

PARCIAL ELECTIVA V

Juan Camilo Torres Beltrán

Institución Universitaria EAM

Electiva V

Armenia, Quindío

2023

Tabla de contenido

Introducción	3
Objetivos	4
Objetivo General:	4
Objetivos Específicos:.....	4
Desarrollo	5
1. Crear un archivo ejecutable .sh (Taller 2)	5
2. Crear un servidor nginx, junto con tu propio archivo default.....	9
3. Crear partición de disco duro en máquina virtual Ubuntu	13

Introducción

En la era digital actual, la capacidad de automatizar tareas y configurar entornos de servidor es esencial para optimizar la eficiencia y el rendimiento de las infraestructuras informáticas. Este trabajo se adentra en un proceso paso a paso para crear un archivo de script, configurar un servidor web utilizando Nginx y crear un disco duro en una máquina virtual con Ubuntu, todo ello utilizando la potencia de la línea de comandos. Estos pasos son fundamentales para quienes buscan comprender cómo gestionar y mantener sistemas Linux, así como para aquellos que desean aprender a automatizar tareas y administrar servidores de manera eficiente.

A lo largo de este trabajo, exploraremos cómo crear un script, para luego ejecutarlo en una máquina virtual Ubuntu. Además, aprenderemos a configurar un servidor web Nginx para alojar sitios web y aplicaciones web, lo que es esencial en el mundo de la administración de servidores. Por último, abordaremos la creación de un disco duro virtual en la máquina virtual, lo que permitirá el almacenamiento y la gestión de datos de manera efectiva.

Este proceso, que involucra la interacción con una máquina virtual a través de comandos de terminal, es una habilidad esencial para quienes desean sumergirse en el mundo de la administración de sistemas y servidores. A lo largo de esta guía, se presentarán los comandos y las técnicas necesarias para lograr estos objetivos de manera efectiva, llevando cada proceso siempre paso a paso, y finalizando con el aprendizaje de estos.

Objetivos

Objetivo General:

El objetivo general de este trabajo es proporcionar una guía detallada que permita a los usuarios aprender y comprender el proceso de creación de un archivo script, configuración de un servidor web Nginx y creación de un disco duro en una máquina virtual con Ubuntu, utilizando comandos de terminal. Se busca capacitar a los lectores en habilidades esenciales de administración de sistemas y servidores, así como en la automatización de tareas en entornos Linux.

Objetivos Específicos:

1. Detallar el proceso paso a paso para crear un archivo script en el sistema Ubuntu. Esto incluirá la elección de un lenguaje de scripting, la escritura de código y la configuración de permisos adecuados.
2. Explicar cómo instalar y configurar el servidor web Nginx en una máquina virtual Ubuntu. Esto incluirá la creación de sitios web virtuales y la configuración de reglas de enrutamiento.
3. Guiar a los usuarios a través del proceso de creación de un disco duro virtual en la máquina virtual Ubuntu. Esto permitirá el almacenamiento y la gestión de datos de manera efectiva.
4. Presentar una serie de comandos de terminal específicos para cada objetivo, explicando su uso y función. Los lectores aprenderán cómo ejecutar estos comandos de manera efectiva.

Desarrollo

1. Crear un archivo ejecutable .sh (Taller 2)

- **PRIMERO:** creamos una nueva carpeta, con el siguiente comando y luego nos situamos en esa carpeta.

```
juan@juan-VirtualBox:~/EAM_CNC$ mkdir parcial
juan@juan-VirtualBox:~/EAM_CNC$ cd parcial
juan@juan-VirtualBox:~/EAM_CNC/parcial$
```

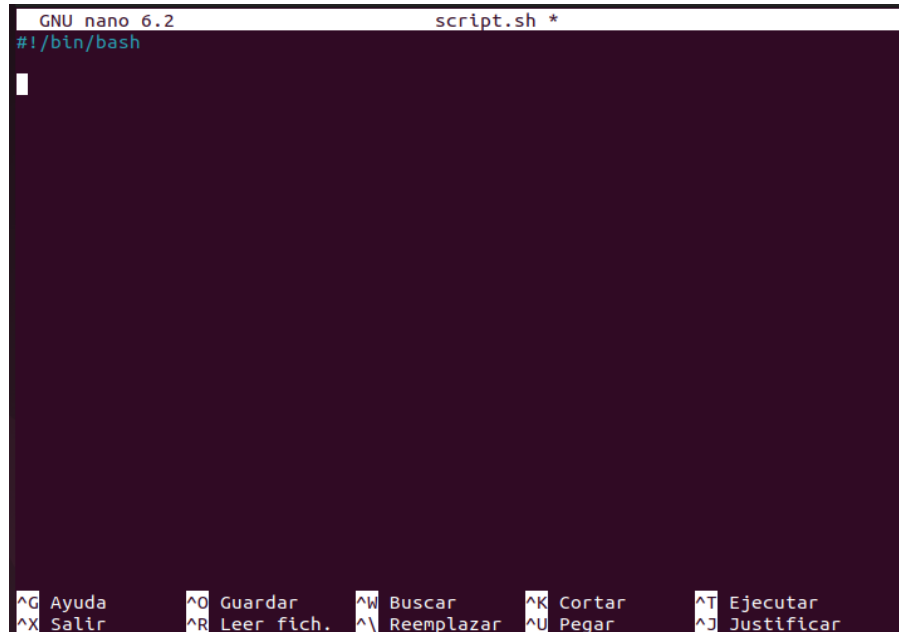
- **SEGUNDO:** creamos un archivo en la carpeta que acabamos de crear, que tenga la extensión “*.sh*”, con la finalidad de que sea un script de Shell.

```
juan@juan-VirtualBox:~/EAM_CNC/parcial$ nano script.sh
```

Utilizamos el comando “*nano*” para crearlo y editarlo de una vez, con el editor nano. Nos abriría una ventana como esta en la que podemos empezar a escribir sobre nuestro archivo.



- **TERCERO:** Agregamos la notación “**#!/bin/bash**”, esto para que el sistema identifique que interprete de comandos va a ejecutar el archivo, en este caso, “**bash**”. También sirve para que lo podamos ejecutar directamente, luego de que se le den permisos de ejecución.

