

Parcial uno electiva V

Jhersson Stiven Jojoa Lozano



Institución universitaria EAM

Electiva V

Armenia, Quindio

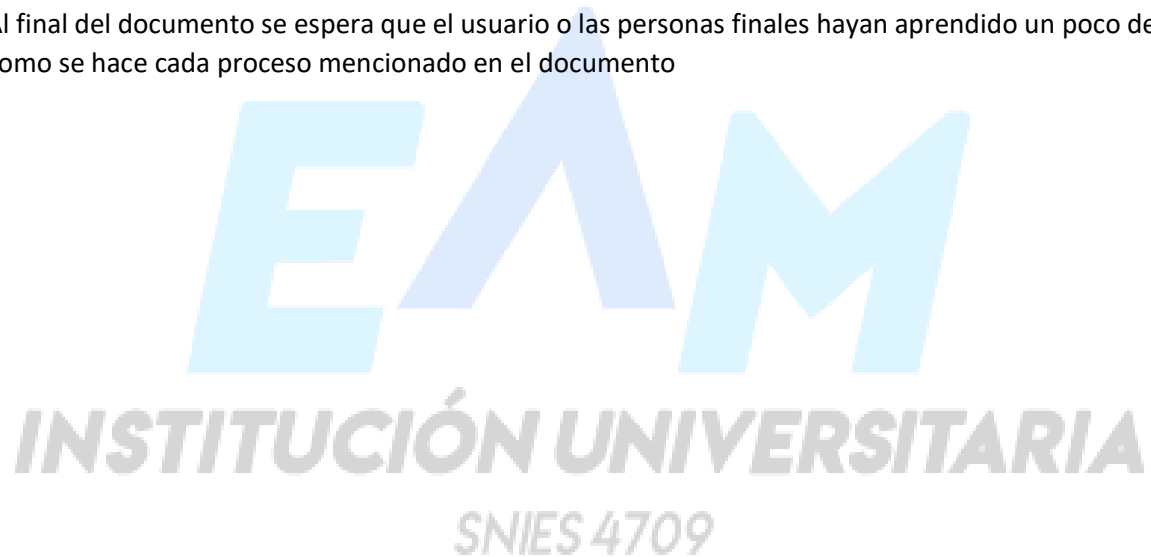
2023

Introducción

Las tecnologías son herramientas y conocimientos aplicados para resolver problemas, mejorar procesos y enriquecer la vida humana. Estas pueden ser tanto físicas como digitales, abarcando una amplia gama de disciplinas y sectores. Desde la invención de la rueda hasta la era de la inteligencia artificial y la robótica, las tecnologías han impulsado el progreso humano y transformado la forma en que vivimos, trabajamos y nos relacionamos. Están en constante evolución, brindando nuevas oportunidades y desafíos a medida que la sociedad avanza en su búsqueda de innovación y mejora continua

En el transcurso de este documento iremos observando como se descarga una herramienta de proxy, balanceador de carga o servidor web llamado nginx, haremos el proceso correcto de descarga configuración, como también iremos mostrando el paso a paso de como particionar un disco dentro del emulador virtual box

Al final del documento se espera que el usuario o las personas finales hayan aprendido un poco de como se hace cada proceso mencionado en el documento





Objetivo General

El objetivo general de este documento es proporcionar una guía rápida de instalación y ejecución de nginx y particionamiento del disco, mediante un emulador (virtual box)

SNIES 4709

Crear un archivo ejecutable mediante bash

Para dar orden al documento seria bueno crear una carpeta donde iremos guardando lo explicado,

- 1) Crearemos una carpeta con el siguiente comando

Mkdir “nombre de la carpeta” ejemplo

```
root@server2:/home/stiven1920/EAM_CNC/taller2# mkdir taller2
```

- 2) Dentro de la carpeta crearemos el ejecutable con funciones sencillas, el siguiente comando seria para crear el archivo

Nano “nombre del archivo”.sh, ejemplo

```
root@server2:/home/stiven1920/EAM_CNC/taller2# nano script.sh
```

