

# SERVIDORES WEB

## //2021



UBUNTU-DESKTOP  
VIRTUALBOX  
APACHE

SONIA ORIVE GIL

# CONTENIDO



03

*INTRODUCCIÓN*

06

*MÁQUINA VIRTUAL*

08

*CONFIGURAR ORACLE VM  
VIRTUALBOX*

12

*CONFIGURAR UBUNTU-  
DESKTOP*

15

*INSTALAR CLIENTE SSH*

18

*INSTALAR SERVIDOR  
APACHE*

20

*DIRECTORIOS*

---

22

PERMISOS DE USUARIO

---

---

24

CREAR DIRECTORIO webs

---

---

25

CREAR HOSTS VIRTUALES  
HOST www.miweb.com  
ALIAS miweb.com

---

---

31

CREAR CARPETA PRIVADA

---

---

33

CREAR CARPETA PRIVADA  
www.acceso.com

---

---

36

CREAR CARPETA PRIVADA  
www.accesobasico.com

---

---

39

HOST CON CERTIFICADO SSL  
www.missl.com

---



# INTRODUCCIÓN

Las herramientas que vamos a utilizar para desarrollar este ejercicio son:

- **Una máquina física**, un ordenador con el sistema operativo windows 10.
- **Oracle VM Virtualbox 6.1.18**, consiste en un software de virtualización para arquitecturas x86/amd64.
  - **Editor de texto GNU Nano**, para crear, modificar y lectura de archivos.
  - **Mozilla FireFox**, navegador web desarrollado para distintas plataformas.
  - **Ubuntu 20.04.2.0 desktop**, consiste en una versión de escritorio del sistema operativo Linux.
- **PuTTy-64bit-0.74**, emulador de terminal que soporta SSH y otros protocolos.
- **Apache2**, software que se ejecuta en un servidor web para la ejecución de los hostVirtuales.

**Oracle VirtualBox** es una aplicación gratuita, que sirve para la configuración de máquinas virtuales con instalaciones de sistemas operativos. Es decir nos permite crear servidores virtuales.

Se puede descargar desde este enlace:

<https://www.virtualbox.org/wiki/Downloads>

**Ubuntu** es un sistema operativo de código abierto y licencia libre.

Utilizado para denominar una distribución GNU/LINUX. Además, tiene un entorno gráfico gnome.

Compuesto por diversos paquetes de software, con un conjunto de aplicaciones incorporadas para satisfacción de sus usuarios.

Está enfocado a la facilidad de uso e instalación, pensado para una fácil interacción del usuario.

Se puede descargar desde este enlace:

<https://releases.ubuntu.com/20.04/>

**GNU Nano** es el editor de texto para la terminal, consola (shell), predeterminado en Ubuntu.

Permite:

- Modificar, leer, copiar archivos de configuración que se encuentra en un servidor, sin interfaz gráfica.
- Escribir código.
- Hacer y deshacer acciones.
- Operaciones de búsqueda.

**FireFox**, es un navegador web libre y de código abierto desarrollado para distintas plataformas.

Usa el motor Gecko para renderizar páginas webs, el cual implementa actuales y futuros estándares web.

Predeterminado en Ubuntu.

Se puede descargar desde este enlace:

<https://www.mozilla.org/es-ES/firefox/>

**PuTTY** es un cliente ssh de acceso a servidores por medio de Raw, Serial, Telnet o SSH.

Permite realizar conexiones con otra máquina en modo remoto o mediante serial.

Utilizado para la gestión de un servidor o una máquina remota con Linux, sin interfaz gráfica, desde cualquier ordenador.

Una conexión ssh es un protocolo que permite conectar dos ordenadores, creando la posibilidad de realizar un acceso remoto mediante un canal seguro y con la información cifrada.

Se puede descargar desde este enlace:

<https://www.putty.org/>

**Apache2** es un software especializado en ofrecer servicios de servidor web http y https.

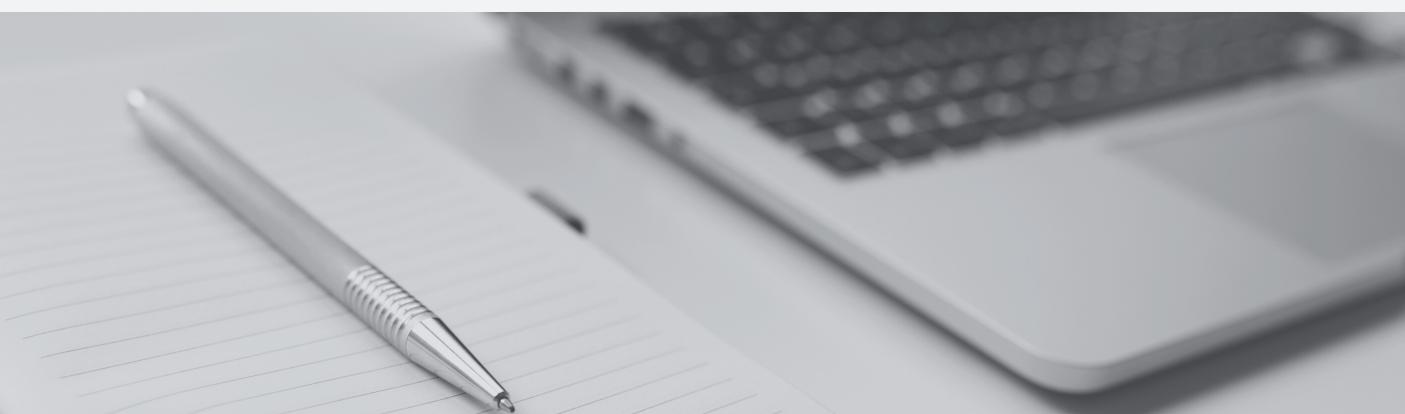
Es versátil, ligero y muy útil, con el cual se ejecutan sitios web. Además de ser gratuito y de código abierto para plataformas Unix, Windows, Macintosh y otras, que implementan el protocolo HTTP/1.1 y la noción de sitio virtual.

Su nombre completo es Apache HTTP Server.

Permite a sus propietarios servir contenido web, modificarlo según sus necesidades, gestionar todos los ficheros alojados en el servidor de forma fácil y sencilla.

Se puede descargar desde este enlace:

<https://httpd.apache.org/download.cgi>



---

Todas las herramientas utilizadas son gratuitas y de código abierto.

Primero, crearemos una máquina virtual con Virtual Box en nuestra máquina física, para instalar el software necesario posteriormente. Lo que es una ventaja, ya que nos permite realizar los mínimos cambios en nuestro ordenador.

Segundo, la instalación y configuración del software ubuntu 20.04.0. en una de sus versiones de escritorio , lo que pone a disposición del usuario una interfaz gráfica, en la máquina virtual creada.

Tercero, la instalación del cliente ssh (putty) en nuestro ordenador y realizar una conexión en remoto a la máquina virtual para comprobar que se realiza correctamente.

Pero todas las instalaciones requeridas y acciones necesarias para el desarrollo de los ejercicios han sido realizadas por línea de comandos desde la terminal (shell) del escritorio. Aunque también se pueden realizar desde el cliente ssh para trabajar en un entorno más real.

Cuarto instalaremos el servidor web apache, versión Apache2, desde los repositorios de Ubuntu que cuentan con herramientas para la activación de módulos y el uso de hosting virtuales.

En el directorio de publicación de Apache, crearemos nuestras carpetas o directorios dónde almacenaremos nuestra web.

En lo referente a la web he creado una carpeta compartida en la máquina virtual que me permite subir los archivos de la web que hice en el módulo de html de este curso..

El desarrollo de los ejercicios han sido expuestos desde el punto de vista de un manual, dado que en nuestro trabajo la formación es una de las actividades que podemos desempeñar. Por lo que incluyo breves o básicas explicaciones de la línea de comandos, en que consisten los permisos de usuario, desplazamientos entre directorios...

Para el ejercicio de autenticación de usuarios, utilizaremos el sistema de Autenticación básico que tenemos a nuestra disposición en el servidor web Apache. Consiste en la creación de un archivo en el que almacenaremos los usuarios y sus respectivas contraseñas. También podemos realizarlo por grupos de usuarios.

En el certificado ssl, crearemos un certificado autofirmado que cifrara la comunicación entre el servidor y cualquier cliente. Sin embargo, dado que no está firmado por ninguna de las autoridades certificadoras de confianza incluidas con los navegadores y los sistemas operativos, el resultado será un mensaje de error de seguridad cuando visiten el sitio web.

Pero la transmisión de datos entre cliente y servidor estarán cifradas y solo se podrá utilizar para una intranet, un entorno local, por ejemplo.

# MÁQUINA VIRTUAL

## → ¿Qué es una máquina virtual?

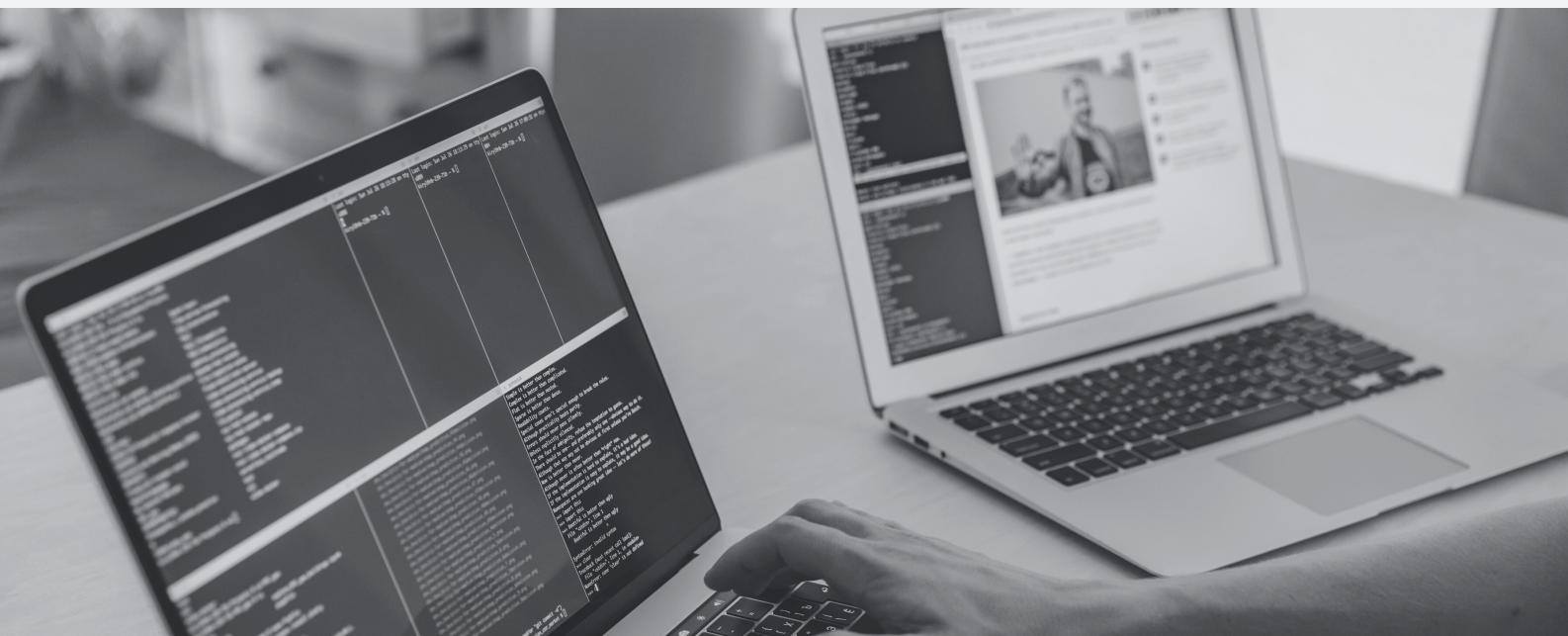
En términos generales, una máquina virtual es un software que permite emular el funcionamiento de un ordenador dentro de otro ordenador gracias a un proceso de encapsulamiento que aísla a ambos. Pero manteniendo, a cada cual, la independencia de su huésped.

Este software crea una capa independiente, donde se emula el funcionamiento de un ordenador real con todos los componentes de hardware que necesita para funcionar (disco duro, memoria RAM, tarjetas de red, tarjeta gráfica, etc.) y que puede ejecutar cualquier sistema operativo o programa, tal y como lo haría un ordenador real. Toda esta emulación se encapsula en una serie de archivos que actúan como contendor desde el que se ejecuta la máquina virtual en una ventana de tu ordenador como si de un programa más se tratara y sin que nada de lo que suceda en el interior de esa ventana afecte al ordenador que la ejecuta.

A grandes rasgos, la única diferencia entre tu ordenador real y una máquina virtual que ejecutes en ese mismo ordenador, es que tu ordenador sí cuenta con un hardware real, mientras que la máquina virtual emula todos sus componentes de forma que no tiene por qué corresponderse con el hardware físico que tienes instalado en tu ordenador real.

De ese modo, se pueden ajustar las características del hardware para hacerlo compatible con el sistema operativo que vayas a usar en esa máquina virtual. Es decir, aunque tu hardware real no sea compatible con un determinado sistema operativo, el de una máquina virtual que se ejecute en ese ordenador sí puede serlo. Esto facilita que, una vez creada la máquina virtual en un ordenador, puedes copiar o mover el archivo contenedor que se crea y ejecutarla en cualquier otro ordenador, aunque tenga un hardware totalmente diferente.

Este tipo de máquinas virtuales, conocidas como máquinas virtuales de sistema, son las más conocidas por la mayoría de usuarios ya que son ideales para virtualizar sistemas operativos completos, pero no el único tipo de máquinas virtuales que existe. También existen las máquinas virtuales de procesos, que son más habituales en servidores y en las que únicamente se virtualizan determinados procesos o servicios y no el sistema operativo completo.



## → ¿Para qué sirven las máquinas virtuales?

Existen diversos usos para las máquinas virtuales, pero ya que permiten emular casi cualquier sistema operativo estándar (Windows, GNU/Linux, MacOS, Android, etc.), y dado que se ejecutan en una capa de software diferente y totalmente aislada, uno de los usos más frecuentes es el de probar diferentes sistemas operativos, programas o configuraciones con total seguridad para tu ordenador real ya que, si algo falla en la máquina virtual, este fallo no afectará en absoluto al ordenador que la ejecuta.

De ese modo, si por ejemplo, sospechas que un archivo que te han enviado pudiera estar infectado por un virus o software malicioso, puedes ejecutarlo en una máquina virtual para comprobar su fiabilidad. Si no sucede nada, podrás usarlo en tu equipo. En caso contrario, infectará solo a la máquina virtual y tu equipo se mantendrá inmune al ataque.

Esto puede ser muy útil para aprender cómo funcionan las redes informáticas y ahorrarte muchos euros ya que no necesitarás tener varios ordenadores reales para probar la seguridad de tus redes. Sin dejar de lado la tecnología de redes, llegamos a una de las principales utilidades para las máquinas virtuales de procesos, de las que ya te hemos hablado antes, y es que es más que probable que, sin saberlo, estás usando máquinas virtuales a diario cada vez que usas tu almacenamiento en la nube o navegas por Internet.

Las máquinas virtuales de procesos se ejecutan en un mismo servidor para ejecutar varias instancias de la misma aplicación de forma separada. De ese modo, si una de ellas falla, no afectará al funcionamiento del resto.

Esta solución es habitual en los servicios de alojamiento web y almacenamiento en la nube en las que se optimizan los costes de los servidores ejecutando una máquina virtual para alojar los archivos de usuarios distintos. De ese modo, en lugar de necesitar un servidor para cada uno de los usuarios, todos comparten el mismo hardware dividido en varias máquinas virtuales aisladas entre sí.



# CONFIGURAR ORACLE VM VIRTUALBOX

Para la creación de una máquina virtual, previamente debemos de haber instalado Virtualbox en nuestra máquina física.

Véase el vídeo:

<https://www.youtube.com/watch?v=UfCkZZQ8N4k&t=27s>



Abrimos la aplicación Oracle VM Virtual Box.



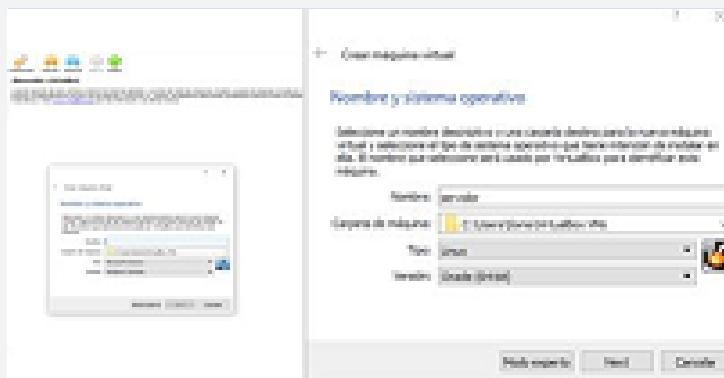
## Panel de configuración

Como se observa en la imagen, en la parte izquierda, aparecen las máquinas virtuales que hemos configurado. Para realizar una configuración nueva.