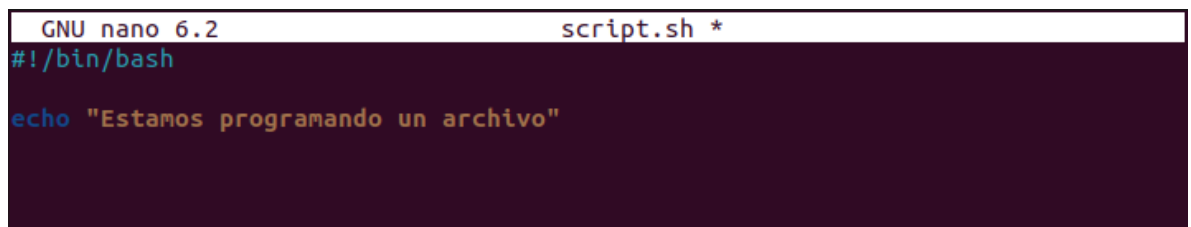


```
GNU nano 6.2 script.sh *
#!/bin/bash

```

^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar

Posteriormente le agregamos código para poder ver le funcionamiento. Podemos agregar algo como un simple mensaje, usando el comando “echo”. Para guardar los cambios oprimimos “**CTRL + O**”, “**ENTER**”; para confirmar y luego “**CTRL + X**”, para salir.



```
GNU nano 6.2 script.sh *
#!/bin/bash
echo "Estamos programando un archivo"

```

- **CUARTO:** Ahora ya tenemos el archivo, pero todavía no lo podemos ejecutar. Si ejecutamos el comando “**ls -l**”, nos permitirá ver los permisos que tienen los archivos de la carpeta en la que nos encontramos, en este caso si lo ejecutamos y vemos los permisos del archivo que creamos veremos que.....

```
juan@juan-VirtualBox:~/EAM_CNC/parcial$ ls -l
total 4
-rw-rw-r-- 1 juan juan 51 ago 26 22:59 script.sh
```

Como podemos ver en la imagen, nos tenemos permisos de ejecución (x), por lo tanto, aun no podemos ejecutar el archivo, tendremos que primero dárselos. Esto lo podemos hacer fácilmente con el comando “**chmod**”, el cual nos permite cambiar los permisos de un archivo o carpeta, conforme queramos.

```
juan@juan-VirtualBox:~/EAM_CNC/parcial$ chmod 777 script.sh
```

El siguiente comando seguido de “777” y el archivo al que le queremos modificar los permisos. En este caso se coloca “777” para asignar permisos de lectura, escritura y ejecución a las tres categorías disponibles, siendo la primera correspondiente, al usuario propietario, el segundo al grupo, y el tercero a otros usuarios. Además, también teniendo en cuenta que: lectura = 4, lectura = 2 y ejecución = 1; siendo esta la razón por la cual se coloca 3 veces 7. Ahora si volvemos a verificar los permisos con el comando “**ls -l**” veremos que:

```
juan@juan-VirtualBox:~/EAM_CNC/parcial$ ls -l
total 4
-rwxrwxrwx 1 juan juan 51 ago 26 22:59 script.sh
```

Ahora tenemos permisos de ejecución para las tres categorías, en caso que no queramos que las tres categorías tengan permisos de ejecución podemos modificarlo, por ejemplo, haciendo que solo el propietario tenga permisos para llevar a cabo esta labor. Sería algo como:

```
juan@juan-VirtualBox:~/EAM_CNC/parcial$ chmod 764 script.sh
juan@juan-VirtualBox:~/EAM_CNC/parcial$ ls -l
total 4
-rwxrw-r-- 1 juan juan 51 ago 26 22:59 script.sh
```

- **QUINTO:** Ya teniendo los permisos de ejecución sobre el archivo, podemos pasar a ver si se ejecuta lo que hemos programado, ósea nuestro mensaje. Para hacer esto podemos utilizar el comando “**sh NOMBRE_ARCHIVO**” con este, podemos ejecutar el archivo que hemos creado.

```
juan@juan-VirtualBox:~/EAM_CNC/parcial$ sh script.sh
Estamos programando un archivo
```

- **SEXTO:** Ya teniendo esto hecho y claro, podemos manipular el archivo como queramos, agregando funciones, variables y elementos de programación que queramos.

```
GNU nano 6.2 script.sh
#!/bin/bash

echo "Estamos programando un archivo"
echo "-----"

numero=${1}
mostrarNumero(){
echo "El numero ingresado por el usuario es:$numero"
}

mostrarNumero

[ 11 líneas leídas ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar      ^J Justificar
```

```
juan@juan-VirtualBox:~/EAM_CNC/parcial$ sh script.sh 1
Estamos programando un archivo
-----
El numero ingresado por el usuario es:1
```


2. Crear un servidor nginx, junto con tu propio archivo default

- **PRIMERO:** Ingresamos como usuario “*root*”, con el siguiente comando “*sudo su*”, después de esto, si es la primera vez, seguramente nos las contraseña, solo nos encargamos de ingresarla y veremos algo como esto.

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

- **SEGUNDO:** Ejecutamos el comando “*apt update*”, con la finalidad de actualizar los paquetes de nuestro sistema.

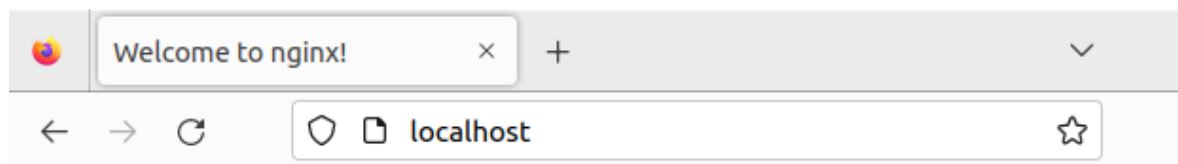
```
root@juan-VirtualBox:/home/juan# apt update
```

- **TERCERO:** Ahora ejecutamos el comando “*apt install nginx*” para poder instalarlo. Nos aparecerá algo como esto:

```
root@juan-VirtualBox:/home/juan# apt install nginx
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx-common nginx-core
Paquetes sugeridos:
  fcgiwrap nginx-doc
Se instalarán los siguientes paquetes NUEVOS:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx nginx-common nginx-core
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 193 no actualizados.
Se necesita descargar 697 kB de archivos.
Se utilizarán 2.395 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Oprimimos la tecla “*S*”, para seguir con la instalación, y tendríamos instalado nginx.

- **CUARTO:** Para verificar si, efectivamente el servidor se encuentra corriendo, podemos abrir un navegador y colocar *“localhost”*, en donde nos saldrá algo como esto:



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Otra forma de verificar en qué estado se encuentra el servidor es ejecutando el comando *“systemctl status nginx”*:

```
root@juan-VirtualBox:/home/juan# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor prese
   Active: active (running) since Sun 2023-08-27 08:18:09 -05; 5min ago
     Docs: man:nginx(8)
   Process: 3840 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_pro
   Process: 3841 ExecStart=/usr/sbin/nginx -g daemon on; master_process on;
  Main PID: 3927 (nginx)
    Tasks: 5 (limit: 4456)
   Memory: 3.6M
      CPU: 78ms
   CGroup: /system.slice/nginx.service
           └─3927 "nginx: master process /usr/sbin/nginx -g daemon on; mast
             └─3930 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" ""
             └─3931 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" ""
             └─3932 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" ""
             └─3933 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" ""

ago 27 08:18:09 juan-VirtualBox systemd[1]: Starting A high performance web s
ago 27 08:18:09 juan-VirtualBox systemd[1]: Started A high performance web se
lines 1-19/19 (END)
```

Podemos ver que nos dice que el servicio se encuentra activo y corriendo perfectamente.

- **QUINTO:** Podemos confirmar si la sintaxis de nuestro archivo es correcta, ejecutando el comando **“nginx -t”**

```
root@juan-VirtualBox:/home/juan# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

- **SEXTO:** Ahora nos vamos a mover a la carpeta **“/etc/nginx/”**

```
root@juan-VirtualBox:/home/juan# cd /etc/nginx/
```

Si listamos los archivos que tenemos aquí, veremos:

```
conf.d          koi-win         nginx.conf      sites-enabled
fastcgi.conf    mime.types     proxy_params   snippets
fastcgi_params  modules-available  scgi_params    uwsgi_params
koi-utf         modules-enabled  sites-available win-utf
```

Ahora si nos vamos a la carpeta **“sites-available”**, encontraremos la configuración default, la cual la podemos modificar como queramos.

```
root@juan-VirtualBox:/etc/nginx# cd sites-available/
root@juan-VirtualBox:/etc/nginx/sites-available# ls
default
```

Si vemos lo que tiene dentro veremos detalles del puerto al que apunta, entre otros.

```
root@juan-VirtualBox:/etc/nginx/sites-available# cat default
```

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;
```

- **SEPTIMO:** Ahora vamos a configurar el archivo **“.html”**, que se muestra cuando vamos a la ruta **“localhost/”**, para ello necesitamos acceder a la ruta, en donde se encuentra ese archivo, la cual es **“/var/www/html”**, que se encuentra en el archivo de configuración mostrado anteriormente. Si listamos los archivos que se encuentran aquí nos encontraremos con el archivo por defecto:

```
root@juan-VirtualBox:/var/www/html# ls
index.nginx-debian.html
```

Lo que haremos será borrar este archivo y crear uno propio, que tenga el aspecto que nosotros queramos, para esto ejecutamos **“rm NOMBRE_ARCHIVO”**, para borrar.

```
root@juan-VirtualBox:/var/www/html# rm index.nginx-debian.html
root@juan-VirtualBox:/var/www/html# ls
root@juan-VirtualBox:/var/www/html#
```

Vemos que ya no tenemos el archivo, ahora lo que haremos será crear uno propio, que tenga extensión **“.html”**

```
root@juan-VirtualBox:/var/www/html# nano index.html
```

Ya aquí, podemos personalizar nuestro archivo como queramos.

```
GNU nano 6.2 index.html *
<h1 style="color:red">Hola bienvenido a mi primer server!</h1>
```

Y le damos **“CTRL + O”** para guardar, y **“CTRL + X”** para salir, ahora lo que necesitamos es recargar el servidor, al menos es lo más recomendable cuando se hace un cambio sobre este, para ello ejecutaremos el comando **“systemctl restart nginx”**.