Cuando se crea el archivo.sh se vera de la siguiente forma

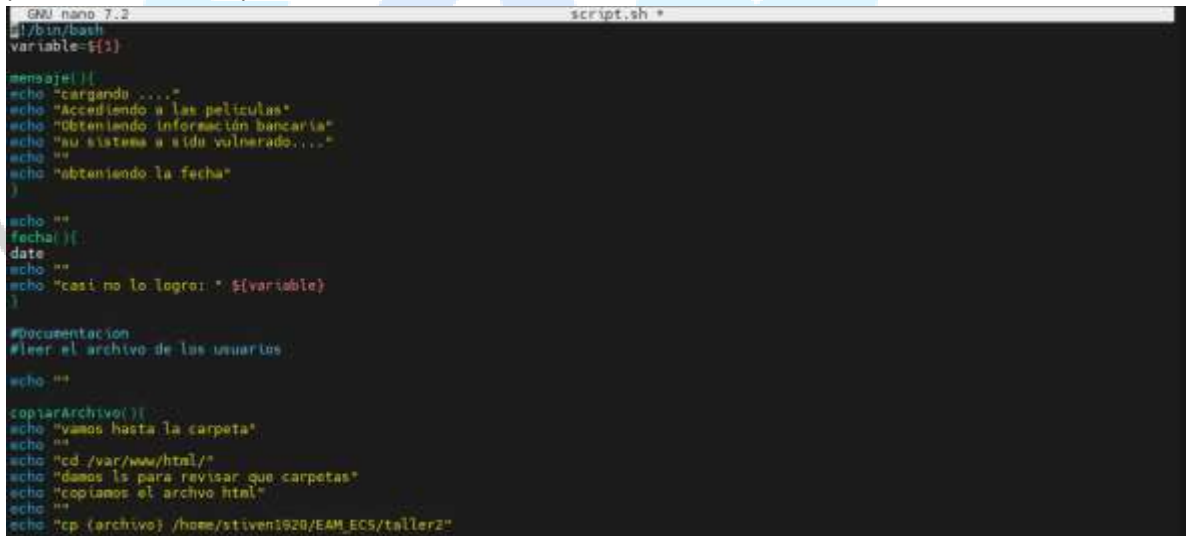


- 3) Dentro del archivo.sh debemos darle forma de un script para que sea ejecutable, debe de hacer de la siguiente forma

i# /bin/bash

A screenshot of a terminal window with a dark background. The title bar at the top reads "GNU nano 7.2" on the left and "script.sh *" on the right. The first line of the script is "i# /bin/bash".

- 4) Ya teniendo la forma del script, aremos algunas funciones básicas
Vemos tres funciones(mensaje, fecha, copiarArchivo) estas funciones son básicas a lo que puede hacer un script

A screenshot of a terminal window with a dark background. The title bar at the top reads "GNU nano 7.2" on the left and "script.sh *" on the right. The script content is as follows:

```
variable=${1}

mensaje(){
  echo "cargando ...."
  echo "Accediendo a las películas"
  echo "Obteniendo información bancaria"
  echo "su sistema a sido vulnerado..."
  echo ""
  echo "obteniendo la fecha"
}

echo ""
fecha(){
  date
  echo ""
  echo "casi no lo logre: " ${variable}
}

#Documentacion
#leer el archivo de los usuarios
echo ""

copiarArchivo(){
  echo "vamos hasta la carpeta"
  echo ""
  echo "cd /var/www/html/"
  echo "damos ls para revisar que carpetas"
  echo "copiamos el archivo html"
  echo ""
  echo "cp (archivo) /home/stiven1920/EAM_ECS/taller2"
```

- 5) Guardaremos el script con la siguiente combinación de teclas, **ctrl+o** y tecleamos **enter**

```
GNU nano 7.2 script.sh
#!/bin/bash
variable=${1}

mensaje(){
echo "cargando ...."
echo "Accediendo a las películas"
echo "Obteniendo información bancaria"
echo "su sistema a sido vulnerado...."
echo ""
echo "obteniendo la fecha"
}

echo ""
fecha(){
date
echo ""
echo "casi no lo logro: * ${variable}"
}

#Documentacion
#Leer el archivo de los usuarios
echo ""

copiarArchivo(){
echo "vamos hasta la carpeta"
echo ""
echo "cd /var/www/html/"
echo "damos la para revisar que carpetas"
echo "copiamos el archivo html"
echo ""
echo "cp (archivo) /home/stiven1920/EAM EC5/taller2"
}

File Name to Write: script.sh
[?] Help [?] DOS Format [?] Append [?] Backup File
[?] Cancel [?] Mac Format [?] Prepend [?] Browse

server2 2% 0.42 GB / 2.55 GB 0.01 MB/s 0.00 MB/s 113 min 18 stiven1920 (c3) 0% Boot: 17%
```

Ya guardado el script vamos a salirnos del archivo script con la siguiente combinación de teclas **ctrl+x** y tecleamos **enter**

- 6) Para ejecutar el script se debe hacer el siguiente comando
Sh "nombre del archivo".sh ó ./" nombre del archivo".sh ejemplo

```
root@server2:/home/stiven1920/EAM_CNC/taller2# sh script.sh
```

```
jue 31 ago 2023 04:11:05 UTC
```

```
casi no lo logro:  
cargando ....  
Accediendo a las peliculas  
Obteniendo información bancaria  
su sistema a sido vulnerado....
```

```
obteniendo la fecha  
vamos hasta la carpeta
```

```
cd /var/www/html/  
damos ls para revisar que carpetas  
copiamos el archivo html
```

```
cp (archivo) /home/stiven1920/EAM_ECS/taller2  
lista los archivos ls  
root@server2:/home/stiven1920/EAM_CNC/taller2#
```