## Realizar una configuración nueva

Menú superior del panel, hacemos clic en nueva y automáticamente se nos mostrará una ventana.



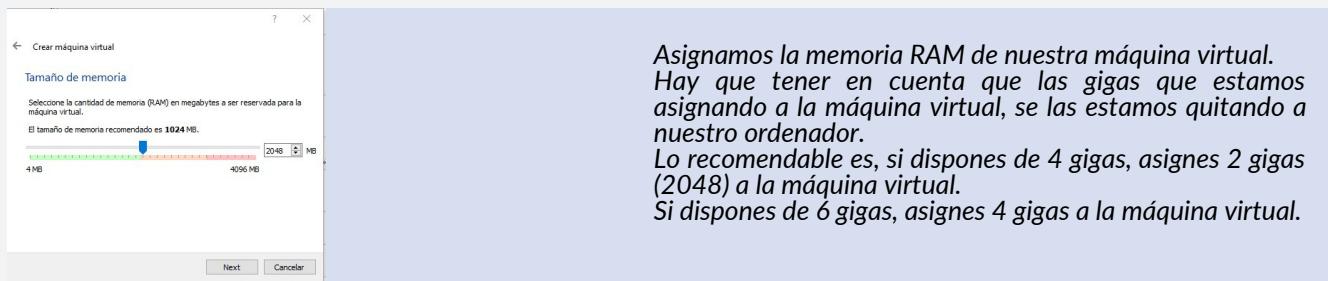
## Panel de configuración

Debemos indicar:

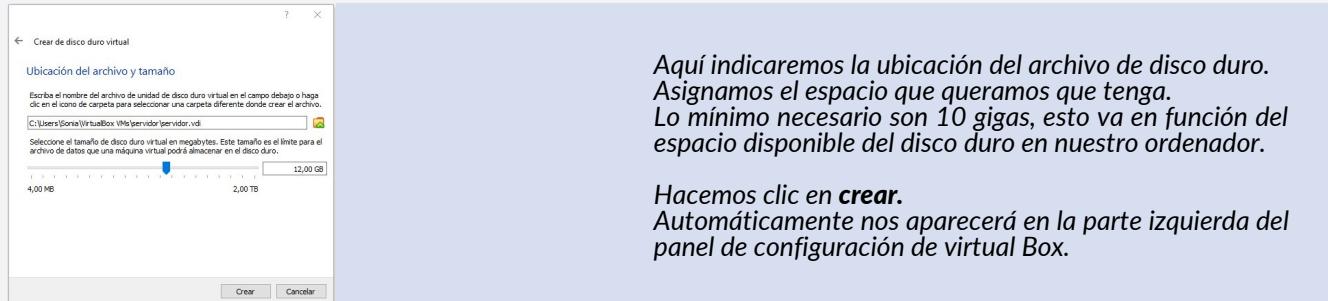
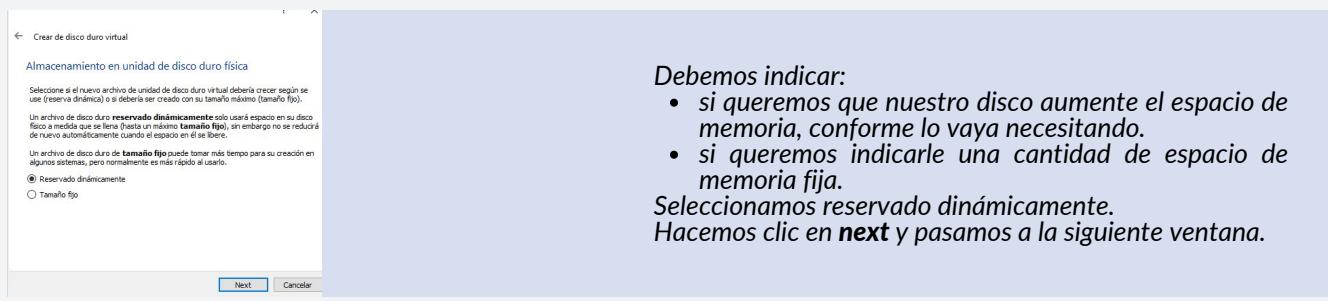
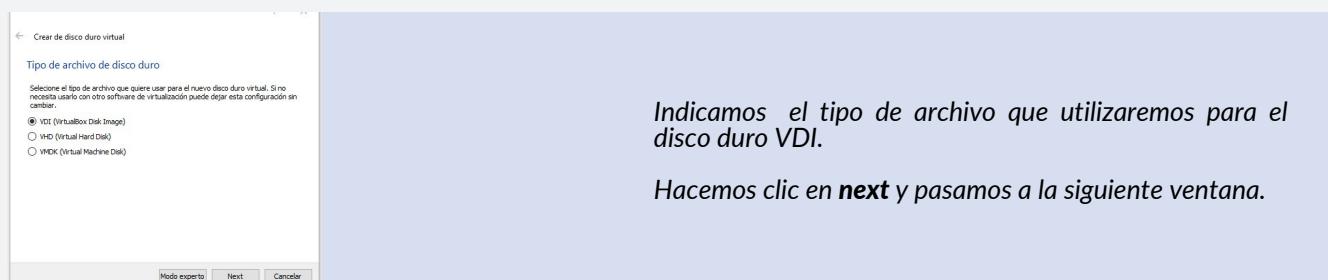
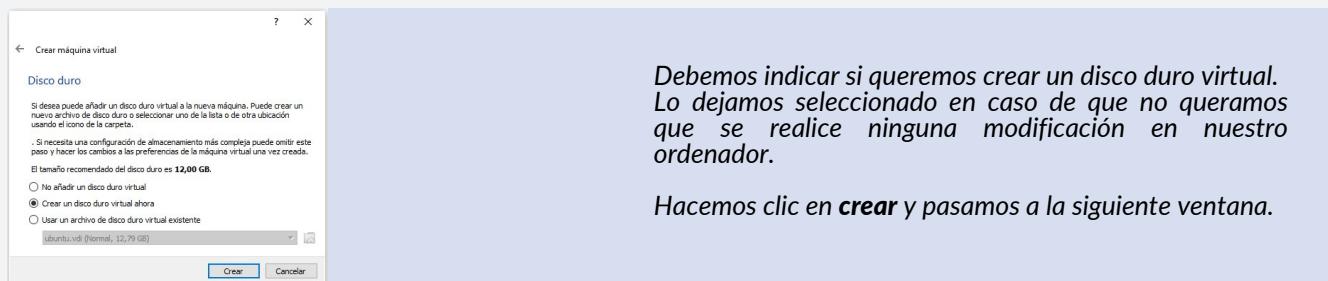
- **Nombre:** indicar el nombre con el que se identificará nuestra máquina virtual.
- **Carpetas de máquina:** indica la ubicación en nuestro ordenador.
- **Tipo:** indica el sistema operativo con el que vamos a trabajar. En nuestro caso Linux.
- **Versión:** indica la versión del sistema operativo que vamos a utilizar, 32 o 64 bits.

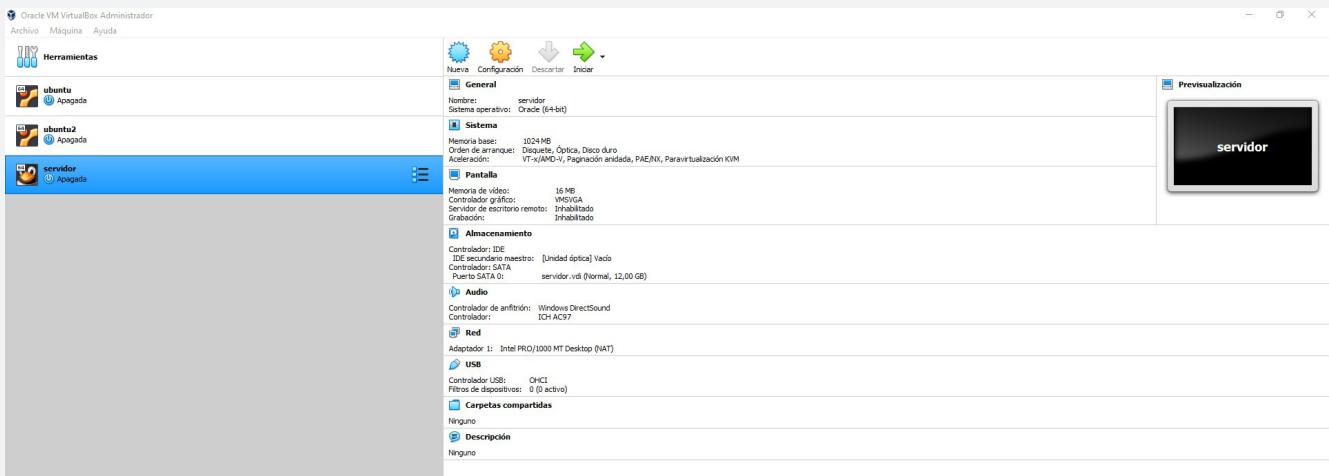
Se recomienda la de 64 bits.

Una vez rellenados todos los campos, hacemos clic en next y pasamos a la siguiente ventana.



Una vez configura la memoria RAM, hacemos clic en **next** y pasamos a la siguiente ventana.





Como se muestra en la imagen, la máquina virtual que hemos creado aparece seleccionada en azul.

Ahora debemos comprobar que en la configuración de nuestro ordenador (CPU) este habilitada la virtualización.

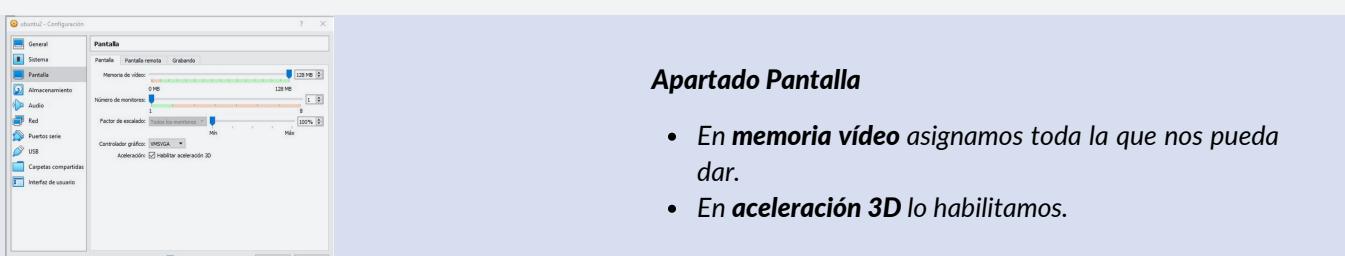
Para su configuración véase el vídeo:

<https://www.youtube.com/watch?v=B1oCkcOHXPA>



Pantalla , de nuestro ordenador, que nos indica si la virtualización de nuestro equipo está activada.

Volvemos al panel de configuración de Virtualbox, y hacemos clic en **configuración** y se nos abrirá una nueva ventana .

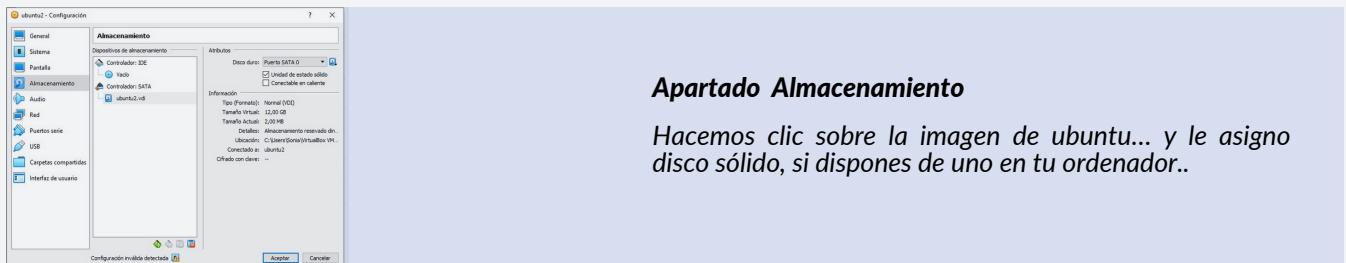


### Apartado Pantalla

- En **memoria vídeo** asignamos toda la que nos pueda dar.
- En **aceleración 3D** lo habilitamos.

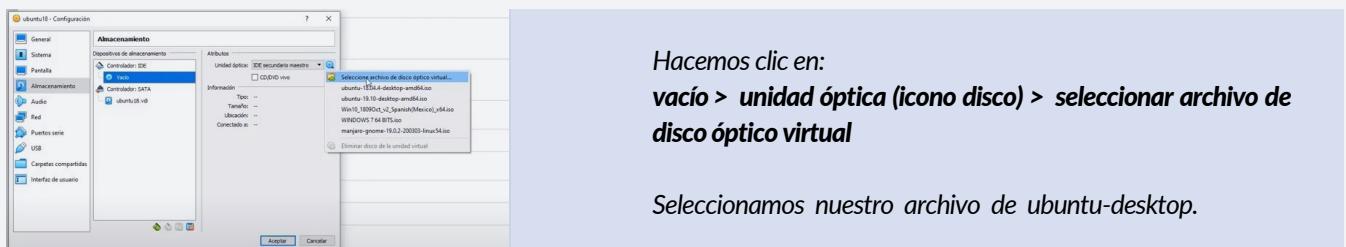


En este apartado estamos indicando que la máquina física este en conexión con la máquina virtual.



### Apartado Almacenamiento

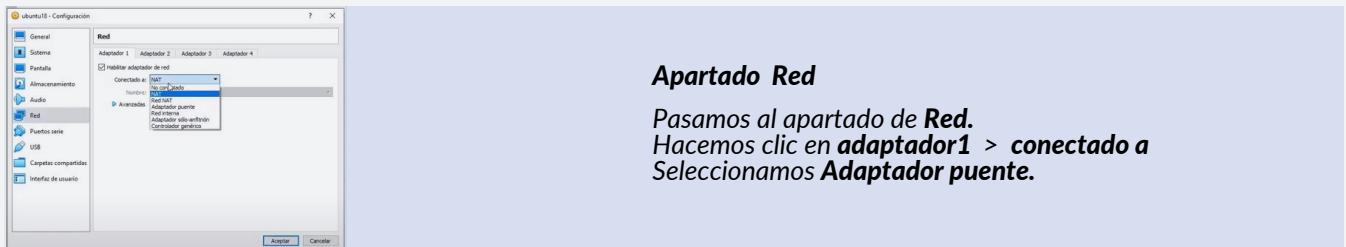
Hacemos clic sobre la imagen de ubuntu... y le asigno disco sólido, si dispones de uno en tu ordenador..



Hacemos clic en:

**vacio > unidad óptica (ícono disco) > seleccionar archivo de disco óptico virtual**

Seleccionamos nuestro archivo de ubuntu-desktop.



### Apartado Red

Pasamos al apartado de Red.

Hacemos clic en **adaptador1 > conectado a**  
Seleccionamos **Adaptador puente**.

En este apartado estamos indicando que la máquina física este en conexión con la máquina virtual.  
Hacemos clic en aceptar para que se guarde toda la configuración.



# CONFIGURAR UBUNTU-DESKTOP

Primero debemos iniciar nuestra máquina virtual, creada anteriormente.