```
root@juan-VirtualBox:/var/www/html# systemctl restart nginx
```

Ahora si vamos y recargamos **“localhost/”** nos encontraremos con el contenido del archivo que modificamos.



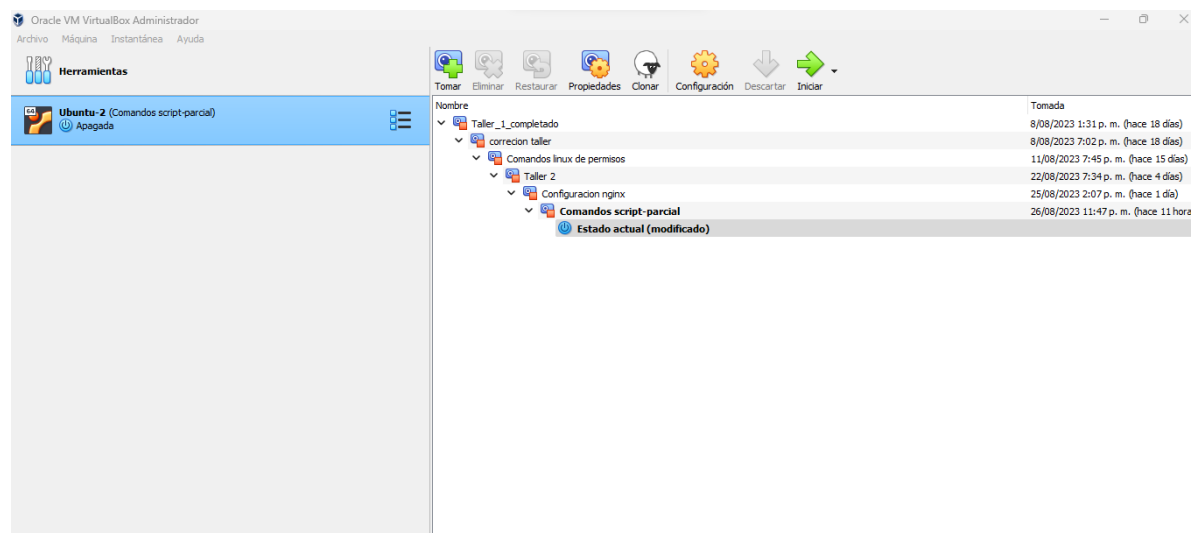
Hola bienvenido a mi primer server!

- **OCTAVO:** Si queremos que el servidor se mantenga encendido siempre, podemos ejecutar el comando “*systemctl enable nginx*”

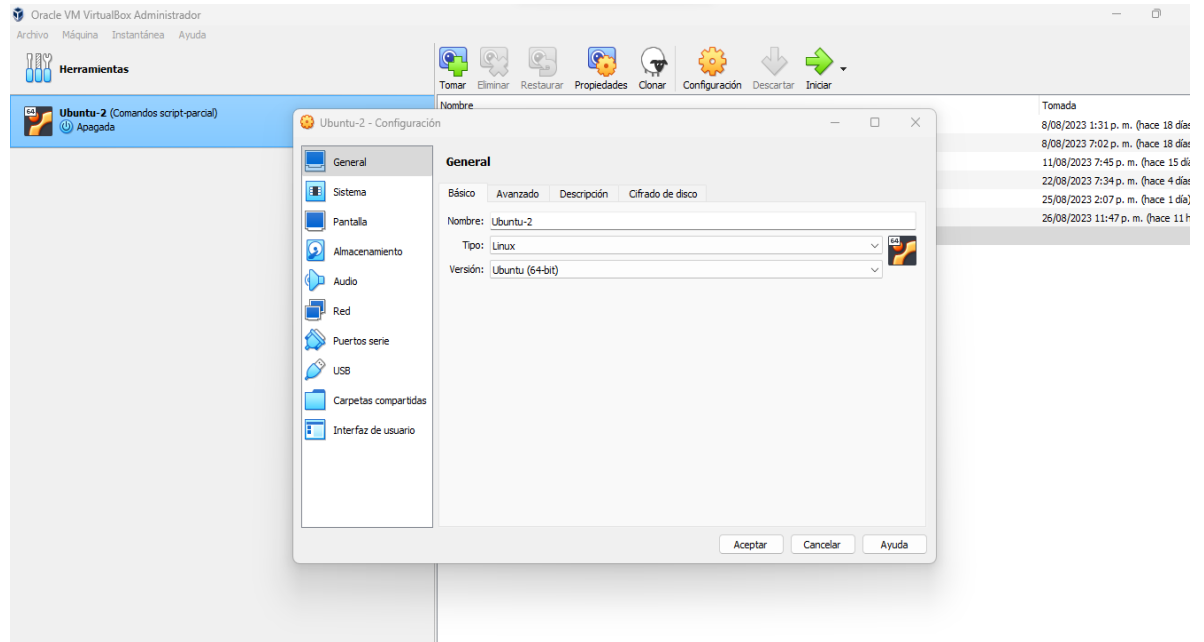
```
root@juan-VirtualBox:/var/www/html# systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd
/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nginx
root@juan-VirtualBox:/var/www/html#
```

3. Crear partición de disco duro en máquina virtual Ubuntu

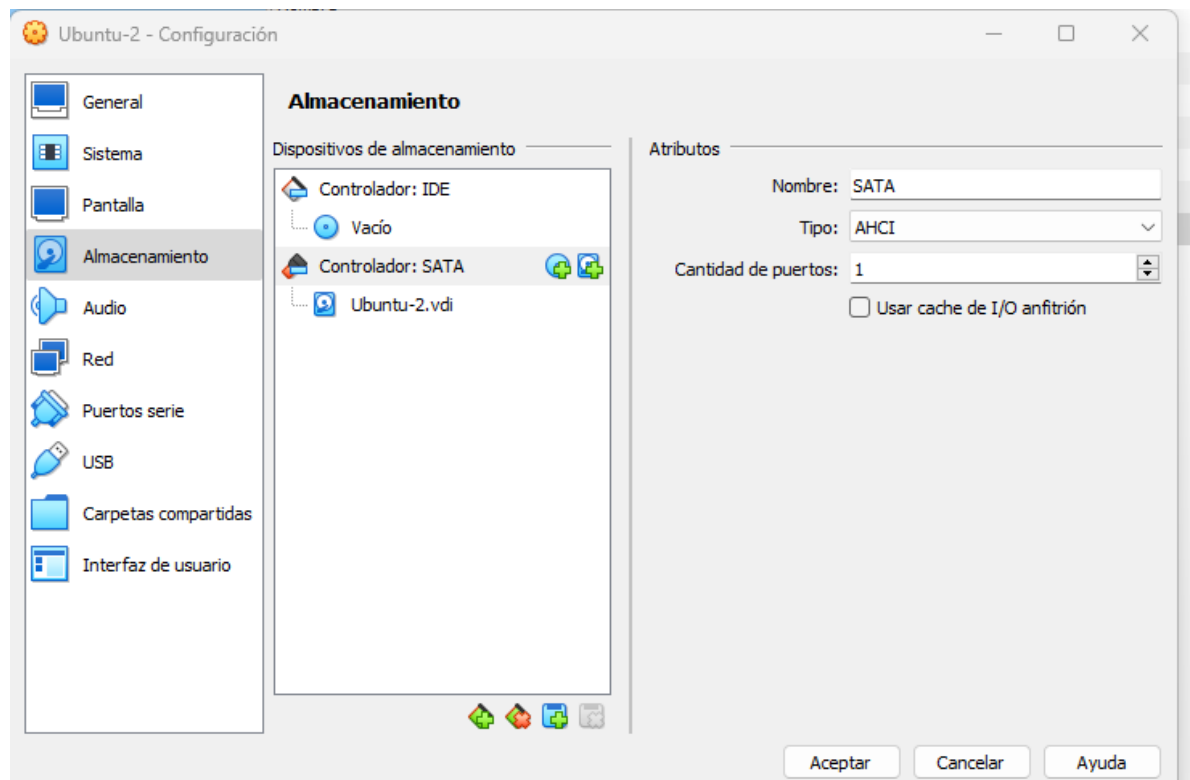
- **PRIMERO:** Ingresamos al programa que estemos utilizando, en mi caso virtual box, y nos paramos en la máquina virtual a la cual le queremos agregar el nuevo disco.



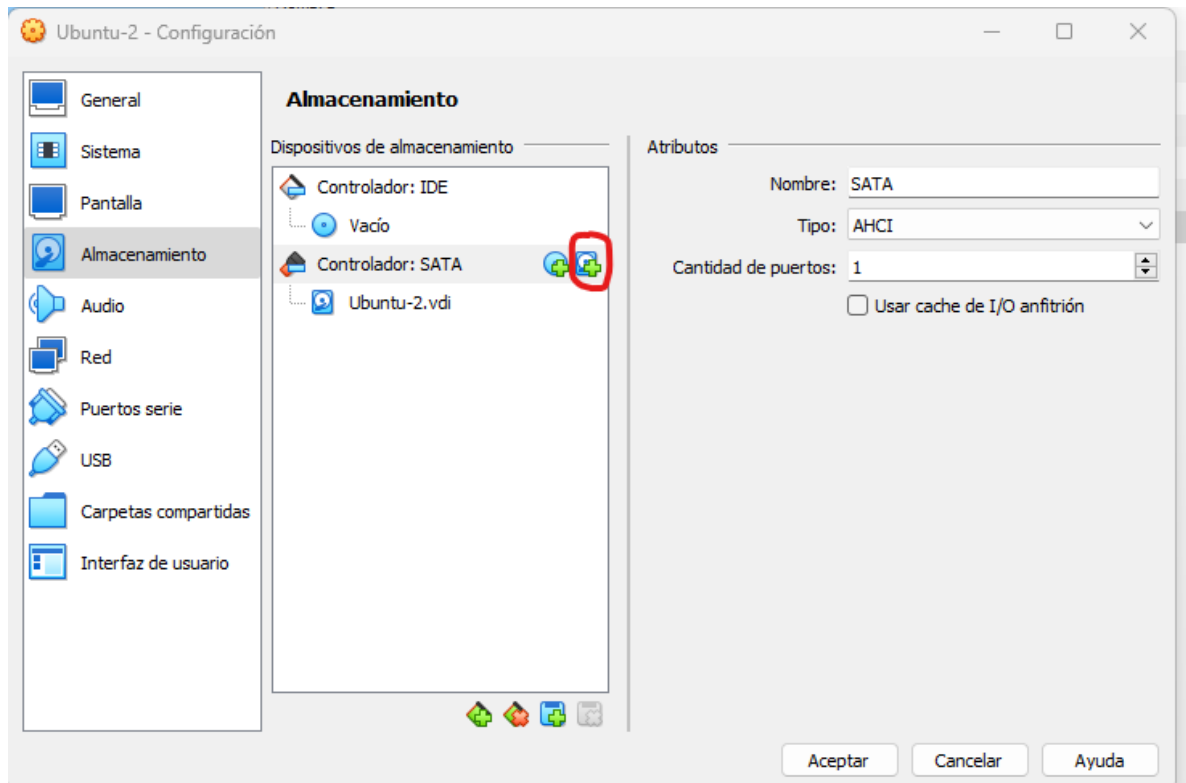
