasswd .htpasswd eva
New password:
Re-type new password:
Adding password for user eva
antonio-v@despacho:/home/webs/accesobasico$
```

En esta imagen mostramos como son las instrucciones para crear el fichero y los distintos usuarios.

Si hacemos un “ls -a” nos muestra lo que hay en el directorio “accesobasico”, y nos muestra el fichero index.html y el fichero “.htpasswd” que es un fichero oculto.

```
antonio-v@despacho: /home/webs/accesobasico
antonio-v@despacho:/home/webs/accesobasico$ ls -a
.  ..  .htpasswd  index.html
antonio-v@despacho:/home/webs/accesobasico$
```

Si queremos, podemos ver lo que hay dentro del archivo .htpasswd con “nano”:

```
antonio-v@despacho: /home/webs/accesobasico
GNU nano 6.2 .htpasswd
pepe:$apr1$IugpHn4F$U0eZ7gGDYw/50wsj5NBi0/
ana:$apr1$PD6lSmUz$BL1.GSKc3/oxK1XL0588y.
eva:$apr1$nCj2zp04$RygcCUu/nrAMEM6.2nnTl0
```

Después de crear el archivo y los usuarios con sus contraseñas, tenemos que modificar o configurar el archivo de configuración del host virtual (accesobasico.com.conf) e introducirle las instrucciones que vemos en la imagen:

```
antonio-v@despacho: ~
GNU nano 6.2 /etc/apache2/sites-available/accesobasico.com.conf
VirtualHost *:80>

    ServerName accesobasico.com
    ServerAlias www.accesobasico.com
    ServerAdmin admin@localhost
    DocumentRoot /home/webs/accesobasico

    <Directory /home/webs/accesobasico>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    <Directory /home/webs/accesobasico>
        Order deny,allow
        AuthUserFile "/home/webs/accesobasico/.htpasswd"
        AuthName "Acceso permitido"
        AuthType Basic
        <Limit GET POST>
            Require valid-user
        </Limit>
    </Directory>

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

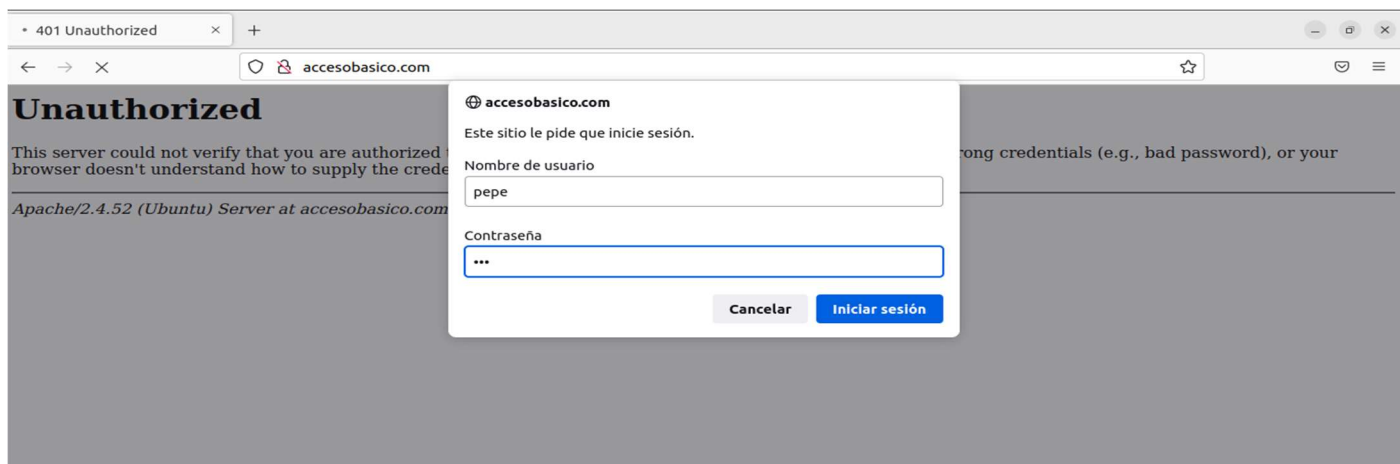
Al hacer una modificación al archivo de configuración “accesobasico.com.conf” tenemos que habilitarlo con la instrucción:

“Sudo a2ensite accesobasico.com.conf”

Y reseteamos apache con: “sudo apachectl restart”.