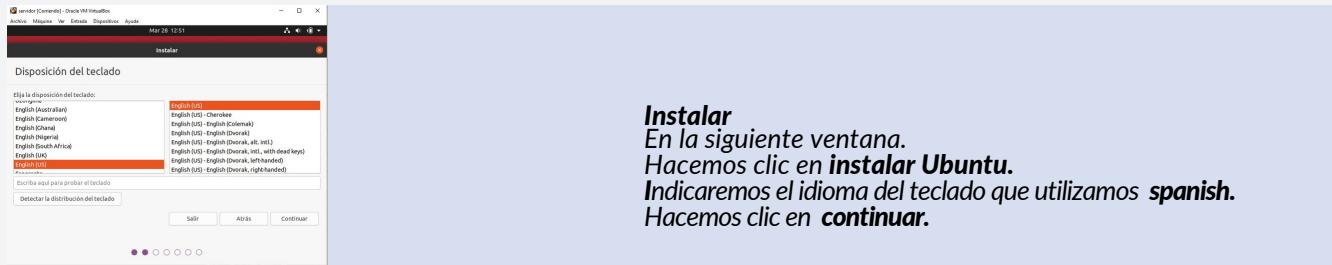
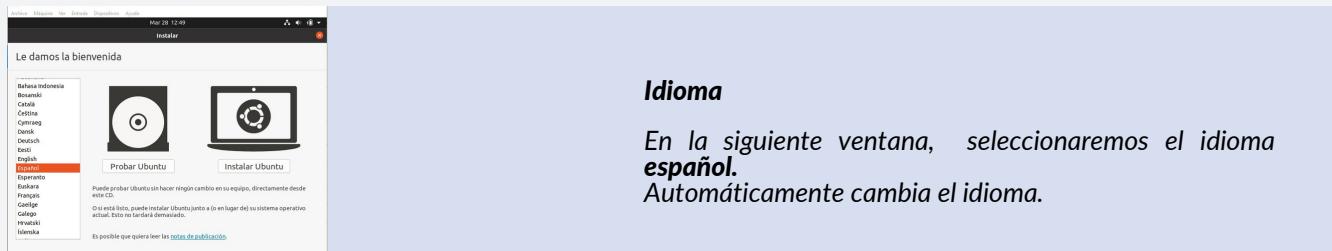
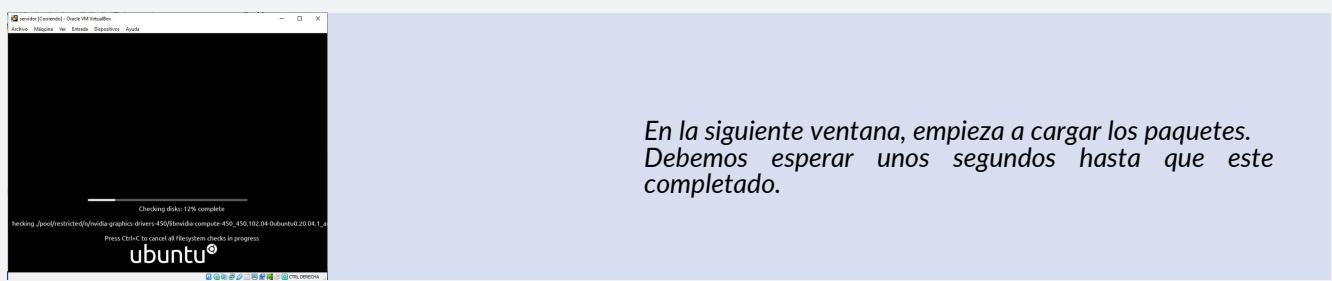
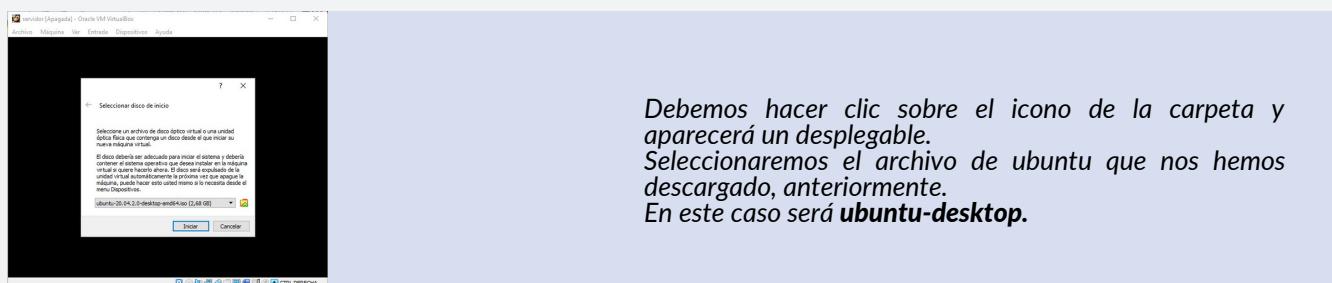
En el menú que aparece en la parte superior del panel de control.

Hacemos clic en **Iniciar**.

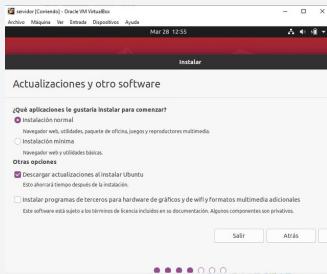
Tener en cuenta que iniciamos la máquina que hay seleccionada de color azul.

También podemos observar, en la parte superior izquierda, una previsualización de las acciones que se están realizando con nuestro servidor virtual.

Una vez iniciada, nos aparecerá una nueva ventana. debemos hacer clic sobre el icono de la carpeta y aparecerá un desplegable.



A partir de ahora nuestra ventana se hace más pequeña, podemos desplazarla con el ratón.

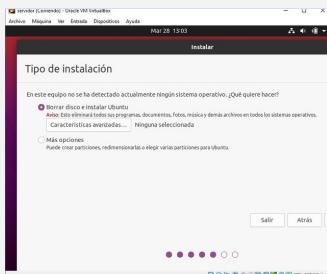


**Indicaremos las aplicaciones que queremos instalar.**

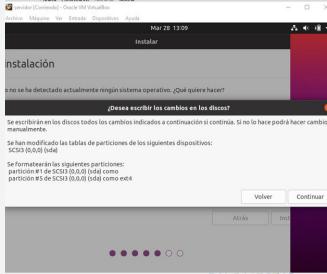
- si queremos la instalación normal, que es bastante completa.
- si queremos que se descarguen actualizaciones.

Esta selección dependerá del espacio que le hemos dado anteriormente a nuestro disco duro virtual.

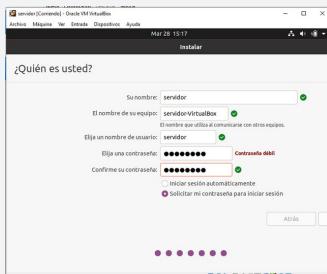
**Seleccionamos la normal y descargar actualizaciones.**  
Debemos esperar unos segundos mientras se cargan.



En la siguiente ventana, le indicaremos si queremos borrar el disco virtual e instalar ubuntu.  
También podemos configurar las particiones del disco duro.  
En nuestro caso borrar e instalar.  
Hacemos clic en **instalar ahora**.



**Mensaje**  
Nos muestra un mensaje como este.  
Hacemos clic en **continuar**.  
Nos muestra un mapa con nuestra ubicación, el país.  
Hacemos clic en **continuar**.

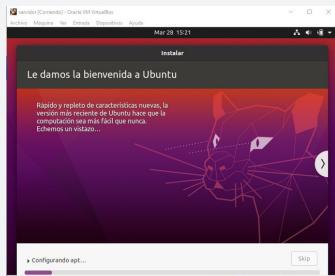


Lo recomendable es solicitar contraseña.  
Hacemos clic en **continuar**.

## Datos Registro

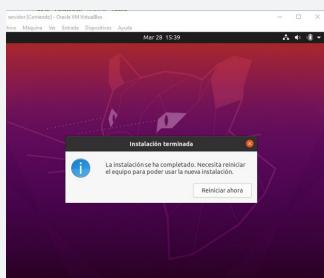
En la siguiente ventana deberemos:

- identificarnos.
- nombrar nuestro equipo para identificarnos con otros equipos.
- indicar un nombre de usuario para acceder.
- la contraseña.
- repetir contraseña.
- si queremos que se inicie de forma automática o solicitar contraseña.



### Instalando

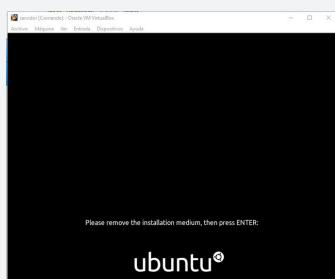
*Debemos esperar a que se instale.*



### Reiniciar

*Una vez completado el proceso de instalación nos pedirá reiniciar la máquina.*

*Hacemos clic en **reiniciar ahora**.*



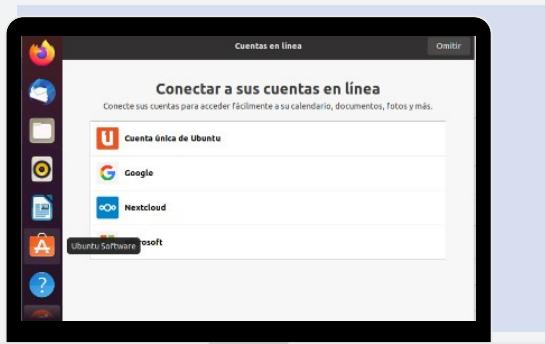
### Reiniciado

*Presionamos enter*



### Loguear

*Una vez se reinicie, nos pedirá loguearnos para acceder.*



### Escritorio Ubuntu

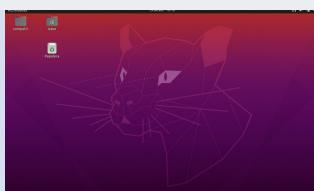
*Hemos finalizado la configuración de ubuntu en la máquina virtual de Virtual Box..*

# INSTALAR CLIENTE SSH

Lo primero que debemos hacer es instalar el cliente ssh (putty) en nuestro ordenador.  
Véase el vídeo:

<https://www.youtube.com/watch?v=FRQJN9RgYLQ>

Desde el panel de control de nuestra máquina virtual, la iniciamos.  
Accedemos con nuestro usuario y contraseña.



Accedemos a nuestro escritorio.



## Consola

Abrir la terminal.

Como se muestra en la imagen, accedemos con el usuario que hemos creado en nuestra configuración.

Aquí describo algunos de los comando más utilizados:

- **clear:** para limpiar la pantalla de la terminal.
- **pwd:** nos indica la ubicación en la que nos encontramos.
- **cd:** para desplazarnos entre directorios.
- **ls:** para generar listados.
- **ls -l:** genera listados en un formato diferente al anterior comando.
- **sudo:** comando se superusuario, se emplea para realizar acciones como administrador del sistema.
- **apt:** para instalar aplicaciones.

La ejecución de los scripts la realizaremos con **enter**.

Antes de realizar cualquier instalación o cuando accedemos al sistema es recomendable actualizar los repositorios.

```
root@lsavo-VirtualBox:~# apt upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Se instalarán los siguientes paquetes NUEVOS:
linux-headers-5.8.0-49-generic linux-hwe-5.8-headers-5.8.0-49
linux-image-5.8.0-49-generic linux-modules-5.8.0-49-generic
linux-modules-extra-5.8.0-49-generic
Se actualizan los siguientes paquetes:
libseccomp2 linux-generic-hwe-20.04 linux-headers-generic-hwe-20.04
linux-image-generic-hwe-20.04
4 actualizados, 5 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 78,9 MB de archivos.
Se utilizarán 389 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

Actualizar repositorios  
**apt upgrade**

```
root@lsavo-VirtualBox:~# apt-get update
Obj 1: http://dl.google.com/linux/chrome/deb stable InRelease
Obj 2: http://es.archive.ubuntu.com/ubuntu focal InRelease
Des 3: http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Des 4: http://security.ubuntu.com/ubuntu focal-security InRelease [309 kB]
Des 5: http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Des 6: http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [914 kB]
Des 7: http://es.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [452 kB]
Des 8: http://es.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [561 kB]
Des 9: http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [758 kB]
Descargados 3.008 kB en 8s (392 kB/s)
Leyendo lista de paquetes... Hecho
root@lsavo-VirtualBox:~#
```

Actualizar las aplicaciones  
**apt-get update**

Para realizar acciones de instalación, actualización, etc.... Son necesarios permisos de administrador. Estos permisos están asignados al usuario root, por lo que deberemos cambiar de usuario.

```
isavo@isavo-VirtualBox:~$ sudo -i
[sudo] contraseña para isavo:
root@isavo-VirtualBox:~#
```

Nos solicita la contraseña.  
En caso de no ser usuario root, las peticiones deberán ser realizadas con el comando **sudo**

```
isavo@isavo-VirtualBox:~$ sudo su root
root@isavo-VirtualBox:/home/isavo#
```

Para cambiar de usuario escribiremos

**sudo su root o**  
**sudo -s o**  
**sudo -i**

La diferencia entre uno u otro es que según el que uses te posiciona en un directorio u otro.

```
isavo@isavo:~$ sudo apt-get update
[sudo] password for isavo:
Hit:1 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
isavo@isavo:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  liburap0 curses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeesphere rssh ssh-askpass
The following NEW packages will be installed:
  liburap0 curses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 5 newly installed, 0 to remove and 53 not upgraded.
Need to get 583 kB of archives.
After this operation, 5,422 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Buscamos **openssh-server**:

**apt search openssh-server**

Una vez que nos hayamos asegurado que nos aparece en la lista y de la sintaxis de su nombre, procederemos a realizar la instalación.

- como usuario root  
**apt-get install openssh-server openssh-client**
- con nuestro usuario  
**sudo apt-get install openssh-server**  
**sudo apt-get install openssh-client**

Como se muestra en la imagen, antes de realizar la instalación, nos muestra los paquetes que se van a instalar, el espacio que van a ocupar y nos pregunta si queremos instalarlo.

A lo que indicaremos **S o Y**

Aparentemente no nos mostrará nada, por lo que debemos comprobar que se haya instalado.

```
isavo@isavo-VirtualBox:~$ apt search openssh-server
Ordenando... Hecho
Buscar en todo el texto... Hecho
openSSH-server/focal-updates,focal-security,now 1:8.2p1-4ubuntu0.2 amd64 [instalado]
  servidor de terminal seguro (SSH) para acceder de forma segura desde máquinas remotas
```

Comprobar actualización  
**apt search openssh-server**

Nos muestra el servidor ssh instalado.

Instalamos el paquete **net-tools**, contiene una colección de programas que forman la base de trabajo en red, en Linux.

```
root@isavo-VirtualBox:~# apt search net-tools
Ordenando... Hecho
Buscar en todo el texto... Hecho
atm-tools/focal 1:2.5.1-4 amd64
  Base programs for ATM in Linux, the net-tools for ATM

ddnet-tools/focal 12.9.2-1ubuntu1 amd64
  Tools for DDNet

hobbit-plugins/focal,focal 20191218 all
  complementos para el monitor de red Xymon

iproute2/focal,now 5.5.0-1ubuntu1 amd64 [instalado, automático]
  Herramientas de redes y control de tráfico

net-tools/focal,now 1.60+git20180626.aebd88e-1ubuntu1 amd64 [instalado]
  Conjunto de herramientas de red NET-3

root@isavo-VirtualBox:~#
```

Instalar net-tools

**apt install net-tools**

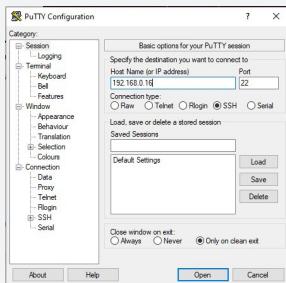
Comprobamos que se haya instalado.

Una vez completada la instalación, procederemos a realizar una **conexión ssh**. Para realizar dicha conexión necesitamos saber la IP de nuestra máquina virtual.

```
lsavo@isavo-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.16 netmask 255.255.255.0 broadcast 192.168.0.255
        inet6 fe80::f73b:61ff:fea15 prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:08:6d:9c txqueuelen 1000 (Ethernet)
            RX packets 2734 bytes 3366871 (3.3 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1041 bytes 101354 (101.3 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

**Saber nuestra IP  
ifconfig**

La IP que necesitamos es:  
**192.168.0.16**



Abrimos el programa putty  
Realizamos la conexión ssh indicando:  

- **IP:** 192.168.0.16
- **Puerto:** 22 normalmente es el que se utiliza.

Hacemos clic en **open**.



En la siguiente ventana.  
Hacemos clic en **sí**.

Y se nos abrirá la consola.

Nos pedirá loguearnos, indicando usuario y contraseña que configuramos en la instalación de Ubuntu-desktop.

```
isavo@isavo-VirtualBox: ~
login as: isavo
isavo@192.168.0.16's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-48-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

67 actualizaciones se pueden instalar inmediatamente.
36 de estas actualizaciones son una actualización de seguridad.
Para ver estas actualizaciones adicionales ejecute: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

isavo@isavo-VirtualBox:~$
```

Si la conexión ha tenido éxito nos dará la bienvenida.

# INSTALAR SERVIDOR APACHE

Desde la terminal de nuestra máquina virtual.

Las peticiones se pueden realizar como usuario root o con nuestro usuario, por lo que, para realizar ciertas acciones no tendremos privilegios de administrador, habrá que iniciar los scripts con el comando sudo y nos solicitará la contraseña para su ejecución.