- **SEGUNDO:** Seleccionamos la opción “*configuración*”, se puede hacer oprimiendo click izquierdo sobre la máquina virtual, y oprimiendo en configuración o seleccionando la opción en la parte superior, luego de esto nos debe aparecer un apartado como esto.



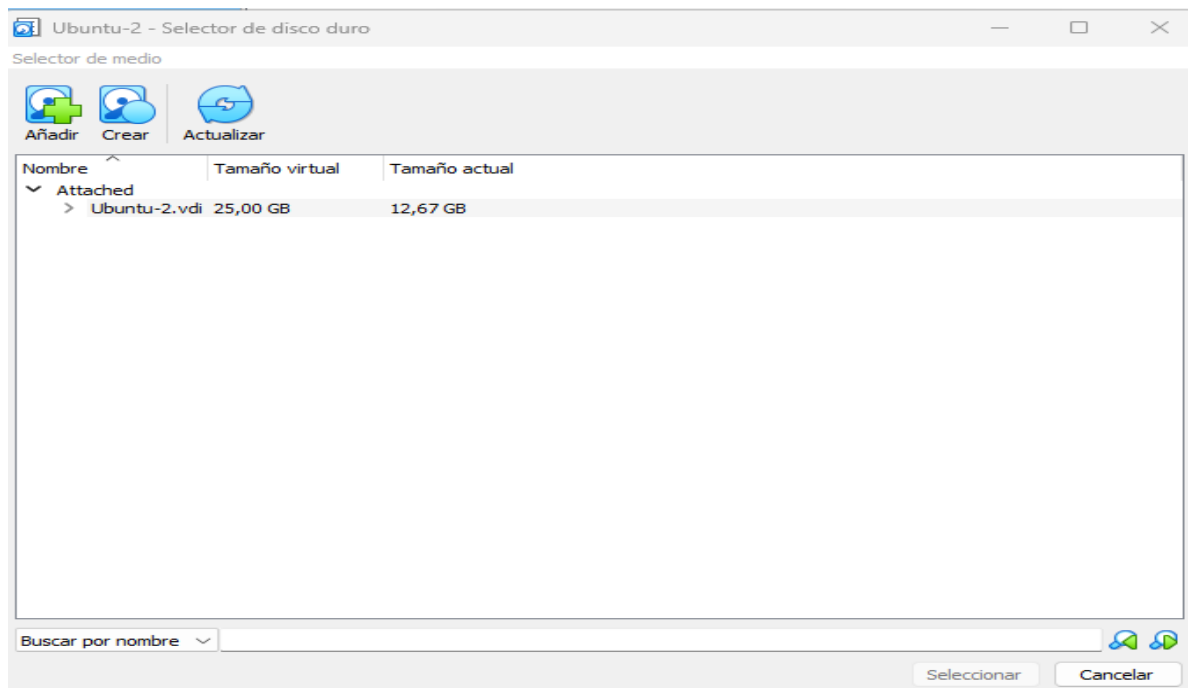
- **TERCERO:** Seleccionamos el apartado “*almacenamiento*”, en donde en el apartado “*SATA*”, veremos los discos que tenemos.



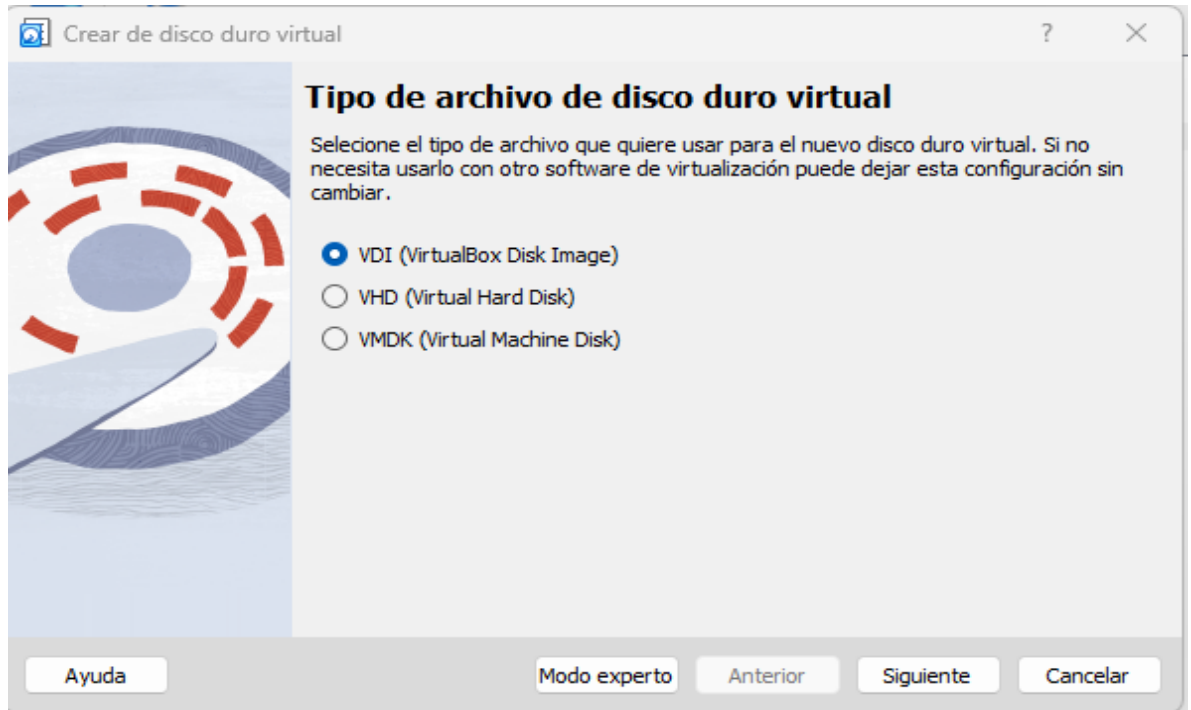
Posteriormente veremos dos opciones para agregar, la primera no nos interesa, ya que es para agregar disco óptico, así que seleccionamos la segunda que es para agregar “disco duro”.



- **CUARTO:** Una vez nos abra la pestaña, oprimiremos en la opción que dice “*Crear*”.



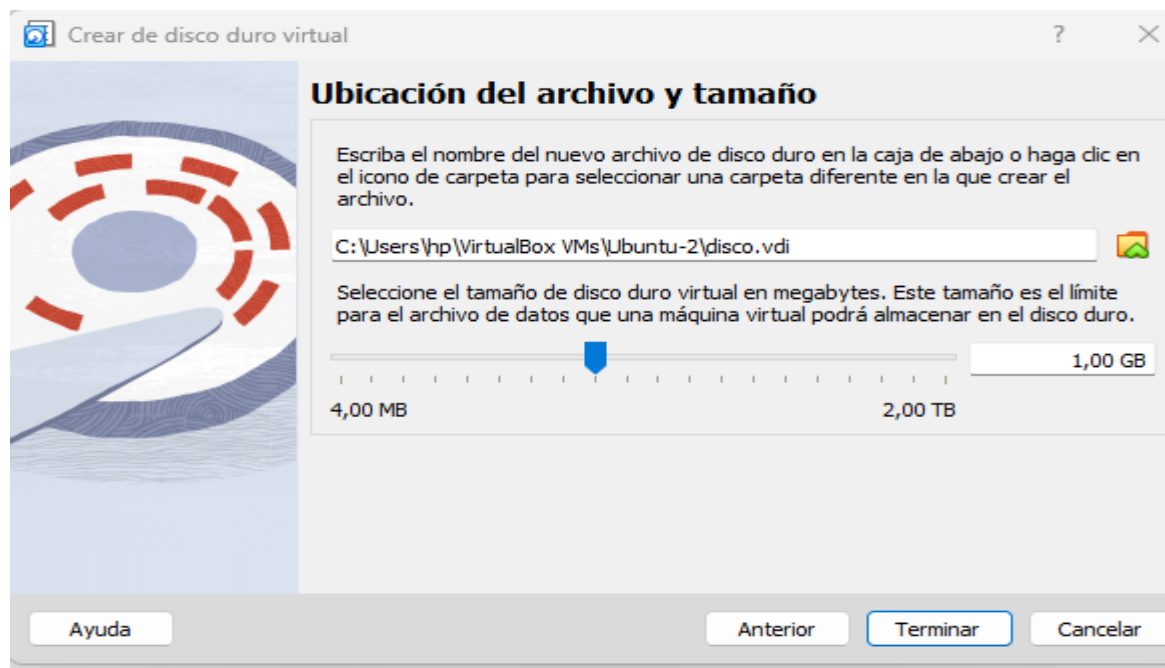
Nos pedirá seleccionar el tipo de disco duro que queremos, en este caso seleccionamos “*VDI*” y damos en siguiente.



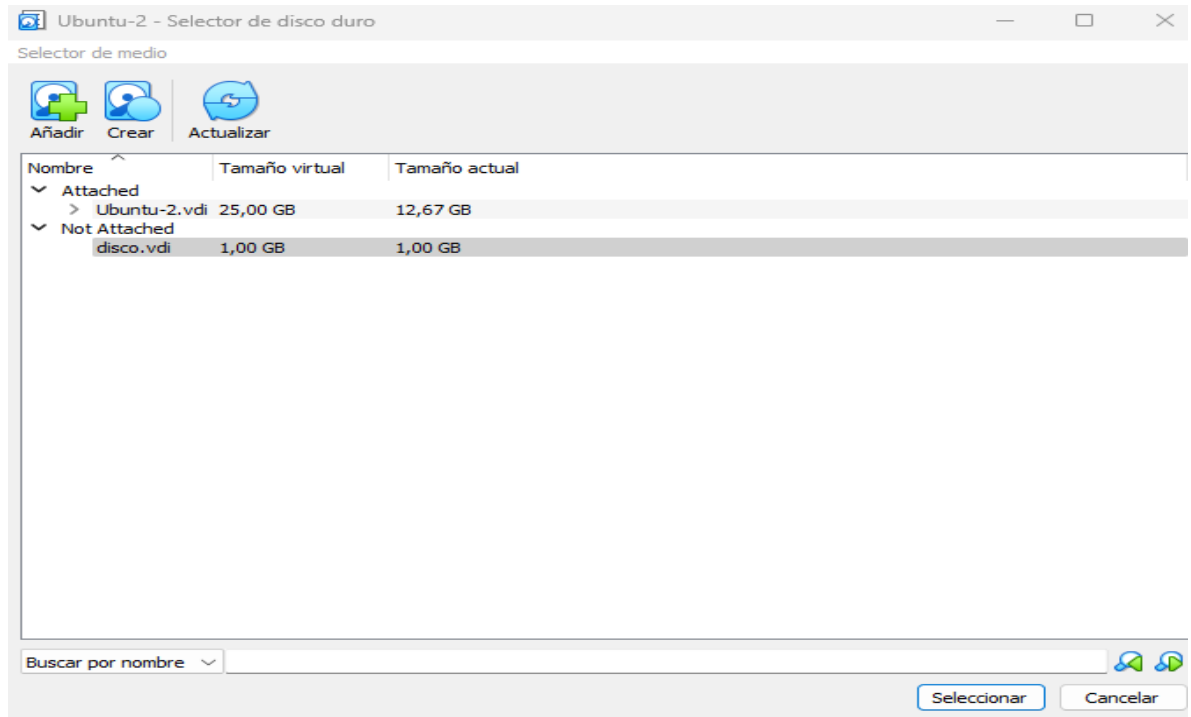