Instalando y ejecutando la herramienta nginx

Primero debemos actualizar los archivos del servidor o máquina virtual

1) Sudo apt-get update, ejemplo

```
root@server2:/home/stiven1920/EAM_CNC/taller2# sudo apt-get update  
Obj:1 http://co.archive.ubuntu.com/ubuntu lunar InRelease  
Des:2 http://co.archive.ubuntu.com/ubuntu lunar-updates InRelease [109 kB]  
Des:3 http://co.archive.ubuntu.com/ubuntu lunar-backports InRelease [99,8 kB]  
Des:4 http://co.archive.ubuntu.com/ubuntu lunar-security InRelease [109 kB]  
Des:5 http://co.archive.ubuntu.com/ubuntu lunar-updates/main amd64 Packages [382 kB]  
Des:6 http://co.archive.ubuntu.com/ubuntu lunar-updates/main Translation-en [95,6 kB]  
Des:7 http://co.archive.ubuntu.com/ubuntu lunar-updates/main amd64 c-n-f Metadata [7.264 B]  
Des:8 http://co.archive.ubuntu.com/ubuntu lunar-updates/restricted amd64 Packages [302 kB]  
Des:9 http://co.archive.ubuntu.com/ubuntu lunar-updates/restricted Translation-en [49,2 kB]  
Des:10 http://co.archive.ubuntu.com/ubuntu lunar-updates/restricted amd64 c-n-f Metadata [504 B]  
Des:11 http://co.archive.ubuntu.com/ubuntu lunar-updates/universe amd64 Packages [759 kB]  
Des:12 http://co.archive.ubuntu.com/ubuntu lunar-updates/universe Translation-en [83,1 kB]  
Des:13 http://co.archive.ubuntu.com/ubuntu lunar-updates/universe amd64 c-n-f Metadata [12,8 kB]  
Des:14 http://co.archive.ubuntu.com/ubuntu lunar-updates/multiverse Translation-en [1.200 B]  
Des:15 http://co.archive.ubuntu.com/ubuntu lunar-backports/universe amd64 Packages [3.940 B]  
Des:16 http://co.archive.ubuntu.com/ubuntu lunar-backports/universe amd64 c-n-f Metadata [172 B]  
Des:17 http://co.archive.ubuntu.com/ubuntu lunar-security/main amd64 Packages [296 kB]  
Des:18 http://co.archive.ubuntu.com/ubuntu lunar-security/main Translation-en [70,0 kB]  
Des:19 http://co.archive.ubuntu.com/ubuntu lunar-security/main amd64 c-n-f Metadata [5.328 B]  
Des:20 http://co.archive.ubuntu.com/ubuntu lunar-security/restricted amd64 Packages [302 kB]  
Des:21 http://co.archive.ubuntu.com/ubuntu lunar-security/restricted Translation-en [49,2 kB]  
Des:22 http://co.archive.ubuntu.com/ubuntu lunar-security/restricted amd64 c-n-f Metadata [508 B]  
Des:23 http://co.archive.ubuntu.com/ubuntu lunar-security/universe amd64 Packages [711 kB]  
Des:24 http://co.archive.ubuntu.com/ubuntu lunar-security/universe Translation-en [64,9 kB]  
Des:25 http://co.archive.ubuntu.com/ubuntu lunar-security/universe amd64 c-n-f Metadata [11,6 kB]  
Descargados 3.526 kB en 3s (1.082 kB/s)
```

Después de tener actualizado la máquina virtual instalaremos el iginx

2) Instalaremos el nginx con el siguiente comando

Sudo apt install nginx

```
root@server2:/home/stiven1920/EAM_CNC/taller2# sudo apt install nginx
```

- 3) Para comprobar si se instalo el **nginx** verificamos con el siguiente comando **systemctl status nginx**, el active debe aparece de la siguiente forma señalada

```
root@server2:/home/stiven1920/EAM_CNC/taller2# sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx ya está en su versión más reciente (1.22.6-1ubuntu3).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 41 no actualizados.
root@server2:/home/stiven1920/EAM_CNC/taller2# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Thu 2023-08-31 02:13:38 UTC; 2h 8min ago
     Docs: man:nginx(8)
    Main PID: 713 (nginx)
      Tasks: 2 (limit: 2969)
     Memory: 13.3M
        CPU: 357ms
    CGroup: /system.slice/nginx.service
            └─713 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              └─723 "nginx: worker process"

ago 31 02:13:31 server2 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
ago 31 02:13:38 server2 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
root@server2:/home/stiven1920/EAM_CNC/taller2#
```

- 4) Listamos los sitios disponibles que tenemos en el momento. Con el siguiente comando lo podemos ver

ls /etc/nginx/sites-available

```
root@server2:/home/stiven1920# ls /etc/nginx/sites-available
default
root@server2:/home/stiven1920#
```

- 5) Listamos los sitios activos por el nginx con el siguiente comando

ls /etc/nginx/sites-enabled

```
root@server2:/home/stiven1920# ls /etc/nginx/sites-enabled
default
root@server2:/home/stiven1920#
```