```
isavo@isavo-VirtualBox:~$ apt search apache2
Ordenando resultados por relevancia... Hecho
Búsquedas en todo el texto... Hecho
apache2/focal-updates,focal-security,now 2.4.41-4ubuntu3.1 amd64 [instalado]
  servidor HTTP Apache
apache2-bin/focal-updates,focal-security,now 2.4.41-4ubuntu3.1 amd64 [instalado, automático]
  Servidor HTTP Apache (binarios y otros archivos binarios)
apache2-data/focal-updates,focal-security,now 2.4.41-4ubuntu3.1 all [instalado, automático]
  Servidor HTTP Apache (archivos comunes)
apache2-dev/focal-updates,focal-security,now 2.4.41-4ubuntu3.1 amd64
  Servidor HTTP Apache (cabeceras de desarrollo)
apache2-doc/focal-updates,focal-security,focal-security,now 2.4.41-4ubuntu3.1 all
  Servidor HTTP Apache (documentación en Instancia)
apache2-ssl-dev/focal-updates,focal-security,now 2.4.41-4ubuntu3.1 amd64
  Apache HTTP Server (mod_ssl desarrollo headers)
apache2-suexec-custom/focal-updates,focal-security,now 2.4.41-4ubuntu3.1 amd64
  Apache HTTP Server configurable suexec program for mod_suexec
apache2-suexec-fattice/focal-updates,focal-security,now 2.4.41-4ubuntu3.1 amd64
  Apache HTTP Server standard suexec program for mod_suexec
apache2-utils/focal-updates,focal-security,now 2.4.41-4ubuntu3.1 amd64 [instalado, automático]
  servidor HTTP Apache (programas de utilidad para servidores web)
corstd/focal,focal 0.6.0-4 all
  streaming applet for Ogg formats
```

Antes de realizar la instalación, igual que en la instalación de openSSH-server, nos mostrará la información.

Indicamos S o Y

Empezará la instalación.

Debemos comprobar que se ha instalado

**apt search apache2**

Nos logueamos como usuario root

**sudo -i**

Hacemos una búsqueda del servidor apache2

**apt search apache2 | grep HTTP**

Procedemos a instalar los paquetes de distribución de apache2

**apt-get install apache2**

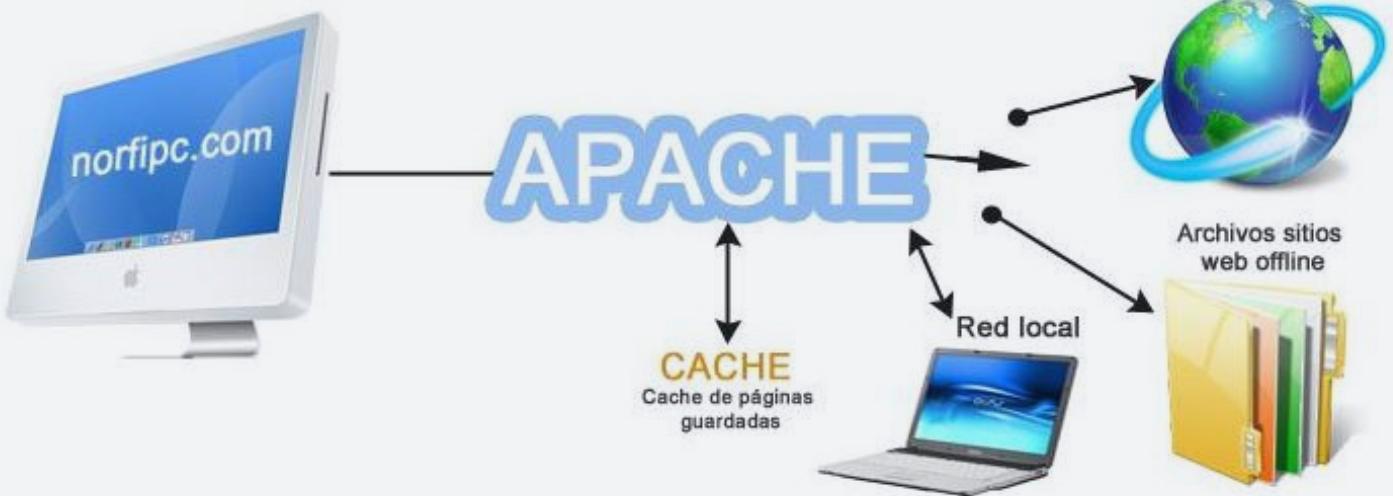
Podemos ver la versión de apache que se ha instalado.

```
isavo@isavo-VirtualBox:~$ apache2 -v
Server version: Apache/2.4.41 (Ubuntu)
Server built:   2020-08-12T19:46:17
isavo@isavo-VirtualBox:~$
```

Versión de apache instalada

**apache2 -v**

## Funcionamiento del servidor Apache



```
root@isavo-VirtualBox:~# systemctl start apache2
root@isavo-VirtualBox:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-04-13 10:06:14 CEST; 45min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 657 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 760 (apache2)
   Tasks: 6 (limit: 2221)
  Memory: 14.9M
    CGroup: /system.slice/apache2.service
            └─ 760 /usr/sbin/apache2 -k start
              ├─ 775 /usr/sbin/apache2 -k start
              ├─ 777 /usr/sbin/apache2 -k start
              ├─ 778 /usr/sbin/apache2 -k start
              ├─ 779 /usr/sbin/apache2 -k start
              └─ 780 /usr/sbin/apache2 -k start

abr 13 10:06:10 isavo-VirtualBox systemd[1]: Starting The Apache HTTP Server...
abr 13 10:06:13 isavo-VirtualBox apachectl[701]: AH00558: apache2: Could not re-read configuration
abr 13 10:06:14 isavo-VirtualBox systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

### Iniciamos nuestro servidor web apache **systemctl start apache2**

Aparentemente nuestra terminal no muestra nada, por lo que debemos comprobar que el estado del servidor sea activo.

### **systemctl status apache2**

Como podemos observar el servidor apache está en funcionamiento.  
Para continuar o salir q

Ahora comprobamos que nuestro servidor web funciona.

Abrimos el navegador de nuestro escritorio, **Firefox**

Hacemos una búsqueda por la IP: 192.168.0.16 de nuestro servidor virtual o localhost.



Búsqueda por IP:

**192.168.0.16**



Búsqueda por servidor local

**localhost**

Como podemos observar se muestra el archivo index.html instalado por defecto y almacenado en el directorio de publicación de nuestro servidor web.

La ruta en la que se encuentra el directorio de publicación: **/var/www/html/index.html**.

Los scripts que vamos a utilizar para el servicio de apache son:

**Iniciar el servicio**

**systemctl start apache2**

**Detener el servicio**

**systemctl stop apache2**

**Reiniciar el servidor apache**

**systemctl reload apache2**

**Comprobar su estado**

**systemctl status apache2**

# DIRECTORIOS

## → ¿Qué es un directorio?

En informática, un directorio es un contenedor virtual en el que se almacenan una agrupación de archivos informáticos y otros subdirectorios, atendiendo a su contenido, a su propósito o a cualquier criterio que decida el usuario. Técnicamente, el directorio almacena información acerca de los archivos que contiene: como los atributos de los archivos o dónde se encuentran físicamente en el dispositivo de almacenamiento. A partir de Windows 3.0, también se los denomina carpetas de archivos.

En el entorno gráfico de los sistemas operativos modernos, el directorio se denomina metafóricamente carpeta y de hecho se representa con un ícono con esta figura. La cual se asocia con el ambiente administrativo de cualquier oficina, donde la carpeta de cartón encierra las hojas de papel (representando a los archivos de datos) de un expediente.

En DOS y sus sucesores Windows y OS/2, la sintaxis de una ruta (o path en inglés), indica una jerarquía de directorios, donde el primer elemento puede ser bien la letra indicativa de cualquier Unidad Lógica (disco) en el sistema, o la barra invertida o backslash (\), designando al "**directorío raíz**" de la unidad lógica actual.

Si la ruta comienza directamente en el nombre de un directorio es una ruta relativa desde el directorio de trabajo actual.

Por ejemplo, en la ruta "C:\AB\AGENDA", "C:" es la unidad lógica (el disco), "AB" un directorio y "AGENDA" un subdirectorio o un archivo. "AB\AGENDA" podría designar el mismo elemento si el directorio actual es "C:\".

En los sistemas operativos de tipo **UNIX**, el directorio se organiza a partir del directorio raíz "/", el cual contiene archivos y otros directorios. Esos directorios pueden contener archivos y directorios y así sucesivamente. Esto puede organizarse por el sistema en una estructura llamada árbol.

## → ¿Para qué sirven?

El término informática define un objeto de sistema de archivos, tratándose de un concepto. Los archivos son un conjunto de caracteres relacionados que cuentan con un nombre, formato y extensión, y son una parte fundamental de la informática para guardar y acceder a la información. Los directorios son la estructura para contener dichos archivos de una forma organizada que facilite su acceso y almacenamiento categorizado, por lo que son muy importantes en la informática en general.

## → ¿Qué es el directorio raíz?

El directorio raíz es el directorio más alto o principal dentro de una jerarquía de carpetas o particiones. Este directorio es muy importante en los sistemas operativos y en las páginas web, pues es donde se guarda toda la información de los mismos. Es habitual que el directorio raíz esté protegido para evitar que terceros no autorizados o hackers puedan acceder a él, puesto que tendrían acceso a toda la información.

## → ¿Qué es un árbol de directorios?

*La organización de los directorios y carpetas se realiza de forma similar a la de un árbol (en referencia a cómo las ramas de los árboles van ampliándose desde el tronco). La representación gráfica en forma de árbol parte del directorio raíz (tronco) y va expandiéndose en carpetas y subcarpetas (ramas).*

*Para apreciar visualmente esta estructura en forma de árbol se puede utilizar el explorador de archivos de Windows/Linux o el Finder de Mac Os y navegar por las distintas carpetas y subcarpetas del sistema operativo. Se apreciará que desde una carpeta raíz se abre una estructura de carpetas, que a su vez contienen otras subcarpetas, etc.*

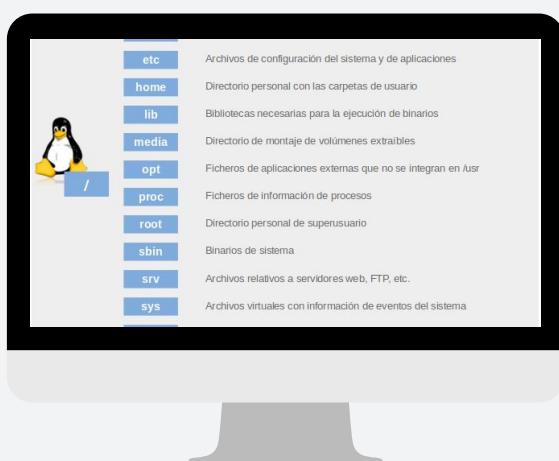
## → ¿Cómo funcionan los directorios?

*Los directorios hacen las veces de inicio o raíz de una estructura de archivos y carpetas, bien de un sistema operativo, de una página web o de una aplicación o sistema de almacenamiento.*

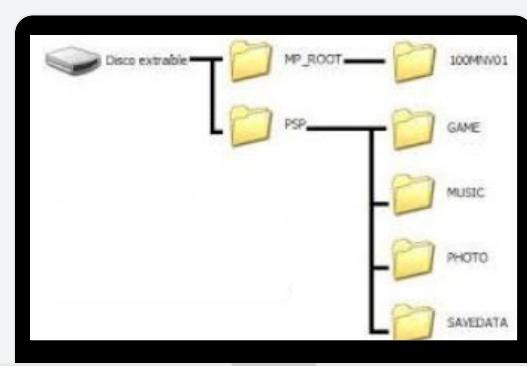
*El directorio raíz de un sistema operativo se encuentra en el disco o dispositivo de almacenamiento que se utilice y a partir de ahí se accede al resto de contenido del mismo. En el caso de Windows, el directorio raíz suele encontrarse bajo el nombre de la unidad principal C:/ y a partir de ahí comienza la estructura de carpetas y archivos del sistema (Windows, Usuarios, Archivos de Programas, etc.).*

*En una página web, el directorio raíz es la base de acceso público al sitio web, donde se situarán todos los archivos y carpetas que contienen la estructura, diseño y contenido del sitio web (index.php, index.html y la estructura de carpetas).*

*Muchas veces se nombra el directorio raíz haciendo referencia a un programa o aplicación, en este caso se está hablando de la carpeta principal donde está alojado o almacenado dicho programa (por ejemplo, en Windows se pueden encontrar los directorios raíz de las aplicaciones instaladas en C:Archivos de programaNombre del programa).*



Directorio Linux



Jerarquización de carpetas

# PERMISOS DE USUARIO

Cuando queramos realizar una acción como crear un fichero o un directorio, lo primero que debemos tener en cuenta son los permisos o privilegios que tienen los usuarios, o tenemos como usuario sobre la carpeta.

Esto significa que se pueden crear diversos tipos de protección a los archivos y directorios.

Dicha protección se realiza con la determinación de permisos de archivos y directorios.

En Linux los directorios se tratan como archivos.

Existen tres tipos de permisos:

- **r:** leer un archivo y listar un directorio,
- **w:** escribir o modificar el contenido de un archivo y crear nuevos directorios o ficheros en su interior,
- **x:** ejecutar el fichero si es un programa y cambiar al directorio, es decir, poder entrar en dicho directorio.

Estos permisos pueden ser otorgados a tres niveles o roles:

- **propietario:** es el usuario que ha creado el fichero.
- **grupo:** es el conjunto de usuarios que pertenecen al mismo grupo que el propietario.
- **otros:** son el resto de usuarios del sistema.

El **propietario** tendrá permisos para realizar diferentes acciones sobre el archivo que ha creado.

El **usuario root**, el superusuario, es la cuenta del administrador del sistema que posee todos los privilegios para la gestión del sistema Linux, puede otorgar permisos a usuarios individuales.

El **comando sudo**, es una utilidad de sistemas operativos tipo Unix, que permite a un usuario ejecutar programas con privilegios de seguridad que solo están asignados al usuario root.

## → Descripción de permisos:

Para listar un directorio y sus permisos:

Deberemos desplazarnos a dicho directorio, es decir debemos entrar en dicho directorio.

ej: **cd ruta\_absoluta\_del\_directorio**

Una vez dentro, listamos los permisos

**ls -l**

También podemos hacer, independientemente del directorio que nos encontramos.

ej: **ls -l ruta\_absoluta\_del\_directorio**

Para poder establecer los permisos sobre un determinado archivo o directorio, se utiliza una cadena de **10 caracteres**.

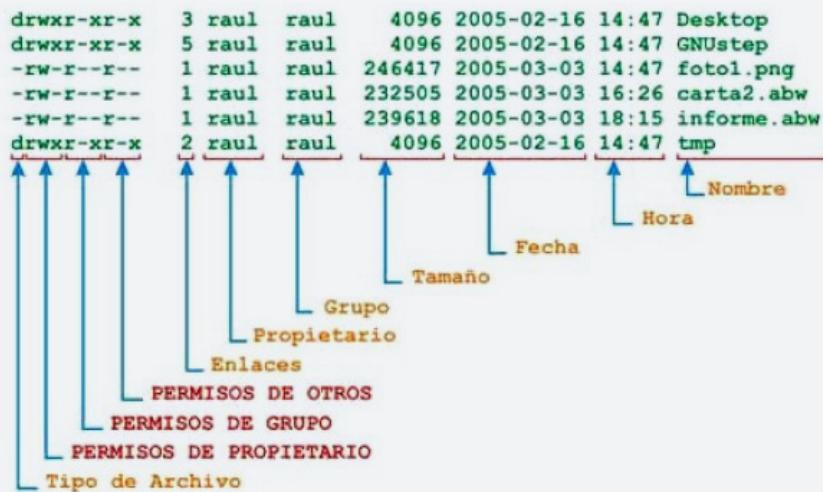
**SIGN UP**

**Username.....**

**Password.....**

**Password again.....**

**Email.....**



En la imagen se muestra la forma utilizada para mostrar los permisos, recibe el nombre de cadena.

Esta cadena se divide en 4 partes para indicar:

- **Tipo de archivo:** es el tipo de archivo con el que estamos trabajando.
- **Propietario:** indica los permisos del propietario.
- **Grupo:** indica los permisos establecidos al grupo principal al que pertenece el propietario.
- **Otros permisos:** indican los permisos del resto de usuarios del sistema.

Los 9 caracteres restantes se dividen en grupos de 3, cada uno representa al propietario, el grupo y otros permisos.

Sólo se pueden cambiar los permisos de:  
un archivo o un directorio, si se es su propietario, o el superusuario root.

El primer carácter de la cadena indica el tipo de archivo: si se trata de "d" representa un directorio, si es "-" equivale a un archivo. Sin embargo, otros caracteres pueden aparecer para indicar otros tipos de archivos, como se muestra en la siguiente tabla:

- d: directorio
- b: archivo de bloque
- c: archivo especial de caracteres
- p: canal
- s: socket
- -: archivo normal

Para asignar los permisos de usuario se utiliza la forma octal, entre otras.

Octal	Lectura	Escritura	Ejecución
0	-	-	-
1	-	-	x
2	-	w	-
3	-	w	x
4	r	-	-
5	r	-	x
6	r	w	-
7	r	w	x

# CREAR DIRECTORIO WEBS

El paso siguiente será crear nuestro directorio público, en la configuración del servidor web, donde se crearán diferentes directorios según las características requeridas por el ejercicio y nuestras webs.

Iniciamos nuestra máquina virtual, abrimos una terminal, nos logueamos como usuario root, iniciamos el servidor apache y comprobamos que esté activo.

```
sudo -i Nos solicitará la contraseña
Actualizar el sistema
apt upgrade
Iniciar apache
systemctl start apache2
Comprobar el estado de apache
systemctl status apache2
Para continuar
q
```

```
savo@isavo-VirtualBox:~$ sudo -i
[sudo] contraseña para isavo:
root@isavo-VirtualBox:~# systemctl start apache2
root@isavo-VirtualBox:~# systemctl status apache2
apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-04-13 11:26:13 CEST; 9s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 10982 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 10982 (apache2)
    Tasks: 6 (limit: 2221)
   Memory: 11.0M
      CPU: 0.000 CPU(s) since start
     CGroup: /system.slice/apache2.service
             ├─10986 /usr/sbin/apache2 -k start
             ├─10987 /usr/sbin/apache2 -k start
             ├─10988 /usr/sbin/apache2 -k start
             ├─10989 /usr/sbin/apache2 -k start
             ├─10990 /usr/sbin/apache2 -k start
             └─10991 /usr/sbin/apache2 -k start

br 13 11:26:13 isavo-VirtualBox systemd[1]: Starting The Apache HTTP Server...
br 13 11:26:13 isavo-VirtualBox apachectl[10985]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for port 80
br 13 11:26:13 isavo-VirtualBox systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (FND)
```

Listamos el directorio de publicación var/www/ del servidor web.