Seleccionamos la opción reservar completamente para el caso del almacenamiento y le damos en siguiente.



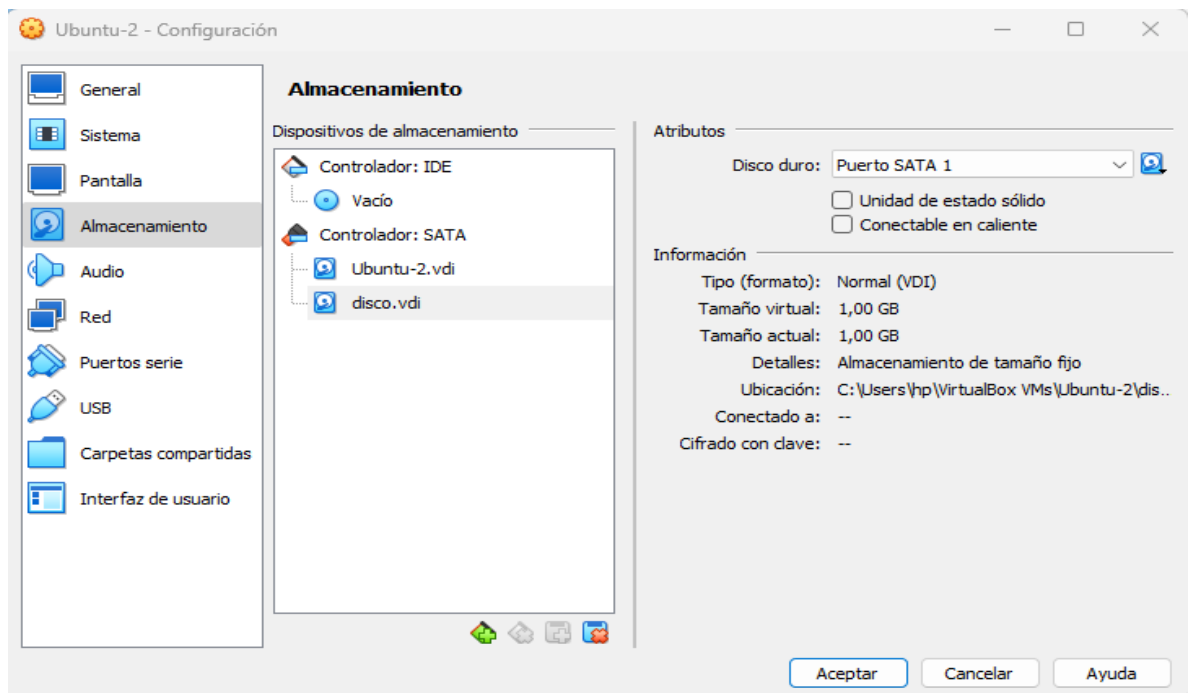
En este apartado, podemos seleccionar la ruta en la cual se va a guardar el nuevo disco que vamos a crear, también podemos definir un tamaño y como se va a llamar, en este caso vamos a llamarlo solo **“disco”**, el cual solo tendrá un tamaño de 1GB y le damos en terminar.



Visualizamos que el nuevo disco ya se observa en la lista de discos disponibles, así que lo seleccionamos y le damos en seleccionar.



Después de esto, volvemos a seleccionar el disco en la lista de controlador SATA, y le damos en aceptar.



- **QUINTO:** Ahora iniciamos nuestra máquina virtual, y abrimos la terminal. Ingresamos como usuario root, utilizando el comando “*sudo su*”, para luego listar los discos que tenemos disponibles con el comando “*lsblk*”, en donde verificamos que este el disco que acabamos de crear.