- 6) Ahora cambiaremos la bienvenida del nginx en la siguiente ruta

Cd /var/www/html/

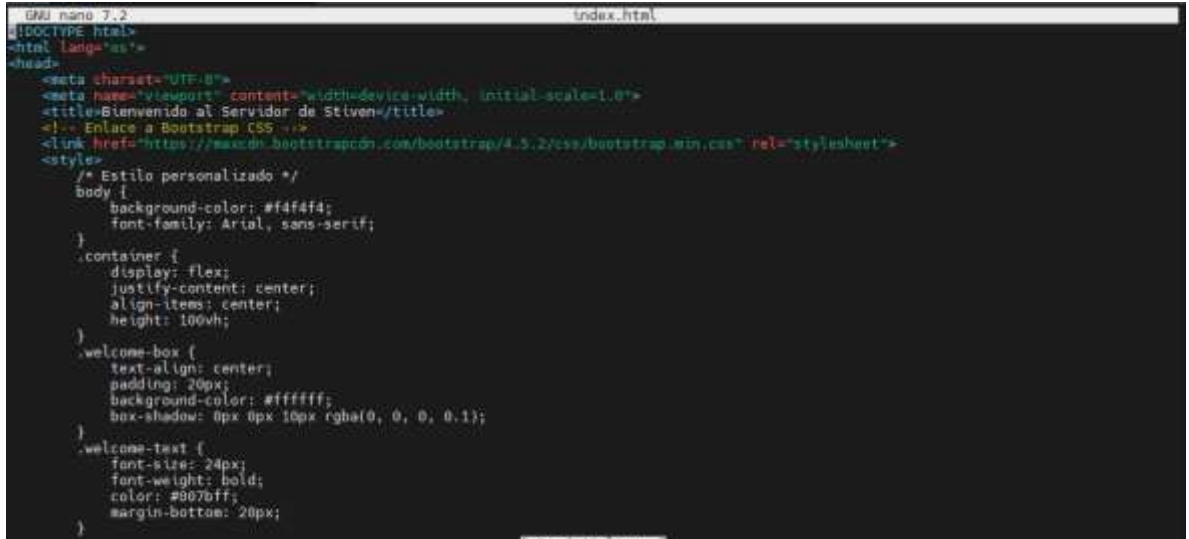
luego listamos lo que hay en esa carpeta con **ls**

```
root@server2:/home/stiven1920# cd /var/www/html/
root@server2:/var/www/html# ls
index.html
root@server2:/var/www/html#
```


- 7) Ahora con el comando **nano** abrimos el archivo index.html de la siguiente forma
nano index.html

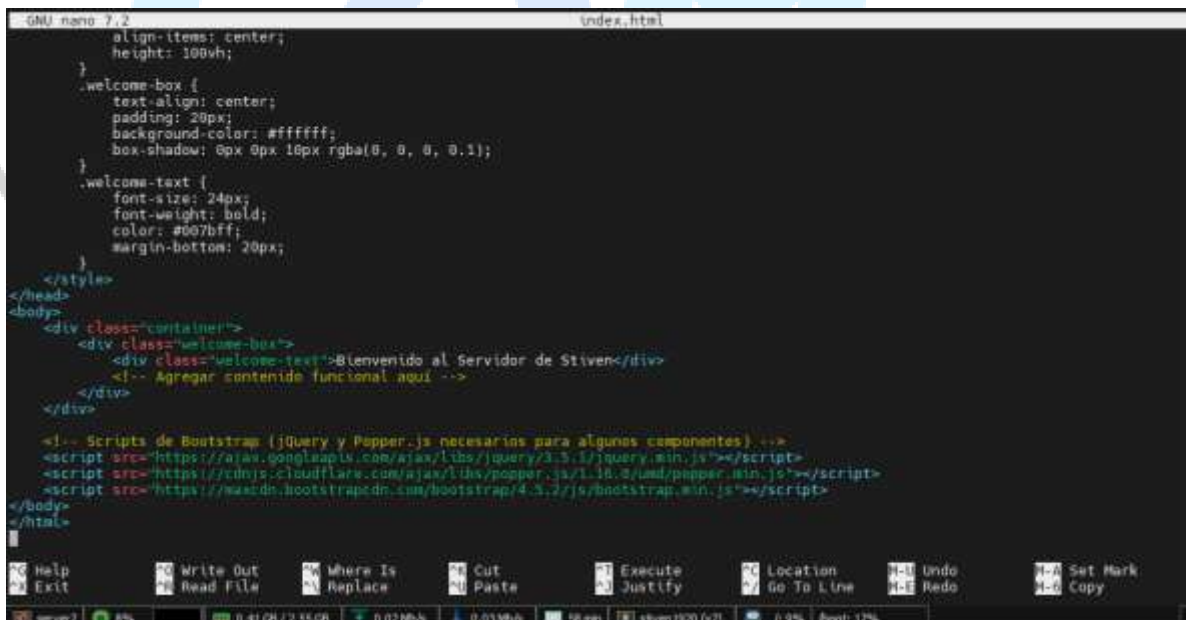
```
root@server2:/var/www/html# nano index.html
```

Nos deberá aparecer algo parecido a esta información.



```
GNU nano 7.2 index.html
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bienvenido al Servidor de Steven</title>
  <!-- Enlace a Bootstrap CSS -->
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
  <style>
    /* Estilo personalizado */
    body {
      background-color: #f4f4f4;
      font-family: Arial, sans-serif;
    }
    .container {
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .welcome-box {
      text-align: center;
      padding: 20px;
      background-color: #ffffff;
      box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
    }
    .welcome-text {
      font-size: 24px;
      font-weight: bold;
      color: #007bff;
      margin-bottom: 20px;
    }
  </style>
</head>
<body>
```