Los documentos que se encuentren dentro de la carpeta raíz, son accesibles vía web.

Listar directorio www  
**ls /var/www/**

Nos muestra el directorio html, dónde se encuentra la web creada por defecto al instalar apache2.

```
root@isavo-VirtualBox:~# ls /var/www/
html
root@isavo-VirtualBox:~#
```

Acceder al directorio html

**ls /var/www/html**

Creamos el directorio/ carpeta webs

**mkdir /var/www/webs/**

Comprobamos que se haya creado.

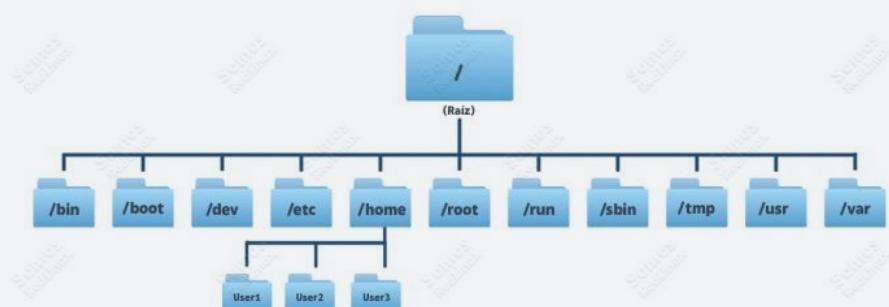
**ls /var/www/**

```
root@isavo-VirtualBox:~# ls /var/www/
html webs
root@isavo-VirtualBox:~#
```

Listamos el directorio para ver los permisos.

**ls -l /var/www/webs/**

```
root@isavo-VirtualBox:~# ls -l /var/www/webs/
total 12
drwxrwxrwx 7 root root 4096 abr 2 20:59 web
drwxr-xr-x 4 root root 4096 abr 2 20:59 .
root@isavo-VirtualBox:~#
```



# CREAR HOSTS VIRTUALES

Un virtualhost es la configuración que permite a Apache tener más de un sitio web en una misma máquina.

Apache es un servidor basado en módulos o componentes que son independientes, por lo que podemos configurarlos independientemente.

Una de estas configuraciones son los virtualhost, que permiten asignar distintos sitios web a través de dominios a una misma dirección IP.

Una de las formas para definir los virtualhost, y la más común, que nos proporciona Apache es la configuración por nombre. Es decir, basándonos en el nombre que viene en la url de la petición.

Esto sucede gracias a que en el protocolo HTTP 1.1 el nombre del servidor es una cabecera de carácter obligatorio. Con lo cual, el servidor web puede distinguir entre los virtual host que tiene disponibles.

Lo más importante de este tipo de configuración es que para que un virtualhost por nombre pueda funcionar debe existir en las DNS.

Crearemos un Virtualhost llamado [www.miweb.com](http://www.miweb.com), con un alias **miweb.com** dentro de nuestro directorio de publicación **/var/www/webs/**.

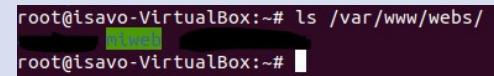
Lo primero que debemos hacer es iniciar la máquina virtual, loguearnos como usuario root, actualizar el sistema, iniciar apache y comprobar su estado.

Creamos el directorio miweb dentro de nuestro directorio de publicación.

```
mkdir /var/www/webs/miweb/
```

Listamos el directorio webs para comprobar que se haya creado  

```
ls /var/www/webs/
```



```
root@isavo-VirtualBox:~# ls /var/www/webs/
miweb
```

Creamos el archivo **index.html**

```
touch /var/www/webs/miweb/index.html
```

Comprobamos que se haya creado

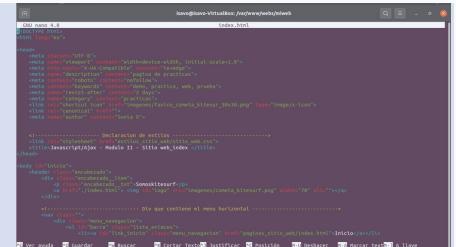
```
ls -l /var/www/webs/miweb/
```



```
@isavo-VirtualBox:~# ls /var/www/webs/miweb
index.html
```

Editamos el archivo para crear el código html de nuestra web

```
nano /var/www/webs/miweb/index.html
```



```
isavo@isavo-VirtualBox:~/var/www/webs/miweb
```

```
index.html
```

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Sitio web de ejemplo</title>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-REDACTED" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-A6A8wAq/SiQDyySDSf2+wxtUEjpc4c3GU805K2Z8xMvElCh/k/TcLKA1f7aw...
```

Para guardar lo editado en el archivo: **ctrl o**

Preguntará si queremos modificar el nombre, yo no lo modifco **Enter**.

Para salir de la edición: **ctrl x**

He creado una carpeta compartida para subir la web.

En el siguiente paso configuraremos los host virtuales.

Comprobar la configuración del directorio de publicación creado anteriormente.

**tree /var/www/**

Nos muestra

```
ot@isavo-VirtualBox:~# tree /var/www/
/var/www/
+-- html
|   '-- index.html
+-- webs
    +-- blue
        +-- img
            +-- deportivas.html
            +-- generales.html
            +-- lector_rss.html
            +-- mapas.html
            +-- tiempo.html
            +-- estilos_sitio_web
                +-- carrusell.css
                +-- contacto_sitio_web.css
                +-- encabezado.css
                +-- enlaces_sitio_web.css
                +-- externa.css
                +-- formulario_registro2.css
                +-- galeria_fotos.css
                +-- nav_horizontal.css
                +-- noticias_sitio_web.css
                +-- pie_pagina.css
                +-- rest.css
                +-- sitio_web.css
            +-- imagenes
                +-- cometa_kitesurf.png
                +-- degradado_1920x600.png
                +-- encabezado_nubes_1500x500.png
                +-- favico_cometa_kitesur_30x30.png
                +-- flecha_drcn_500x500.png
                +-- fondo-playa2000x1900.jpg
                +-- imagen_noticia_campeonato.oliva_2020.jpg
            +-- video
                +-- acariciar-ola-1200x569.jpg
                +-- andando-atardecer.jpg
                +-- beach_cometas_1920.jpg
                +-- campeonato_kitesurf.jpg
                +-- cometa-atardecer.jpg
                +-- cometas_fondo_volando.jpg
                +-- cometas_kite_mar.jpg
                +-- extreme_1280.jpg
                +-- extreme2_1280.jpg
                +-- extreme3_1280.jpg
                +-- haciendo_kitesurf.jpg
                +-- kitesurf_atardecer.jpg
                +-- kitesurfer_1280.jpg
                +-- kitesurfing2_1280.jpg
                +-- kite-surfing-salto1.jpg
                +-- kiting_1280.jpg
                +-- salto-360.jpg
                +-- sliderUnclo_1900x900.jpg
                +-- sobre_ola.JPG
                +-- surfing.JPG
                +-- Kitesurfing - media.mp4
                +-- logo_acceso_azul2_250x250.png
                +-- logo-acceso.jpg
            +-- playas
                +-- amanecer.JPG
                +-- arena.JPG
                +-- fondo-playa2000x1900.jpg
                +-- fondo-playa-agua.jpg
                +-- nature-and-landscapes-sky-ocean-beaches-thumbnail.jpg
                +-- playa1.JPG
                +-- playa2.JPG
                +-- playa3.JPG
                +-- puesta-de-sol.jpg
                +-- sea-.jpg
                +-- pngegg.png
                +-- video-2012-04-05-14-22-32.mp4
            +-- index.html
            +-- js
                +-- enlaces_video.js
                +-- jquery-1.11.1.js
                +-- jquery-3.5.1.js
                +-- jquery.easing-1.3.js
                +-- jquery.gomap-1.3.2.min.js
                +-- validacion_registro.js
            +-- montserrat [allfont.es].eot
            +-- logo_acceso_azul_web
                +-- contacto.html
                +-- enlaces_sitio_web.html
                +-- formulario_registro2.html
                +-- galeria_fotos.html
                +-- index.html
                +-- noticias_sitio_web.html
    0 directories, 76 files
ot@isavo-VirtualBox:~#
```

Nos desplazamos al directorio dónde se encuentran los archivos de configuración de apache2, en la ruta **etc/apache2/** y podemos ver la distribución de los diferentes directorios y archivos que hay.

```
root@isavo-VirtualBox:/etc/apache2# ls
apache2.conf  conf-available  conf-enabled  envvars  magic  mods-available  mods-enabled  ports.conf  sites-available
root@isavo-VirtualBox:/etc/apache2#
```

Directorio de configuración de apache2  
 cd /  
 cd etc/apache2  
 ls

Los que aparecen de color blanco son archivos de configuración, y los de color azul, son directorios.

De todo lo que nos muestra, los que nos interesan son:

- **Archivo apache2.conf**, es el archivo de configuración principal del servidor apache.
- **Directorio sites-available**, contiene los archivos de configuración de los diferentes hosts virtuales creados. Aquí configuraremos las directivas o características de cada uno de los host virtuales.
- **Directorio sites-enabled**, contiene los archivos de configuración de los host virtuales activos.

Anteriormente hemos creado el directorio donde se aloja el archivo de nuestra web.

La ruta es: **var /www/webs/miweb**

Ahora crearemos el archivo de configuración correspondiente para que apunte a nuestro index.html.

Podemos acceder al directorio sites-available y crear el archivo de configuración.

Accedemos al directorio sites-available  
 cd /etc/apache2/sites-available  
 Listamos el directorio de conf. de apache  
 ls  
 Creamos el archivo miweb.conf  
 touch miweb.conf

```
root@isavo-VirtualBox:/etc/apache2# cd sites-available/
root@isavo-VirtualBox:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf
root@isavo-VirtualBox:/etc/apache2/sites-available#
```

Podemos guardar el archivo en el directorio o carpeta correspondiente, a la vez que lo creamos.  
 Indicando la ruta absoluta de la carpeta/directorio y el nombre que le demos al archivo.  
**touch etc/apache2/sites-available/miweb.conf**

También podemos copiar el archivo que se crea por defecto para configurar un virtualhost.

Para copiar el archivo **000-default.conf**, situados en el directorio **etc/apache2/sites-available**

Copiar el archivo 000-default.conf  
**cp 000-default.conf miweb.conf**

```
root@isavo-VirtualBox:/etc/apache2/sites-available# cp 000-default.conf miweb.co
nf
root@isavo-VirtualBox:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf  miweb.conf
root@isavo-VirtualBox:/etc/apache2/sites-available#
```

Ahora editaremos el archivo para definir las directivas de nuestra configuración. Para ello usaremos el editor de texto GNU nano.

```

GNU nano 4.8
root@isavo-VirtualBox: ~
/etc/apache2/sites-available/miweb.conf
<VirtualHost *:80>

    ServerName www.miweb.com
    ServerAlias miweb.com
    ServerAdmin soporte@miweb.com
    DocumentRoot /var/www/webs/miweb
    DirectoryIndex index.html

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

    <Directory /var/www/webs/miweb>
        Options Indexes FollowSymLinks MultiViews
        Require all granted
    </Directory>
</VirtualHost>

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

```

*Editar archivo*

**nano miweb.conf**

Vamos a configurar la directiva **VirtualHost**

- <**virtualHost** \* :80>
  - \*, indicamos que es para cualquier IP.
  - :80, indicamos el puerto.
- **ServerAdmin**, indicamos la dirección de correo a la que deben dirigirse los clientes, cuando el servidor apache lance un error.
- **DocumentRoot**, indicamos la ruta donde está ubicado la estructura de nuestra web, directorio de publicación.
- **ServerName**, indicamos el nombre del dominio o el nombre del host virtual.
- **ServerAlias**, indicamos nombres alternativos para un host.
- **DirectoryIndex**, indicamos que si se accede tanto por el dominio como por el alias, sin indicar la ruta absoluta, siempre mostrará el index.html.
- **Directory**, indicamos las características que se aplican sobre un directorio, usando la directiva Options.
  - **Options**, directiva donde indicamos las características de un directorio. Con + o - se activan o desactivan.
    - **+Indexes**, permite mostrar la estructura de carpetas del directorio. Por seguridad deberían estar desactivados todos.
    - **+FollowSymLinks**, permite seguir los enlaces simbólicos en el directorio. Accesos directos.
    - **+MultiViews**, permite multiviews.
  - **Require all granted**, indica el acceso a todos los usuarios.
- <**virtualHost** >

Para guardar cambios en el editor.

**Ctrl o**

Tras guardar nos indicará si queremos cambiar el nombre del archivo.

En caso contrario

**Enter o Intro**

Para salir de la edición.

**Ctrl x**

El siguiente paso es activar el virtualhost que hemos configurado en el servidor.

En el **directorio sites-available** se encuentran los archivos de configuración de cada uno de los virtual host que vayamos creando. Mientras que el **directorio sites-enabled** contiene los archivos de configuración de cada uno de los host virtuales activos o habilitados que hemos creado en sites-available.

Es decir, para que un host virtual este activo, es decir, para que sea accesible en el servidor, debemos pasar los archivos de configuración del directorio sites-available al directorio sites-enabled con un enlace simbólico.

Listamos el directorio sites-enabled  
**ls /etc/apache2/sites-enabled/**

```
root@isavo-VirtualBox:/etc/apache2/sites-available# ls /etc/apache2/sites-enabled/
000-default.conf
root@isavo-VirtualBox:/etc/apache2/sites-available#
```

Nos muestra el enlace por defecto del localhost de apache.

Creamos el enlace simbólico ejecutando el siguiente script.

Crear un enlace simbólico  
**a2ensite miweb.conf**

```
root@isavo-VirtualBox:/etc/apache2/sites-available# a2ensite miweb.conf
Enabling site miweb.
To activate the new configuration, you need to run:
systemctl reload apache2
```

El servidor apache nos pedirá reiniciar, pero antes podemos realizar una comprobación de sintaxis.

```
root@isavo-VirtualBox:~# apache2ctl 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
root@isavo-VirtualBox:~#
```

Comprobar sintaxis  
**apache2ctl configtest**

La primera línea indica que el **ServerName** no está configurado a nivel global.

En la siguiente línea nos indica que no hay errores de sintaxis,  
**Syntax OK**.

Ahora podemos reiniciar con seguridad el sistema/apache.

Reiniciar el sistema/apache2  
**systemctl reload apache2**

Comprobamos que se haya creado el archivo  
**ls etc/apache2/sites-enabled**