```

juan@juan-VirtualBox:~$ sudo su
[sudo] contraseña para juan:
root@juan-VirtualBox:/home/juan# lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0    4K  1 loop /snap/bare/5
loop1       7:1      0   63,4M  1 loop /snap/core20/1974
loop2       7:2      0   63,5M  1 loop /snap/core20/2015
loop3       7:3      0   73,9M  1 loop /snap/core22/858
loop4       7:4      0  240,6M  1 loop /snap/firefox/2356
loop5       7:5      0  346,3M  1 loop /snap/gnome-3-38-2004/119
loop6       7:6      0  485,5M  1 loop /snap/gnome-42-2204/126
loop7       7:7      0   91,7M  1 loop /snap/gtk-common-themes/1535
loop8       7:8      0   45,9M  1 loop /snap/snap-store/638
loop9       7:9      0  349,7M  1 loop /snap/gnome-3-38-2004/143
loop10      7:10     0   12,3M  1 loop /snap/snap-store/959
loop11      7:11     0   40,8M  1 loop /snap/snapd/19993
loop12      7:12     0   304K  1 loop /snap/snapd-desktop-integration/49
loop13      7:13     0   53,3M  1 loop /snap/snapd/19457
loop14      7:14     0   452K  1 loop /snap/snapd-desktop-integration/83
sda         8:0      0    25G  0 disk
├─sda1      8:1      0   512M  0 part /boot/efi
└─sda2      8:2      0   24,5G  0 part /var/snap/firefox/common/host-hunspell/
sdb         8:16     0    1G  0 disk
sr0         11:0     1  1024M  0 rom

```

- **SEXTO:** Ahora tenemos que entrar a crear la partición que necesitamos, esto lo podemos hacer con el comando “*fdisk*” junto con la ruta “*/dev/sdb*” el cual nos permite una serie de opciones para gestionar las particiones que tengamos o que vayamos a crear.