- 8) Ya estando dentro del html podemos modificar lo que queramos, respetando el modelo del html



```
GNU nano 7.2 index.html
    align-items: center;
    height: 100vh;
  }
  .welcome-box {
    text-align: center;
    padding: 20px;
    background-color: #ffffff;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
  }
  .welcome-text {
    font-size: 24px;
    font-weight: bold;
    color: #007bff;
    margin-bottom: 20px;
  }
</style>
</head>
<body>
  <div class="container">
    <div class="welcome-box">
      <div class="welcome-text">Bienvenido al Servidor de Steven</div>
      <!-- Agregar contenido funcional aquí -->
    </div>
  </div>

  <!-- Scripts de Bootstrap (jQuery y Popper.js necesarios para algunos componentes) -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

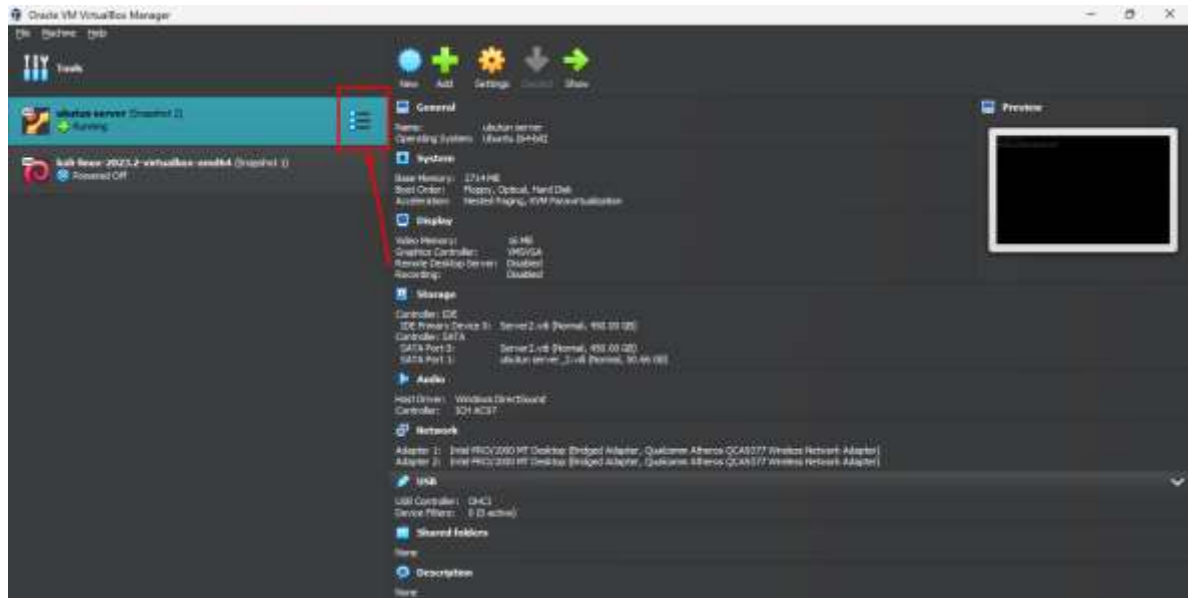
Luego de haber modificado el archivo ejecutamos la combinación de teclas **ctrl+o** para guarda y por último **ctrl+x** para salir

Luego se dirige al navegador de su preferencia y escribe localhost y le deberá aparecer su contenido modificado

Crear particiones de disco en el virtual box

Primero crearemos una copia de seguridad del entorno virtual