```
savo-VirtualBox:/etc/apache2/sites-available# systemctl reload apache2
savo-VirtualBox:/etc/apache2/sites-available# ls /etc/apache2/sites-enabled/
000-default.conf  miweb.conf
savo-VirtualBox:/etc/apache2/sites-available#
```

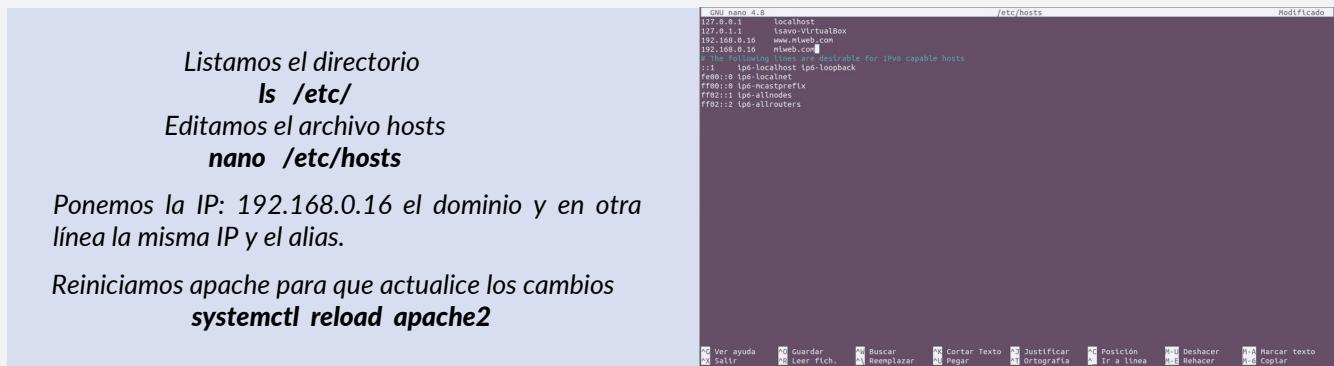
Podemos comprobar los dominios que están activos.

Comprobación de dominios activos  
**apachectl -S**

Realizamos las comprobaciones.

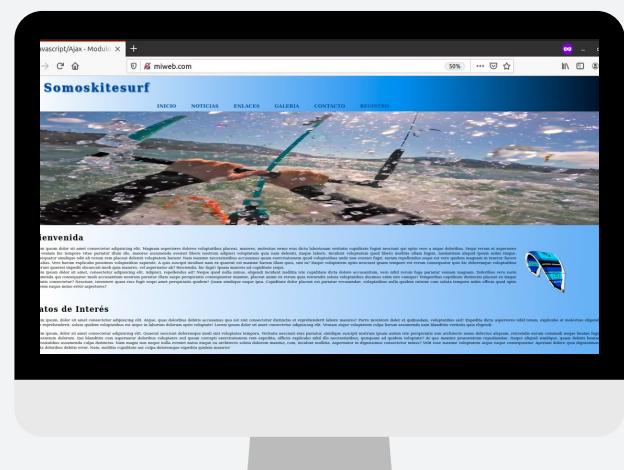
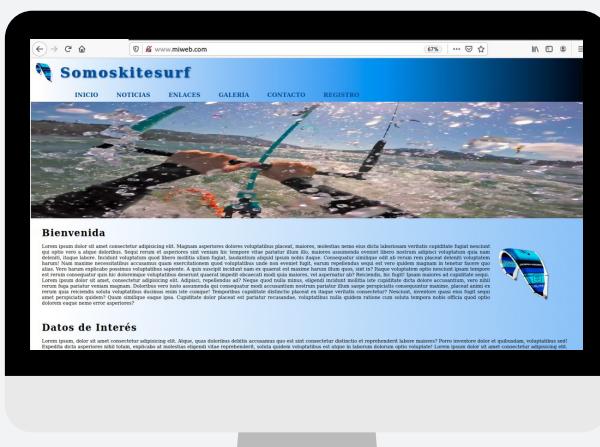
```
root@isavo-VirtualBox:/etc/apache2/sites-available#
root@isavo-VirtualBox:/etc/apache2/sites-available# apachectl -S
AH00558: apache2: could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
VirtualHost configuration:
*:80
    default server 127.0.1.1 (/etc/apache2/sites-enabled/000-default.conf:1)
    port 80 namevhost 127.0.1.1 (/etc/apache2/sites-enabled/000-default.conf:1)
    port 80 namevhost www.miweb.com (/etc/apache2/sites-enabled/miweb.conf:1)
        alias miweb.com
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex watchdog-callback: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: namevhost id=33
Group: namevhost id=33
root@isavo-VirtualBox:/etc/apache2/sites-available#
```

Para que un host virtual por nombre y sus alias funcionen debemos crear sus nombres en las DNS. Si no existe en las DNS no se podrán resolver las direcciones IP y por lo tanto no se podrán atender las peticiones. Para dicha resolución añado los dominios en el fichero hosts que se encuentra en el directorio /etc/



La segunda comprobación es introducir en el navegador los dominios y debe mostrar nuestro index.html, que es en lo que consiste un host virtual por nombre.

El archivo hosts, al poner el dominio comprueba que para ese dominio en concreto, lo envía a la IP del servidor y el servidor detecta a través de que dominio viene y le muestra la página.



# CREAR CARPETA PRIVADA

En un servidor web podemos restringir los accesos a recursos a través del uso de usuarios y contraseñas, los cuales podemos organizar en grupos.

Los mecanismos de autenticación que nos proporciona Apache no incorporan ningún tipo de encriptación, sólo alguna codificación. Por lo que entre cliente y servidor las credenciales viajan a través de la red en texto plano, la encriptación solo se consigue si empleamos mecanismos como **ssl** o **tls**.

Sin embargo el fichero que utiliza apache para almacenar los usuarios y credenciales, sí que está encriptado.

Lo recomendable es el uso de una combinación entre **ssl** o **tls** y autenticación básica, puesto que la información almacenada en el disco del servidor tiene una encriptación más robusta.

## → Autenticación Básica

Este sistema consiste en que al intentar acceder a un recurso restringido, apache devuelve un mensaje 401, **Authentication Required**.

Pueden pasar dos cosas:

- Que el navegador tenga las credenciales guardadas y las envía directamente.
- Que el navegador no tenga las credenciales y las solicita al usuario.

Para poder implementarlo necesitaremos un archivo donde se irán almacenando los usuarios y contraseñas que vayamos creando. Será un archivo de texto que se almacenara en el directorio de configuración de apache, /etc/apache2/, pero no debe ser accesible desde la web.

En el siguiente paso definiremos un **recurso protegido**, un directorio /var/www/webs/acceso/, que deberá estar asociado a un virtual host, **www.acceso.com (acceso.conf)**, y lo activaremos. Finalmente, crearemos una web, dentro de dicho directorio protegido, con acceso por autenticación.

Primero iniciamos la máquina virtual, abrimos una terminal, nos logueamos con usuario root, actualizamos los paquetes del sistema, arrancamos apache y comprobamos que su estado sea activo.

```
sudo -i
apt upgrade
systemctl start apache2
systemctl status apache2
```

```
root@isavo-VirtualBox:~# ls /var/www/webs/
acceso [miweb]
root@isavo-VirtualBox:~#
```

Creamos directorio protegido, en nuestro directorio de publicación webs  
**mkdir /var/www/webs/acceso/**

Comprobamos que se haya creado

**ls /var/www/webs/**

Creamos el archivo index.html

**touch /var/www/webs/acceso/index.html**

En mi caso lo he subido a la carpeta compartida.

Ahora crearemos en el directorio de configuración de apache el archivo donde se almacenarán los usuarios y sus credenciales.

*Creamos el archivo y el primer usuario*

```
htpasswd -c /etc/apache2/passwords.txt empleado1
```

El comando **htpasswd**, utilizado para crear el archivo de usuarios y contraseñas de un servidor.

Al crear el usuario nos pedirá su contraseña correspondiente, repetir contraseña y así con cada uno que vayamos añadiendo.

En los siguientes scripts no ponemos **-c** para que no se generé otro archivo.

*Creamos más usuarios*

```
htpasswd /etc/apache2/passwords empleado2  
htpasswd /etc/apache2/passwords empleado3  
htpasswd /etc/apache2/passwords encargado  
htpasswd /etc/apache2/passwords administracion  
htpasswd /etc/apache2/passwords soporte
```

*Las credenciales serán 123456 para todos.*

Comprobamos que se haya creado el archivo **passwords**  

```
ls /etc/apache2/
```

```
root@isavo-VirtualBox:~# ls /etc/apache2/  
apache2.conf  conf-enabled  mods-available  ports.conf      ssl  
apachegroups  envvars      mods-enabled   sites-available  
conf-available  magic       passwords     sites-enabled  
root@isavo-VirtualBox:~#
```

Leemos el archivo para ver los usuarios creados con el comando **cat**.

Leemos el archivo **passwords**  
**cat /etc/apache2/passwords**

```
root@isavo-VirtualBox:~# htpasswd -c /etc/apache2/passwords empleado1  
New password:  
Re-type new password:  
Adding password for user empleado1  
root@isavo-VirtualBox:~# ls /etc/apache2/passwords  
/etc/apache2/passwords  
root@isavo-VirtualBox:~# cat /etc/apache2/passwords  
empleado1:$apr1$H3FVDbfe$pxdhM1Tj/4qtKkFx4EvpT1  
root@isavo-VirtualBox:~# htpasswd /etc/apache2/passwords empleado2  
New password:  
Re-type new password:  
Adding password for user empleado2  
root@isavo-VirtualBox:~# htpasswd /etc/apache2/passwords empleado3  
New password:  
Re-type new password:  
Adding password for user empleado3  
root@isavo-VirtualBox:~# htpasswd /etc/apache2/passwords administracion  
New password:  
Re-type new password:  
Adding password for user administracion  
root@isavo-VirtualBox:~# htpasswd /etc/apache2/passwords encargado  
New password:  
Re-type new password:  
Adding password for user encargado  
root@isavo-VirtualBox:~# htpasswd /etc/apache2/passwords soporte  
New password:  
Re-type new password:  
Adding password for user soporte  
root@isavo-VirtualBox:~# cat /etc/apache2/passwords  
empleado1:$apr1$H3FVDbfe$pxdhM1Tj/4qtKkFx4EvpT1  
empleado2:$apr1$ln3bYT4U$lv51.Supy65V8f7MIRdeN0  
empleado3:$apr1$R1ldftvUSPINA/tvQJTkau.10X6f0  
administracion:$apr1$vhkU26qS4/93cnI26p7rw.Z.1A0ep/  
encargado:$apr1$SxFB5uXw$Bhw8hj46f0Z.jLS.M8tgr1  
soporte:$apr1$yLIjeLv$8TFRDUIte$pWSDcNpZEHln1  
root@isavo-VirtualBox:~#
```

El siguiente paso será crear el host virtual, definir sus características con las directivas de la autenticación básica por usuarios y habilitarlo.

# Host virtual www.acceso.com

## → Autenticación básica por usuarios

```
Creamos el host virtual www.acceso.com, copiando el archivo miweb.conf.
cp /etc/apache2/sites-available/miweb.conf /etc/apache2/sites-available/acceso.conf
Comprobamos que se haya creado listando el directorio sites-available.
ls /var/www/webs/
Creamos el archivo index.html
touch /var/www/webs/acceso/index.html
```

The screenshot shows a terminal window titled 'Terminal' with the command line interface. It displays the following steps:

```
root@isavo-VirtualBox:~# ls /etc/apache2/
apache2.conf  conf-available  conf-enabled  envvars  magic  mods-available  mods-enabled  ports.conf  sites-available  sites-enabled
root@isavo-VirtualBox:~# cp /etc/apache2/sites-available/miweb.conf /etc/apache2/sites-available/accesobasico.conf
root@isavo-VirtualBox:~# ls /etc/apache2/sites-available/
000-default.conf  default-ssl.conf  miweb.conf
root@isavo-VirtualBox:~# nano /etc/apache2/sites-available/accesobasico.conf
```

Lo editamos para definir las directivas correspondientes a la Autenticación básica de usuarios.

```
GNU nano 4.8                               /etc/apache2/sites-available/acceso.conf
<VirtualHost *:80>

    ServerName www.acceso.com
    ServerAdmin soporte@miweb.com
    DocumentRoot /var/www/webs/acceso
    DirectoryIndex index.html

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

    <Directory /var/www/webs/acceso>
        Options +Indexes +FollowSymLinks +MultiViews
    </Directory>
    <Directory /var/www/webs/acceso>
        AuthType Basic
        Require valid-user
        AuthBasicProvider file
        AuthUserFile "/etc/apache2/passwords"
        AuthName "Acceso Protegido"
    </Directory>
</VirtualHost>
```

Editamos el archivo

```
nano /etc/apache2/sites-available/acceso.conf
```

Directivas de Authentication Basic.

- **Directory:** indicamos la ruta del directorio para la que se solicitará autenticación de usuarios.
- **AuthType:** indicamos el tipo de autenticación.
- **Require:** indicamos que para acceder al recurso protegido requerimos la autenticación de un usuario válido.
- **AuthBasicProvider:** indicamos el proveedor para la autenticación básica, en nuestro caso un fichero.
- **AuthUserFile:** indicamos la ruta del fichero que contiene los usuarios y credenciales, entre comillas.
- **AuthName:** indicamos un texto para este directorio protegido, que se mostrará al solicitar la autenticación.

Guardamos cambios y salimos de la edición del archivo.  
**Ctrl o > Enter > Ctrl x**

Comprobamos errores de sintaxis.

**apachectl configtest**

Activamos el host virtual www.acceso.com

**a2ensite acceso.conf**

Comprobamos los dominios activos

**apachectl -S**

```
root@isavo-VirtualBox:~# apachectl -S
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1
bally to suppress this message
VirtualHost configuration:
*:443                  www.missl.com (/etc/apache2/sites-enabled/missl.conf:8)
*:80                   default server 127.0.1.1 (/etc/apache2/sites-enabled/000-default.conf:1)
port 80 namevhost 127.0.1.1 (/etc/apache2/sites-enabled/000-default.conf:1)
port 80 namevhost www.acceso.com (/etc/apache2/sites-enabled/acceso.conf:1)
port 80 namevhost www.missl.com (/etc/apache2/sites-enabled/missl.conf:1)
port 80 namevhost www.miweb.com (/etc/apache2/sites-enabled/miweb.conf:1)
alias miweb.com
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex default: dir="/var/run/apache2/" mechanism=default
Mutex mpm-accept: using_defaults
Mutex watchdog-callback: using_defaults
Mutex ssl-stapling-refresh: using_defaults
Mutex ssl-stapling: using_defaults
Mutex ssl-cache: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
root@isavo-VirtualBox:~#
```

Lo editamos para definir las directivas correspondientes a la Autenticación básica de usuarios.

Reiniciamos apache

**systemctl reload apache2**

Listamos el directorio sites-enabled para comprobar que aparece el archivo acceso.conf y se crea el enlace simbólico.

**ls /etc/apache2/sites-enabled/**

```
root@isavo-VirtualBox:~# apachectl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name,
bally to suppress this message
Syntax OK
root@isavo-VirtualBox:~# a2ensite acceso.conf
Enabling site acceso.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@isavo-VirtualBox:~# systemctl reload apache2
root@isavo-VirtualBox:~# ls /etc/apache2/sites-enabled/
000-default.conf  acceso.conf  miweb.conf
root@isavo-VirtualBox:~#
```

Creamos el dominio en el archivo hosts para la resolución de las DNS.

Editamos el archivo hosts  
**nano /etc/hosts**

Guardamos cambios y salimos de la edición del archivo.

**Ctrl o > Enter > Ctrl x**

```
root@isavo-VirtualBox: /etc
root@isavo-VirtualBox: /etc/hosts
GNU nano 4.8
127.0.0.1      localhost
127.0.1.1      isavo-VirtualBox
192.168.0.16   www.miweb.com
192.168.0.16   miweb.com
192.168.0.16   acceso.com
192.168.0.16   www.accesobasico.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
```

Creamos el dominio en el archivo hosts para la resolución de las DNS.

Comprobamos que el módulo auth\_basic este activo  
**a2enmod auth\_basic**

Reiniciamos apache para que actualice los cambios  
**systemctl reload apache2**

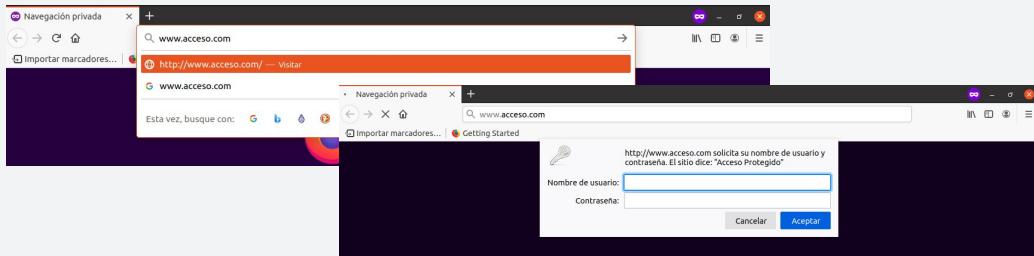
```
root@isavo-VirtualBox:/# a2enmod auth_basic
Considering dependency authn_core for auth_basic:
Module authn_core already enabled
Module auth_basic already enabled
root@isavo-VirtualBox:/#
```

l. Sep 15:53 ..
6. Sep 2015 bin -> usr/bin
19. Sep 09:31 boot
21. Sep 15:56 dev
19. Sep 09:32 etc
21. Sep 15:52 home
7 30. Sep 2015 lib ..

Ahora vamos al navegador FireFox y realizamos las comprobaciones. Abrimos una pestaña en modo incógnito e introducimos nuestro dominio.

→ [www.acceso.com](http://www.acceso.com)

Si intentamos acceder con un usuario que no existe en el archivo **passwords**, nos devuelve a la misma página. Lo mismo sucede si no introducimos las credenciales correctamente.



Ahora vamos a acceder con el usuario y la contraseña de un usuario que existe en el archivo **passwords**.

Name	Last modified	Size	Description
estilos_sitio_web/	2021-04-06 14:50	-	
imagenes/	2021-04-02 20:01	-	
indexhtml	2021-04-06 14:35	8.1K	

# Host virtual www.accesobasico.com

## → Autenticación básica por grupo de usuarios

La metodología de creación es la misma que en el caso de la configuración por usuarios.

Creamos el directorio privado en el directorio de publicación dentro del directorio acceso,

Creamos el directorio privado  
**mkdir /var/www/webs/acceso/privado/**

Creamos el archivo index.html  
**touch /var/www/webs/acceso/index.html**

```
root@isavo-VirtualBox:~# mkdir /var/www/webs/privado
root@isavo-VirtualBox:~# ls /var/www/webs
miweb  privado

root@isavo-VirtualBox:~# ls /var/www/webs/
acceso  miweb
root@isavo-VirtualBox:~# ls /var/www/webs/acceso
privado
root@isavo-VirtualBox:~# ls /var/www/webs/acceso/privado
archivos  formulario_registro2.html  index.html
root@isavo-VirtualBox:~#
```

Creamos y editamos un archivo con el grupo que podrá acceder al directorio privado.

Creamos y editamos el archivo  
**nano /etc/apache2/apache2groups**