```

root@juan-VirtualBox:/home/juan# fdisk /dev/sdb

```

Nos debe aparecer una lista de opciones como esta

```

Bienvenido a fdisk (util-linux 2.37.2).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0x0b
6664e6.

Orden (m para obtener ayuda): m
Ayuda:

DOS (MBR)
a  conmuta el indicador de iniciable
b  modifica la etiqueta de disco BSD anidada
c  conmuta el indicador de compatibilidad con DOS

General
d  borra una partición
l  lista el espacio libre no particionado
l  lista los tipos de particiones conocidos
n  añade una nueva partición
p  muestra la tabla de particiones
t  cambia el tipo de una partición
v  verifica la tabla de particiones
i  imprime información sobre una partición

```

Como aparece en la lista de opciones, oprimimos “*n*”, que corresponde a crear una nueva partición.

```
Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primary, 0 extended, 4 free)
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p):
```

Luego seleccionamos el tipo de partición que vamos a crear, en este caso “*primaria*”. Además, le asignamos un numero de partición.

```
Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primary, 0 extended, 4 free)
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 1): 2
```

Ahora le asignamos un tamaño a la partición en este caso le colocamos 500M

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097151, valor predeterminado 2097151): +500M

Crea una nueva partición 2 de tipo 'Linux' y de tamaño 500 MiB.

Orden (m para obtener ayuda):
```

Ahora oprimimos “*w*”, para guardar los cambios.

```
Orden (m para obtener ayuda): w
Se ha modificado la tabla de particiones.
Llamando a ioctl() para volver a leer la tabla de particiones.
Se están sincronizando los discos.
```

Ahora verificamos que la partición se haya creado correctamente, ejecutando el comando **“lsblk”**. En donde podemos observar, que se creó sin contratiempos.

```
root@juan-VirtualBox:/home/juan# lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0       7:0      0    4K  1 loop /snap/bare/5
loop1       7:1      0  63,4M  1 loop /snap/core20/1974
loop2       7:2      0  63,5M  1 loop /snap/core20/2015
loop3       7:3      0  73,9M  1 loop /snap/core22/858
loop4       7:4      0 240,6M  1 loop /snap/firefox/2356
loop5       7:5      0 346,3M  1 loop /snap/gnome-3-38-2004/119
loop6       7:6      0 485,5M  1 loop /snap/gnome-42-2204/126
loop7       7:7      0  91,7M  1 loop /snap/gtk-common-themes/1535
loop8       7:8      0  45,9M  1 loop /snap/snap-store/638
loop9       7:9      0 349,7M  1 loop /snap/gnome-3-38-2004/143
loop10      7:10     0  12,3M  1 loop /snap/snap-store/959
loop11      7:11     0  40,8M  1 loop /snap/snapd/19993
loop12      7:12     0   304K  1 loop /snap/snapd-desktop-integration/49
loop13      7:13     0   53,3M  1 loop /snap/snapd/19457
loop14      7:14     0   452K  1 loop /snap/snapd-desktop-integration/83
sda         8:0      0   25G   0 disk
├─sda1      8:1      0   512M   0 part /boot/efi
└─sda2      8:2      0  24,5G   0 part /var/snap/firefox/common/host-hunspell/
sdb         8:16     0    1G   0 disk
└─sdb2      8:18     0   500M   0 part ←
sr0        11:0     1 1024M   0 rom
```

- **SEPTIMO:** Ahora vamos a formatear la partición para agregarle la **“ext4”**, el cual es uno de los sistemas de archivos más usados y confiables usados en Linux. Esto lo hacemos ejecutando el comando **“mkfs.ext4 /dev/sdb2”**, la parte de **“/dev/sdb2”** se usa para especificar en cual partición haremos el proceso.

```
root@juan-VirtualBox:/home/juan# mkfs.ext4 /dev/sdb2
mke2fs 1.46.5 (30-Dec-2021)
Se está creando un sistema de ficheros con 128000 bloques de 4k y 128000 nodos-i
UID del sistema de ficheros: a13c3f21-0907-4fcd-b471-3c2942920a93
Respalos del superbloque guardados en los bloques:
    32768, 98304

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (4096 bloques): hecho
Escribiendo superbloques y la información contable del sistema de archivos:hech
o
```

Creamos una carpeta para montar la ext4

```
root@juan-VirtualBox:/home/juan# mkdir /mnt/ext4
```

Luego con el comando “*mount*”, montamos el sistema de archivos de la partición.

```
root@juan-VirtualBox:/home/juan# mount /dev/sdb2 /mnt/ext4
```

- **OCTAVO:** Verificamos que la partición se haya modificado correctamente con la ext4.

```
root@juan-VirtualBox:/home/juan# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0        7:0      0    4K  1 loop /snap/bare/5
loop1        7:1      0  63,4M  1 loop /snap/core20/1974
loop2        7:2      0  63,5M  1 loop /snap/core20/2015
loop3        7:3      0  73,9M  1 loop /snap/core22/858
loop4        7:4      0 240,6M  1 loop /snap/firefox/2356
loop5        7:5      0 346,3M  1 loop /snap/gnome-3-38-2004/119
loop6        7:6      0 485,5M  1 loop /snap/gnome-42-2204/126
loop7        7:7      0  91,7M  1 loop /snap/gtk-common-themes/1535
loop8        7:8      0  45,9M  1 loop /snap/snap-store/638
loop9        7:9      0 349,7M  1 loop /snap/gnome-3-38-2004/143
loop10       7:10     0  12,3M  1 loop /snap/snap-store/959
loop11       7:11     0  40,8M  1 loop /snap/snapd/19993
loop12       7:12     0   304K  1 loop /snap/snapd-desktop-integration/49
loop13       7:13     0   53,3M  1 loop /snap/snapd/19457
loop14       7:14     0   452K  1 loop /snap/snapd-desktop-integration/83
sda          8:0      0   25G  0 disk
├─sda1       8:1      0   512M  0 part /boot/efi
└─sda2       8:2      0  24,5G  0 part /var/snap/firefox/common/host-hunspell
sdb          8:16     0    1G  0 disk
└─sdb2       8:18     0   500M  0 part /mnt/ext4 ←
sr0         11:0     1 1024M  0 rom
```