Una vez hecho todo esto solo queda probarlo en el navegador:

Introducimos accesobasico.com y nos muestra:



Al introducir el usuario y la contraseña nos puede salir otro letrero preguntando si queremos guardar la sesión del usuario y contraseña introducida, tras esto, si el usuario y la contraseña son correcta accedemos al sitio web.



**Bienvenido a accesobasico.com!**

**Solo puede acceder Pepe, Ana y Eva**

## Ejercicio: crear un host virtual con certificado ssl

En este como en el ejercicio anterior creamos un host virtual llamado [www.missl.com](http://www.missl.com)

Primeramente, creamos el sitio como lo hemos hecho anteriormente, aquí mostramos las instrucciones resumidas:

```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo mkdir /home/webs/missl  
[sudo] contraseña para antonio-v:  
antonio-v@despacho:~$ sudo mkdir /var/log/apache2/missl  
antonio-v@despacho:~$ sudo nano /home/webs/miweb/index.html  
antonio-v@despacho:~$ sudo nano /home/webs/missl/index.html  
antonio-v@despacho:~$ sudo nano /etc/apache2/sites-available/missl.com.conf  
antonio-v@despacho:~$ apachectl configtest  
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  
Syntax OK  
antonio-v@despacho:~$ sudo systemctl reload apache2 | sudo service apache2 restart  
antonio-v@despacho:~$ sudo nano /etc/hosts  
antonio-v@despacho:~$ sudo a2ensite missl.com.conf  
Enabling site missl.com.  
To activate the new configuration, you need to run:  
    systemctl reload apache2  
antonio-v@despacho:~$ systemctl reload apache2  
antonio-v@despacho:~$ service apache2 reload  
antonio-v@despacho:~$
```

Para crear el certificado ssl debemos de tener instalado el “openssl”, para ello lo primero es hacer un “sudo apt update” y seguidamente “sudo apt install apache2 openssl”, después activamos el módulo ssl en apache con “sudo a2enmod ssl”, nos pide la contraseña y reiniciamos apache “systemctl restart apache2”.

Para crear el certificado y la clave utilizaremos la instrucción openssl:

```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/clave.key -out /etc/ssl/certs/missl-certificate.crt
```



Cuando presionamos la tecla intro se va a ir generando el certificado y nos irá haciendo una serie de preguntas que tenemos que contestarlas:

```

antoni@antoni:~$ openssl req -x509 -nodes -newkey rsa:2048 -keyout key.pem -out cert.pem
-----
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]:Toledo
Locality Name (eg, city) []:Torrijos
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Antonio Mata
Organizational Unit Name (eg, section) []:Servidores
Common Name (e.g. server FQDN or YOUR name) []:www.missl.com
Email Address []:admin@missl.com
antoni@antoni:~$

```

Le indicamos el país, la provincia, la población, nombre de la empresa, unidad o departamento, la dirección web o la IP y por último una dirección de correo electrónico, una vez confirmado todo, ya tenemos generados los dos ficheros, uno con la clave y el otro con el certificado.

## Ejercicio: Crear una carpeta privada en un sitio web con certificado SSL.

Creamos un nuevo sitio web con certificado SSL como en el ejercicio anterior:

En esta ocasión la carpeta privada la vamos a crear para el sitio web <https://missl1.com>.

Lo primero es crear un archivo oculto que irá dentro de la carpeta missl1 y que contendrá el usuario y contraseña de acceso al sitio web.

```
antonio-v@despacho: ~  
antonio-v@despacho:~$ sudo htpasswd -c /home/webs/missl1/.htpasswd antonio  
New password:  
Re-type new password:  
Adding password for user antonio  
antonio-v@despacho:~$
```

Esta vez solo creamos un usuario.