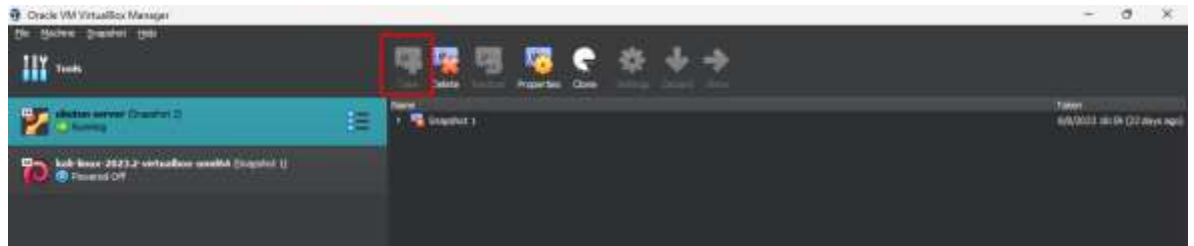
1) Daremos click en la parte seleccionada



2) Damos click en snapshots



3) Le damos click en la parte que dice take



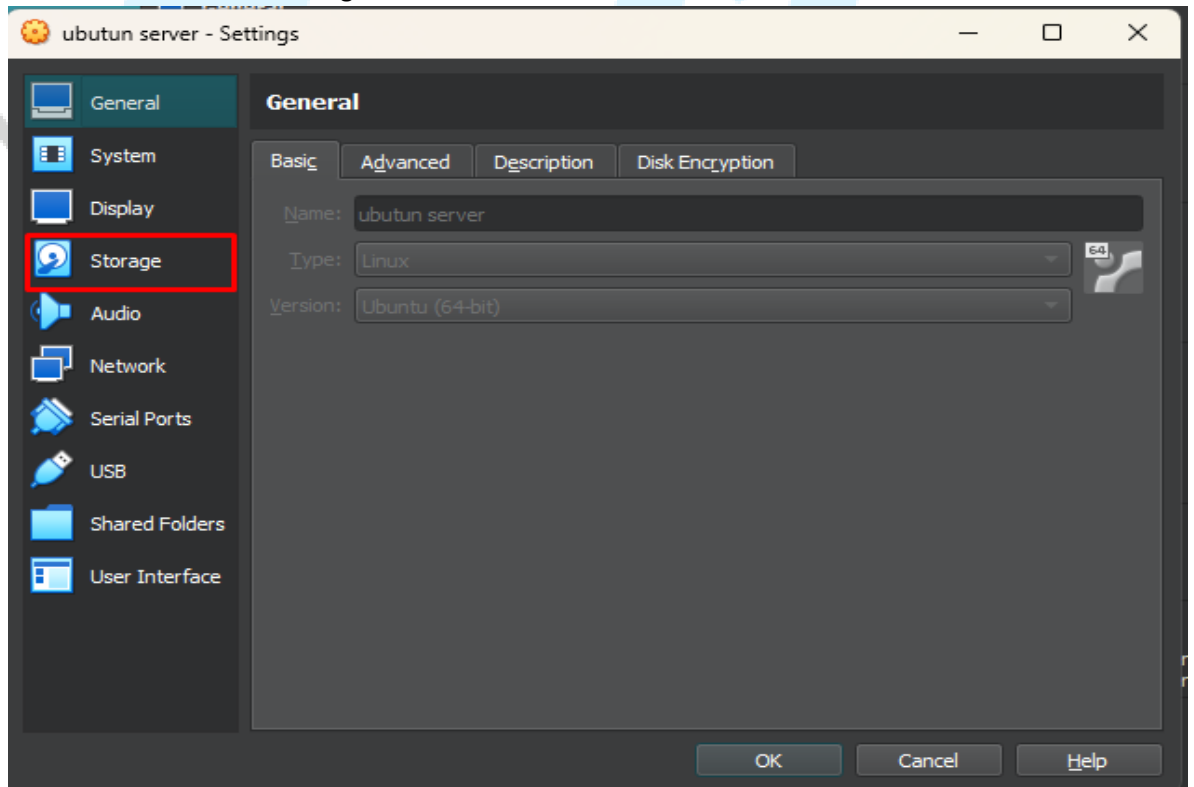
Después le saldrá un cuadro de dialogo donde le podrá poner nombre a su copia de seguridad de la máquina virtual, después le da guardar.

Después de hacer esto vamos a configurar el disco para poderlo particionar

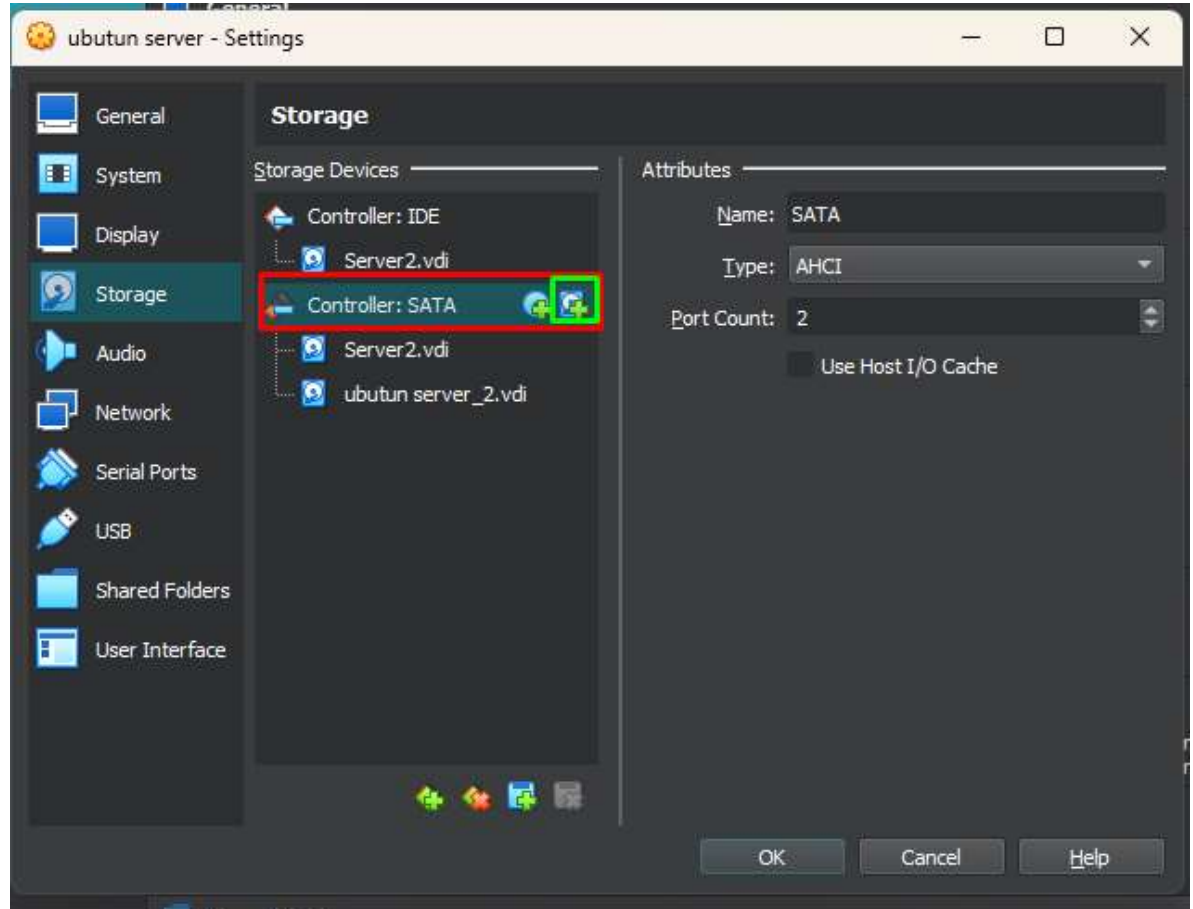
- 4) Daremos click en setting



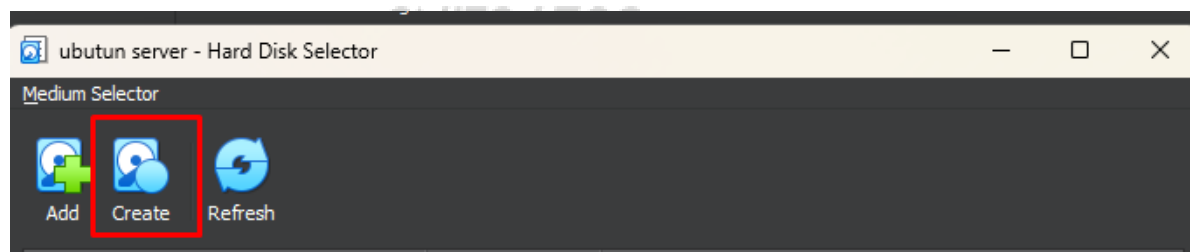
- 5) Vamos a darle click en storage



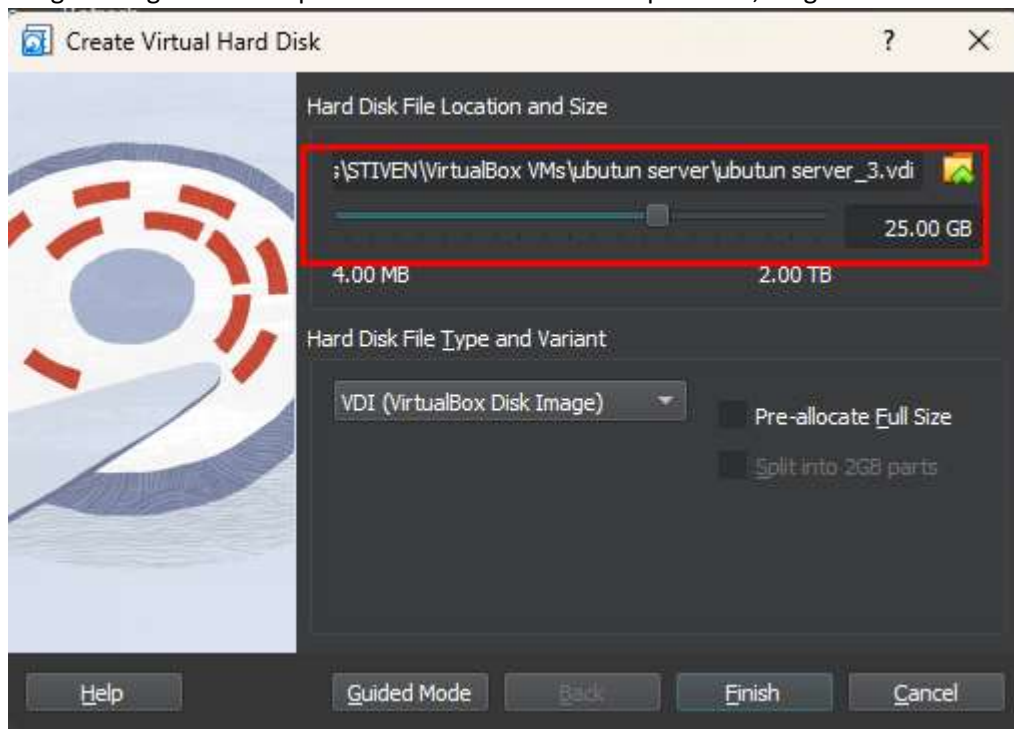
- 6) Luego damos seleccionamos lo que esta en rojo y por ultimo damos click en el icono de mas color verde



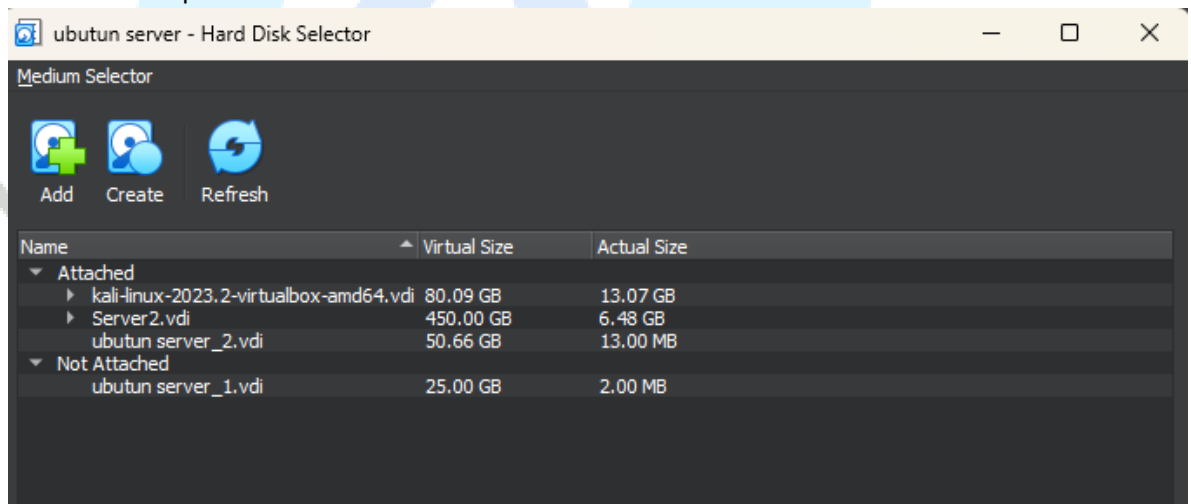
- 7) Luego damos click en create disco, donde este seleccionado color rojo



8) Luego le asignaremos espacio de memoria a la nueva partición, luego finalizamos



9) Aca vemos las particiones o los discos creados en el momento



10) Luego de haber hecho esta configuración inciamos la maquina virtual



- 11) Luego entramos a al terminal para verificar la creación del disco, para verlo ejecutamos el siguiente comando.

Sudo su lsblk