```
GNU nano 4.8
privado: encargado administracion soporte
```

Escribimos el nombre del grupo seguido de dos puntos (:) y a continuación los usuarios que forman parte de ese grupo, los cuales deben aparecer en el archivo de usuarios y credenciales creado anteriormente.

Copiamos el archivo **acceso.conf** y creamos el host virtual **accesobasico.conf**.

Copiamos y editamos el archivo  
**cp /etc/apache2/sites-available/acceso.conf /etc/apache2/sites-available/accesobasico.conf**

Configuramos las directivas correspondientes a un grupo de usuarios

Las directivas, diferentes al caso anterior, que añadimos son:

- Require:** indicamos el grupo que tendrá acceso.
- AuthGroupFile:** indicamos la ubicación del fichero donde se ha creado el grupo entre comillas.

Comprobación de sintaxis  
**apachectl configtest**

Activamos el virtualhost **www.accesobasico.com**  
**a2ensite accesobasico.conf**

Reiniciamos el sistema/apache  
**systemctl reload apache2**

Comprobamos que se ha creado el enlace simbólico  
**ls /etc/apache2/sites-enabled/**

Editamos el archivo hosts  
**nano /etc/hosts/**

Comprobamos los dominios que están activos.  
**apachectl -S**

```
GNU nano 4.8
<VirtualHost *:80>
  ServerName www.accesobasico.com
  ServerAdmin soporte@miweb.com
  DocumentRoot /var/www/webs/acceso/privado
  DirectoryIndex index.html

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

  <Directory /var/www/webs/acceso/privado>
    Options +Indexes +FollowSymLinks +MultiViews
  </Directory>
  <Directory /var/www/webs/acceso/privado>
    AuthType Basic
    Require group privado
    AuthBasicProvider file
    AuthUserFile /etc/apache2/passwords
    AuthGroupFile /etc/apache2/apache2groups
    AuthName "Acceso Privado"
  </Directory>
</VirtualHost>
```

```
GNU nano 4.8
localhost          127.0.0.1
isavo-VirtualBox   127.0.1.1
www.miweb.com      192.168.0.16
miweb.com          192.168.0.16
acceso.com         192.168.0.16
www.accesobasico.com 192.168.0.16

# The following lines are desirable for IPv6 capable hosts
::1               ip6-localhost ip6-loopback
fe00::0           ip6-mcastprefix
ff00::0           ip6-allnodes
ff02::1           ip6-allrouters
ff02::2           ip6-allrouters
```

```
root@isavo-VirtualBox:/# nano /etc/apache2/sites-available/accesobasico.conf
root@isavo-VirtualBox:/# service apache2 restart
Job for apache2.service failed because the control process exited with error code.
See "systemctl status apache2.service" and "journalctl -xe" for details.
```

*Si reiniciamos el servicio de apache y nos indica que ha fallado por un error de sintaxis, significa que no debemos tener activado el módulo por grupos authz\_groupfile*

Por lo que debemos activar dicho módulo.  
**a2enmod authz\_groupfile**

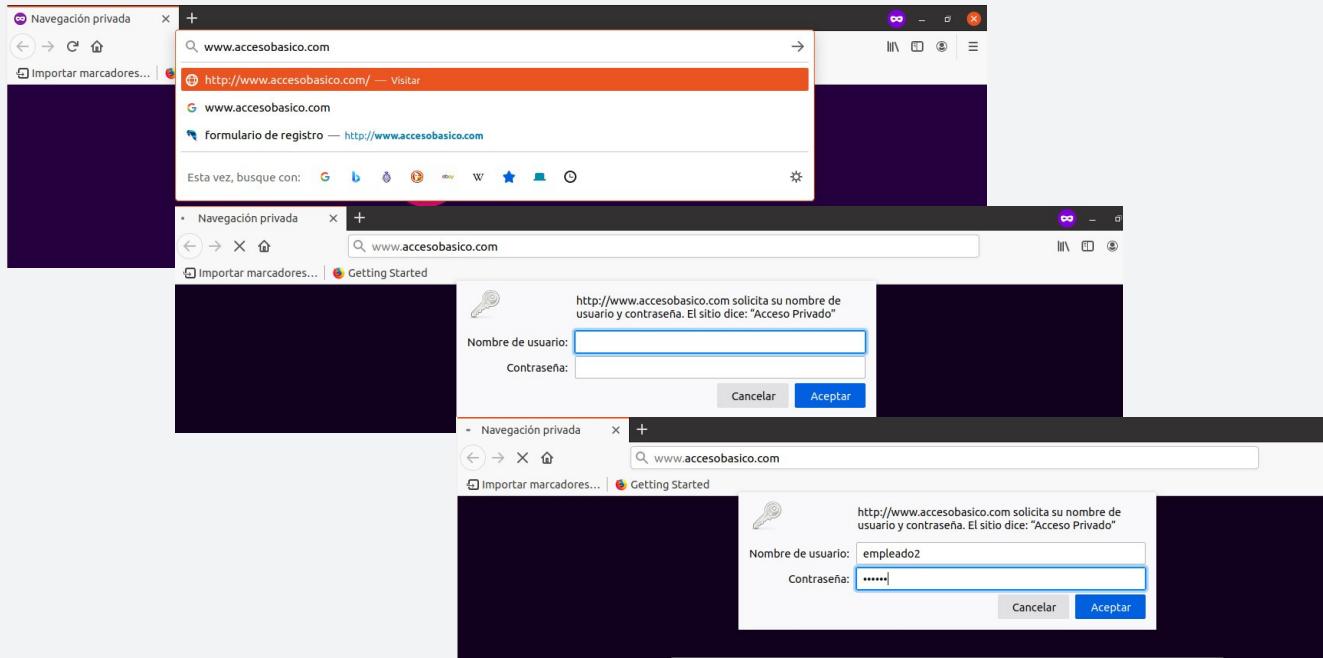
```
root@isavo-VirtualBox:/# a2enmod authz_groupfile
Considering dependency authz_core for authz_groupfile:
Module authz_core already enabled
Enabling module authz_groupfile.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@isavo-VirtualBox:/# █
```

Editamos el index.html para definir el esqueleto de la web.  
**nano /var/www/webs/acceso/privado/index.html**

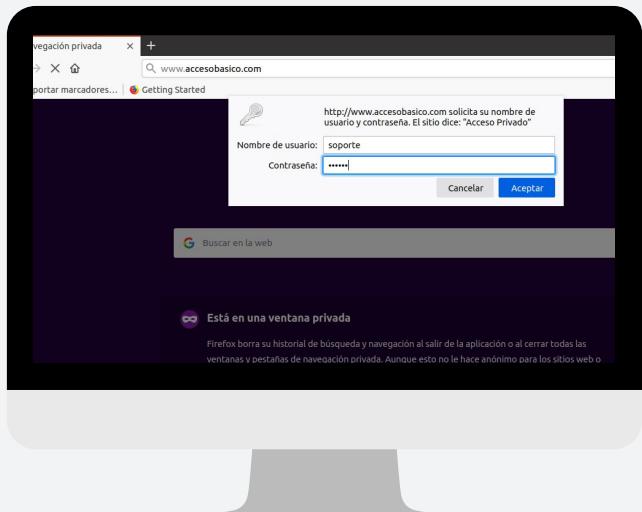
Realizamos las comprobaciones de acceso en el navegador FireFox:

→ **www.accesobasico.com**

Si accedo con un usuario que no pertenece al grupo, me devuelve a la misma página.



Si accedo con un usuario y la contraseña de un usuario que existe en el archivo del grupo, creado anteriormente.



## Index of /

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">estilos_sitio_web/</a>	2021-04-06 14:50	-	
<a href="#">imagenes/</a>	2021-04-02 20:01	-	
<a href="#">indexhtml</a>	2021-04-06 14:35	8.1K	



# CREAR HOST CON CERTIFICADO SSL

Uno de los grandes problemas que tiene la web es con el protocolo http, ya que la transmisión de datos entre cliente y servidor viajan sin encriptar. Esto resulta un gran inconveniente si en estas transmisiones viajan usuarios, credenciales u otros datos sensibles. Una de las soluciones es la activación de las transmisiones ssl o tls de nuestro servidor apache.

**SSL** y **TLS** son protocolos utilizados para envolver el tráfico normal en un envoltorio protegido y cifrado. Usando esta tecnología, los servidores pueden enviar de forma segura información a sus clientes sin que puedan ser interceptados o leídos por una parte externa.

Para ello necesitamos de un certificado que deberemos instalar en el servidor y lo adquirimos de alguna de las autoridades de certificación.

Para instalar dicho certificado necesitamos un directorio donde almacenar las claves y los certificados generados, la instalación de ciertos paquetes como **openssl** y **ca-certificates** en nuestro sistema y un host virtual para establecer el certificado.

Iniciamos nuestra máquina virtual, nos loguearemos como usuario root, en caso contrario tener en cuenta el comando sudo para la ejecución de los scripts. Actualizamos el sistema y arrancamos nuestro servidor apache y comprobar el status del servidor.

```
lsavoglsavo-VirtualBox:~$ sudo -i
root@lsavo-VirtualBox:~# systemctl start apache2
root@lsavo-VirtualBox:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-04-05 11:18:38 CEST; 10min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 654 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 719 (apache2)
      Tasks: 55 (limit: 2221)
     Memory: 8.1M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/apache2.service
               ├─719 /usr/sbin/apache2 -k start
               ├─720 /usr/sbin/apache2 -k start
               └─721 /usr/sbin/apache2 -k start

abr 05 11:18:35 lsavo-VirtualBox systemd[1]: Starting The Apache HTTP Server...
abr 05 11:18:38 lsavo-VirtualBox apachectl[686]: apache2: Could not reliably determine the server's fully qualified domain name, using lsavo-VirtualBox for Port 80
abr 05 11:18:38 lsavo-VirtualBox systemd[1]: Started The Apache HTTP Server.
root@lsavo-VirtualBox:~# a2enmod ssl
```

Primero que debemos comprobar es que el módulo ssl de apache este habilitado.

```
a2enmod ssl
systemctl restart apache2
```

```
root@lsavo-VirtualBox:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
root@lsavo-VirtualBox:~# systemctl restart apache2
root@lsavo-VirtualBox:~#
```

Esto permitirá que las transmisiones entre cliente y servidor vayan encriptadas.

En el caso de tener un **firewall ufw** configurado, hay que abrir puertos **http** y **https**.

<pre>Abrir puertos ufw allow "ApacheFull"</pre>	<pre>root@lsavo-VirtualBox:~# ufw allow "Apache Full" Reglas actualizadas Reglas actualizadas (v6) root@lsavo-VirtualBox:~#</pre>
---	---

Para generar el certificado necesitamos los paquetes **openssl** y **ca-certificates** instalados en nuestro sistema.

En caso de que no estuvieran instalados:

```
apt install openssl ca-certificates
```

Creamos el certificado autofirmado y la clave.

Usaremos el comando **openssl**.

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

- **openssl**: es la herramienta de línea de comandos para crear y administrar certificados, claves y otros archivos de OpenSSL.
- **req -x509**: especifica que deseamos usar la administración de la solicitud de firma de certificados (CSR) X.509. El "X.509" es un estándar de infraestructura de claves públicas al que se adhieren SSL y TLS para la administración de claves y certificados.
- **-nodes**: indica a OpenSSL que omita la opción para proteger nuestro certificado con una frase de contraseña. Necesitamos que Apache pueda leer el archivo, sin intervención del usuario, cuando se inicie el servidor. Una frase de contraseña evitaría que esto suceda porque tendríamos que introducirla tras cada reinicio.
- **-days 365**: esta opción establece el tiempo durante el cual el certificado se considerará válido. En este caso, lo configuramos por un año. Muchos navegadores modernos rechazarán cualquier certificado válido por más de un año.
- **-newkey rsa:2048**: especifica que deseamos generar un nuevo certificado y una nueva clave al mismo tiempo. No creamos la clave que se requiere para firmar el certificado en un paso anterior, por lo que debemos crearla junto con el certificado. La parte **rsa: 2048** le indica que cree una clave RSA de 2048 bits de extensión.
- **-keyout**: indica a OpenSSL dónde colocar el archivo de clave privada generada que estamos creando.
- **/etc/ssl/private/apache-selfsigned.key**: indica el nombre del archivo que contiene la clave y el directorio donde lo ubicaremos.
- **-out**: indica a OpenSSL dónde colocar el certificado que creamos.
- **/etc/ssl/certs/apache-selfsigned.crt**: indica el nombre del archivo que almacenará el certificado y el directorio donde lo ubicaremos.
- **Common Name**: indica el nombre del dominio que utilizará para acceder al servidor o a la IP pública del servidor. Es importante que este campo coincida con lo que pondremos en la barra de direcciones del navegador para acceder al sitio (**ServerName**), ya que un error de concordancia causará más errores de seguridad.

Tras ejecutar el script aparecerá un mensaje en el que hay que introducir información sobre nuestro sitio web.

```
Generating a RSA private key
.....+++++
writing new private key to '/etc/ssl/private/apache-selfsigned.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]:Valencia
Locality Name (eg, city) []:Valencia
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MD
Organizational Unit Name (eg, section) []:MD
Common Name (e.g. server FQDN or YOUR name) []:www.missl.com
Email Address: soporte@missl.com
```

- **Nombre del país en 2 letras según su código iso:** ES
- **Una provincia:** Valencia
- **Una localidad:** Valencia
- **Nombre de una empresa:** MD
- **Una unidad organizativa:** MD
- **Common Name:** indicamos el dominio o la ip que queremos securizar,  
[www.missl.com](http://www.missl.com)
- **El correo del administrador:** soporte@missl.com

El resto de datos se pueden dejar en blanco.

Comprobamos que se hayan creado los archivos listando los directorios.

Si se ha generado la clave, archivo con extensión .key

Listamos directorio  
ls -l /etc/ssl/private/

```
vo-VirtualBox:~$ ls -l /etc/ssl/private/
puede abrir el directorio '/etc/ssl/private/': Permiso
vo-VirtualBox:~$
```

Si intentamos listar el directorio dónde se encuentra la clave con nuestro usuario, nos deniega el permiso.

Esto es debido a que por motivos de seguridad no se puede mostrar. Ya que se trata de la clave privada que utilizará nuestro servidor para desencriptar el tráfico generado hacia él, con la clave pública que haya compartido con el cliente. Y esta debe estar a disposición única y exclusivamente del administrador del sistema.

En cambio en el archivo crt o certificate sign in request, los permisos no son exclusivos del administrador del sistema.

Lo normal es



```
root@isavo-VirtualBox:~# ls /etc/apache2/ssl
apache-selfsigned.crt  apache-selfsigned.key
root@isavo-VirtualBox:~# ls -l /etc/apache2/ssl
total 8
-rw-r--r-- 1 root root 1428 abr  6 12:14 apache-selfsigned.crt
-rw----- 1 root root 1704 abr  6 12:13 apache-selfsigned.key
root@isavo-VirtualBox:~#
```

En caso de no ser así, modificamos dichos permisos.

Modificar permisos del archivo .key  
**chmod 600 apache-selfsigned.key**

Creamos el directorio privadossal en nuestro directorio de publicación.

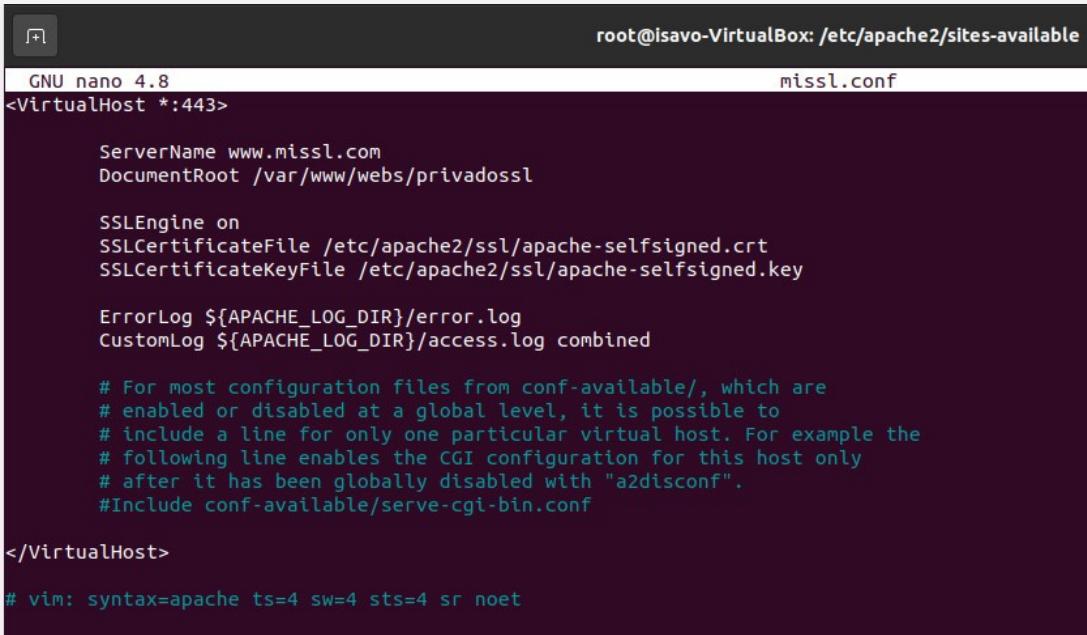
**mkdir /var/www/webs/privadossal**

Comprobamos que se haya creado.

**ls /var/www/webs/**

```
root@isavo-VirtualBox:~# ls /var/www/webs/
acceso  miweb  privadossal
root@isavo-VirtualBox:~#
```

Ahora crearemos el archivo de configuración de nuestro host virtual.