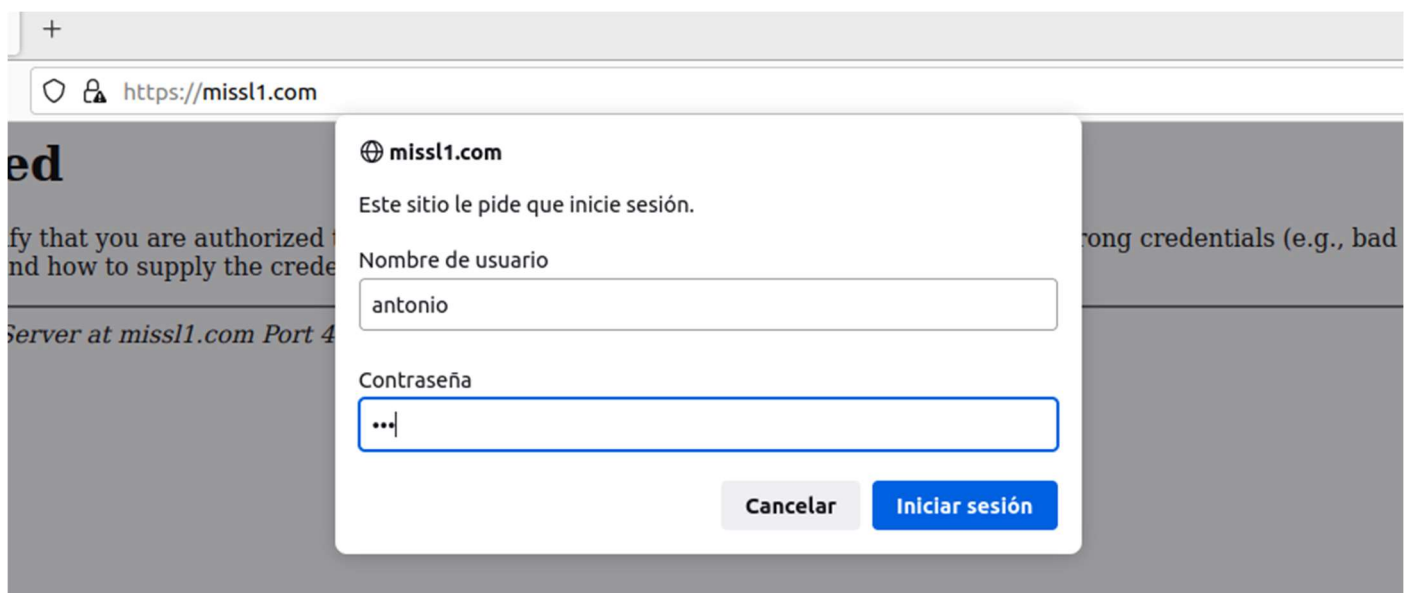


Una vez creado los log, el virtual host, el archivo .htpasswd, el certificado y clave SSL, procedemos a introducir la directiva correspondiente en el archivo “misssl1.com.conf” y que requiera el usuario y contraseña para acceder al sitio web, estas instrucciones las insertamos tanto en el contenedor del puerto 80 como en el del puerto 443.

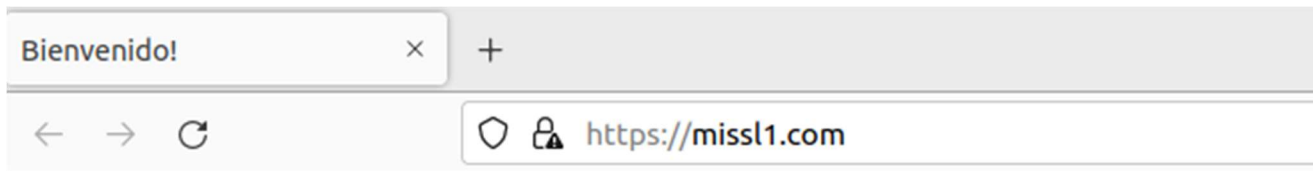
```
antonio-v@despacho: ~  
GNU nano 6.2 /etc/apache2/sites-available/misssl1.com.conf  
<VirtualHost *:443>  
  
    ServerName misssl1.com  
    ServerAlias www.misssl1.com  
    ServerAdmin root@localhost  
  
    DocumentRoot /home/webs/misssl1  
  
    <Directory /home/webs/misssl1>  
        Require all granted  
    </Directory>  
    <Directory /home/webs/misssl1>  
        Order deny,allow  
        AuthUserFile "/home/webs/misssl1/.htpasswd"  
        AuthName "Acceso permitido"  
        AuthType Basic  
        <Limit GET POST>  
            Require valid-user  
        </Limit>  
    </Directory>  
  
    ErrorLog ${APACHE_LOG_DIR}/misssl1/error.log  
    CustomLog ${APACHE_LOG_DIR}/misssl1/access.log combined  
  
    SSLEngine on  
    SSLCertificateFile /etc/ssl/certs/misssl1-certificate.crt
```

Guardamos, activamos el virtual host, reiniciamos apache y ya está.

Nos vamos al navegador, introducimos nuestro sitio “misssl1.com” y como se muestra en la imagen nos solicita el usuario y contraseña.



Una vez introducido el usuario y la contraseña, accedemos al sitio web sin problema, como se puede ver en la imagen.



**Bienvenido a missl1.com!**

**Solo puede acceder a este sitio Antonio**

**Fin del trabajo: Administración de servidores web (Apache)**

Antonio Mata Herruzo