```
stiven1920@server2:~$ sudo su lsblk
[sudo] password for stiven1920: 
root@server2:/home/stiven1920# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
loop0                               7:0      0   73,9M  1 loop /snap/core22/858
loop1                               7:1      0   63,5M  1 loop /snap/core20/2015
loop2                               7:2      0  173,5M  1 loop /snap/lxd/25112
loop3                               7:3      0   53,3M  1 loop /snap/snapd/19457
loop4                               7:4      0    71M  1 loop /snap/prometheus/86
loop5                               7:5      0   68,8M  1 loop /snap/powershell/244
loop6                               7:6      0   63,4M  1 loop /snap/core20/1974
loop7                               7:7      0  128,9M  1 loop /snap/docker/2893
loop8                               7:8      0  175,5M  1 loop /snap/lxd/25505
loop9                               7:9      0   40,8M  1 loop /snap/snapd/19993
loop10                              7:10     0   73,9M  1 loop /snap/core22/864
loop11                              7:11     0   68,8M  1 loop /snap/powershell/242
sda                                 8:0      0  450G   0 disk
├─sda1                              8:1      0    1M   0 part
├─sda2                              8:2      0    2G   0 part /boot
├─sda3                              8:3      0  448G   0 part
│   └─ubuntu--vg-ubuntu--lv 253:0   0  100G   0 lvm  /
sdb                                 8:16     0  450G   0 disk
├─sdb1                              8:17     0    1M   0 part
├─sdb2                              8:18     0    2G   0 part
└─sdb3                              8:19     0  448G   0 part
sdc                                 8:32     0  50,7G   0 disk
└─sdc1                              8:33     0    1,9G   0 part
root@server2:/home/stiven1920#
```

- 12) Luego de tener claro el disco que vamos a particionar hacemos el siguiente comando para comenzar a particionar

Fdisk /dev/sdc