```
root@isavo-VirtualBox: /etc/apache2/sites-available
GNU nano 4.8                         missl.conf
<VirtualHost *:443>

    ServerName www.missl.com
    DocumentRoot /var/www/webs/privadossal

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/apache2/ssl/apache-selfsigned.key

    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
```

Copiamos un archivo .conf.

**cp /etc/apache2/sites-available/miweb.conf /etc/apache2/sites-available/missl.conf**

Editamos el archivo para la configuración de las directivas, en principio una configuración mínima.

**nano /etc/apache2/sites-available/missl.conf**

En las directivas debemos modificar:

- **puerto 443:** por el puerto 80 del container VirtualHost.
- **ServerName:** indicamos el dominio al que queremos que se dirija, www.missl.com. Puede ser un nombre de host, un nombre de dominio completo o una dirección ip. Lo más importante es que debe coincidir con el que hemos indicado en el common Name.
- **DocumentRoot:** indicamos el directorio de publicación de nuestra web.
- **Opciones SSL:** necesarias para apuntar Apache a nuestros archivos de certificado y clave.
- **SSLEngine:** muy importante asegurarnos de que sea on.
- **SSLCertificateFile:** indicar la ruta de nuestro archivo .crt
- **SSLCertificateKeyFile:** indicar la ruta de nuestro archivo .key.

Guardamos cambios y salimos de la edición del archivo.

**Ctrl o > Enter > Ctrl x**

Ahora creamos una web en nuestro directorio de publicación privado o la podemos copiar de otro directorio. Es decir, creamos el archivo **index.html**.

En caso de no ser así, modificamos dichos permisos.

```
Crear el archivo
touch /var/www/webs/privadossal/index.html
Copiar el archivo
cp /var/www/webs/index.html /var/www/webs/privadossal/index.html
```

```
[GNU nano 4.8]
/etc/hosts
127.0.0.1 localhost
127.0.1.1 isavo-VirtualBox
192.168.0.16 www.miweb.com
192.168.0.16 miweb.com
192.168.0.16 www.acceso.com
192.168.0.16 www.accesobasico.com
192.168.0.16 www.mssl.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
```

Lo editamos y construimos el esqueleto de la web en caso de querer modificarla.

**nano /var/www/webs/privadossal/index.html**

Activamos el virtualhost creado.

**a2ensite missl.conf**

Comprobación de sintaxis.

**apache2ctl configtest**

Creamos el dominio en el archivo hosts para la resolución de las DNS.

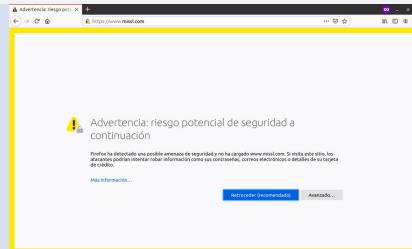
**nano /etc/hosts**

recargamos Apache de forma segura para implementar los cambios.

**systemctl reload apache2**

Realizamos las pruebas introduciendo en el navegador, en modo incognito, e introducimos el dominio.

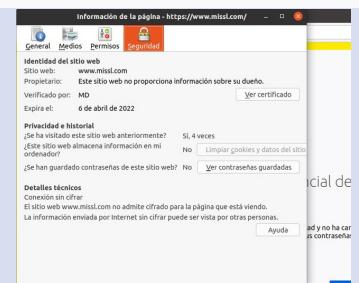
Al introducir el dominio <https://www.mssl.com> nos muestra una advertencia de seguridad.



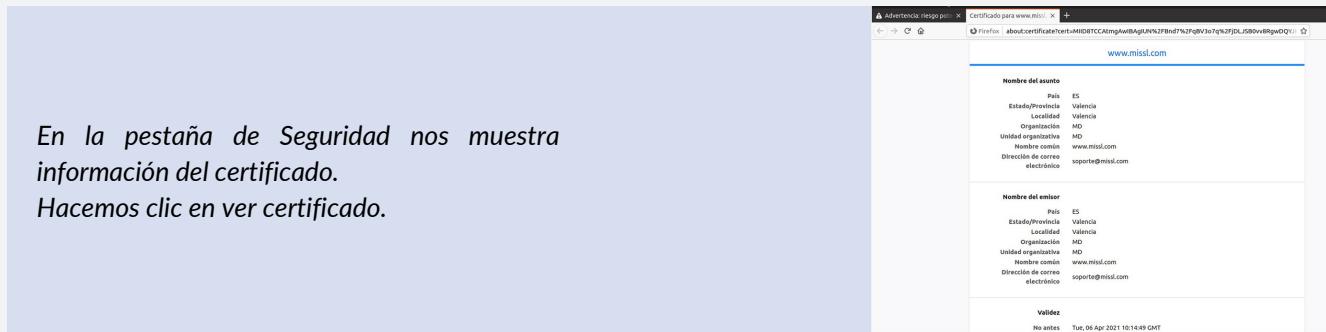
Si hacemos clic en el candado no muestra una ventana, hacemos clic en conexión no segura. Nos muestra un texto indicándonoslo.



Ahora creamos una web en nuestro directorio de publicación privado o la podemos copiar de otro directorio. Es decir, creamos el archivo index.html.



Realizamos las pruebas introduciendo en el navegador, en modo incognito, e introducimos el dominio.



En la pestaña de Seguridad nos muestra información del certificado.

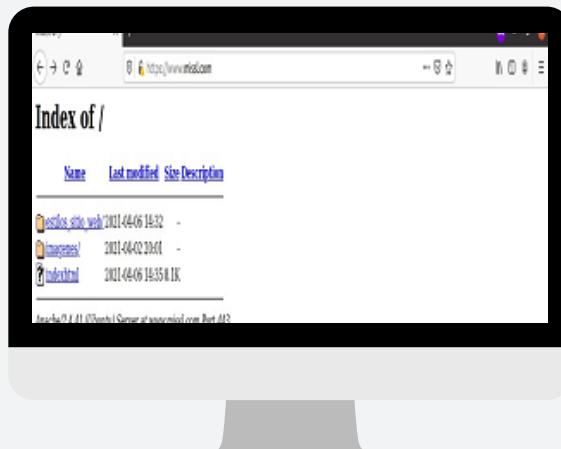
Hacemos clic en ver certificado.

Nos muestra toda la información del certificado que hemos creado, por lo que las conexiones entre cliente y servidor estarán cifradas.

Hacemos clic en Avanzado...en el mensaje de advertencia de nuestro navegador y nos mostrará un cuadro de diálogo.

Acepto el riesgo y continúo.

Ya hemos comprobado que el protocolo <https://> funciona en el dominio que hemos creado.



Al introducir <http://>, en el navegador, nos muestra la página de apache, por lo que en nuestro archivo missl.conf vamos a añadir una redirección.

**Editamos el archivo nano /etc/apache2/sites-available/missl.conf**

Añadimos una nueva sección en la directiva VirtualHost para el puerto 80.

**Directiva VirtualHost:** <VirtualHost \*:80>

**ServerName:** indicamos el dominio [www.missl.com](https://www.missl.com)

**Redirect / https://www.missl.com**

Cerramos: </VirtualHost>

Con la directiva **Redirect** realizamos la redirección del dominio.

Comprobación de errores en la configuración.  
**apache2ctl configtest**

Recargar Apache de forma segura para implementar los cambios.  
**systemctl reload apache2**

```
GNU nano 4.8
<VirtualHost *:80>
    ServerName www.missl.com
    Redirect / https://www.missl.com

</VirtualHost>

<VirtualHost *:443>
    ServerName www.missl.com
    DocumentRoot /var/www/webs/privadossal

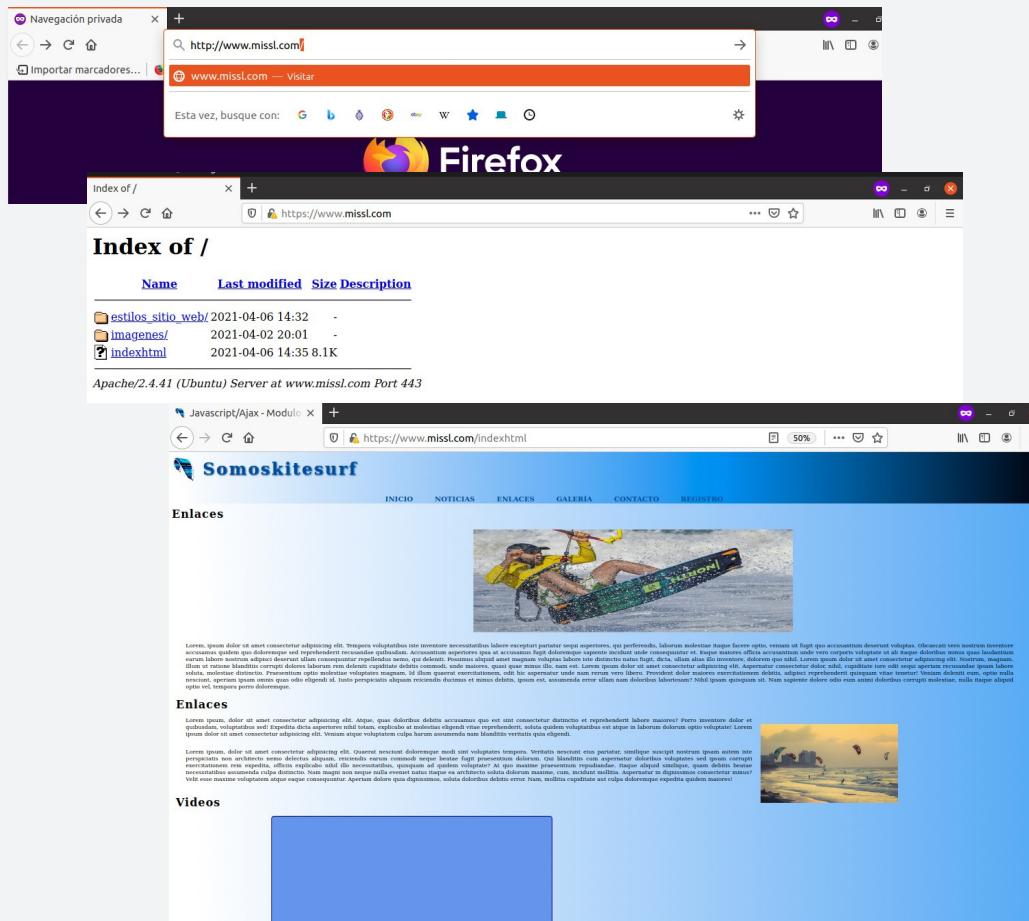
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/apache2/ssl/apache-selfsigned.key

    <Directory /var/www/webs/privadossal>
        Options +Indexes +FollowSymLinks +MultiViews
        AuthType Basic
        Require valid-user
        AuthBasicProvider file
        AuthUserFile passwords
        AuthName "Solo personal autorizado"
    </Directory>

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

Ahora en el navegador introducimos <http://www.missl.com> y nos redirige a la página con el certificado.

Hacemos clic en el enlace index.html y nos muestra la web.



Ahora configuramos el directorio con autenticación de usuarios.

Los usuarios que tendrán acceso son los que hemos creado anteriormente en el archivo **passwords**

Configuramos las directivas para la autenticación básica en el archivo **missl.conf**

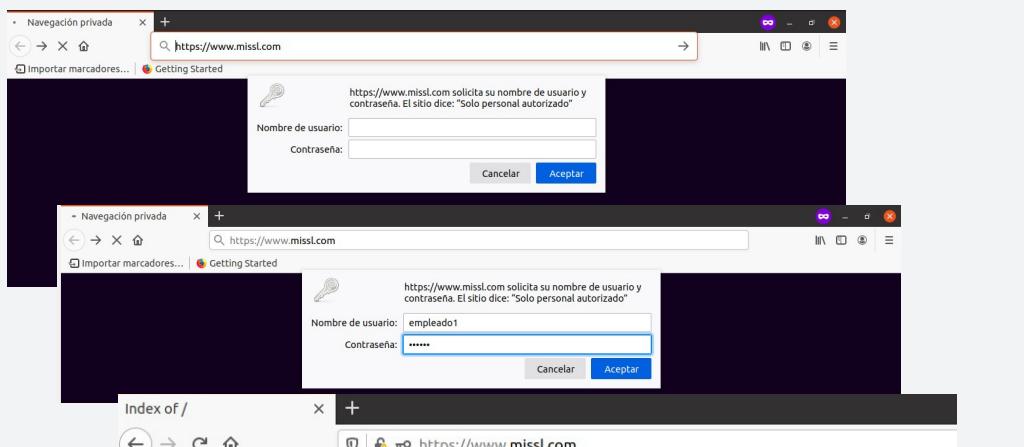
```
GNU nano 4.8
/etc/apache2/sites-available/missl.conf
<VirtualHost *:80>
    ServerName www.missl.com
    Redirect / https://www.missl.com
</VirtualHost>

<VirtualHost *:443>
    ServerName www.missl.com
    DocumentRoot /var/www/webs/privadossal
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/apache2/ssl/apache-selfsigned.key
    <Directory /var/www/webs/privadossal>
        Options +Indexes +FollowSymLinks +MultiViews
        AuthType Basic
        Require valid-user
        AuthBasicProvider file
        AuthUserfile passwords
        AuthName "Solo personal autorizado"
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

**Editamos el archivo**  
**nano /etc/apache2/sites-available/missl.conf**  
**Comprobación de sintaxis.**  
**apache2ctl configtest**  
**Recargar Apache de forma segura para implementar los cambios.**  
**systemctl restart apache2**  
**Creamos el archivo index.html.**  
**nano /var/www/webs/privadossal/index.html**

Realizamos las pruebas introduciendo en el navegador, modo incognito, e introducimos el dominio.

Nos solicita autenticación.



Name	Last modified	Size	Description
estilos_sitio_web/	2021-04-06 14:50	-	
imagenes/	2021-04-02 20:01	-	
indexhtml	2021-04-06 14:35	8.1K	

Apache/2.4.41 (Ubuntu) Server at www.missl.com Port 443



Lorem, ipsum dolor sit amet consectetur adipisicing elit. Tempora voluptatibus iste inventore necessitatibus labore excepturi pariatur sequi asperiores, qui perferendis, laborum molestiae itaque facere optio, veniam sit fugit quo accusantium deserunt voluptas. Obcaecati vero nostrum

Probamos con [www.missl.com](http://www.missl.com)

The screenshot shows a web browser window with two tabs. The top tab is titled 'Index of /' and displays a file list. The bottom tab shows a login dialog box.

**Login Dialog:**

- Title: 'Getting Started'
- Message: 'https://www.missl.com solicita su nombre de usuario y contraseña. El sitio dice: "Solo personal autorizado"'
- Fields: 'Nombre de usuario:' (empty) and 'Contraseña:' (empty)
- Buttons: 'Cancelar' and 'Aceptar'

**Index of / Tab:**

- Header: 'Index of /'
- Message: 'Apache/2.4.41 (Ubuntu) Server at www.missl.com Port 443'
- Table:
 

Name	Last modified	Size	Description
estilos_sitio_web/	2021-04-06 14:50	-	
imagenes/	2021-04-02 20:01	-	
indexhtml	2021-04-06 14:35	8.1K	

Si probamos con <http://www.missl.com> nos redirige al dominio.

The screenshots show three separate browser tabs, all displaying the same directory index page for the root directory ('/').

**Top Tab (http://www.missl.com):**

- Header: 'Index of /'
- Message: 'Apache/2.4.41 (Ubuntu) Server at www.missl.com Port 443'
- Table:
 

Name	Last modified	Size	Description
estilos_sitio_web/	2021-04-06 14:50	-	
imagenes/	2021-04-02 20:01	-	
indexhtml	2021-04-06 14:35	8.1K	

**Middle Tab (https://www.missl.com):**

- Header: 'Index of /'
- Message: 'Apache/2.4.41 (Ubuntu) Server at www.missl.com Port 443'
- Table:
 

Name	Last modified	Size	Description
estilos_sitio_web/	2021-04-06 14:50	-	
imagenes/	2021-04-02 20:01	-	
indexhtml	2021-04-06 14:35	8.1K	

**Bottom Tab (https://www.missl.com):**

- Header: 'Index of /'
- Message: 'Apache/2.4.41 (Ubuntu) Server at www.missl.com Port 443'
- Table:
 

Name	Last modified	Size	Description
estilos_sitio_web/	2021-04-06 14:50	-	
imagenes/	2021-04-02 20:01	-	
indexhtml	2021-04-06 14:35	8.1K	