```
root@server2:/home/stiven1920# fdisk /dev/sdc
Welcome to fdisk (util-linux 2.38.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

- 13) Vemos las ayudas para tener claro que es lo que vamos hacer

```
Command (m for help): m
```

```
DOS (MBR)
a  toggle a bootable flag
b  edit nested BSD disklabel
c  toggle the dos compatibility flag

Generic
d  delete a partition
F  list free unpartitioned space
l  list known partition types
n  add a new partition
p  print the partition table
t  change a partition type
v  verify the partition table
i  print information about a partition

Misc
m  print this menu
u  change display/entry units
x  extra functionality (experts only)

Script
I  load disk layout from sfdisk script file
O  dump disk layout to sfdisk script file

Save & Exit
w  write table to disk and exit
q  quit without saving changes

Create a new label
g  create a new empty GPT partition table
G  create a new empty SGI (IRIX) partition table
o  create a new empty MBR (DOS) partition table
s  create a new empty Sun partition table
```

- 14) Luego damos en la opción **n** para crear una nueva partición

```
Command (m for help): n
Partition type
  p  primary (1 primary, 0 extended, 3 free)
  e  extended (container for logical partitions)
Select (default p):
```

- 15) Luego le añadimos la cantidad de memoria que deseamos en el momento

```
Select (default p): p
Partition number (2-4, default 2): 2
First sector (3907584-106231871, default 3907584): 3907584
```

- 16) Después de haberle añadido la cantidad de memoria tecleamos la letra **w** para guardar

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (3907584-106231871, default 106231871): 106231871
Created a new partition 2 of type 'Linux' and of size 48,8 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```


- 17) Con el comando lsblk listamos los discos montados y verificamos el que acabamos de agregar

```
root@server2:/home/stiven1920# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0                               7:0      0   73,9M  1 loop /snap/core22/858
loop1                               7:1      0   63,5M  1 loop /snap/core20/2015
loop2                               7:2      0  173,5M  1 loop /snap/lxd/25112
loop3                               7:3      0   53,3M  1 loop /snap/snapd/19457
loop4                               7:4      0    71M  1 loop /snap/prometheus/86
loop5                               7:5      0   68,8M  1 loop /snap/powershell/244
loop6                               7:6      0   63,4M  1 loop /snap/core20/1974
loop7                               7:7      0  128,9M  1 loop /snap/docker/2893
loop8                               7:8      0  175,5M  1 loop /snap/lxd/25505
loop9                               7:9      0   40,8M  1 loop /snap/snapd/19993
loop10                              7:10     0   73,9M  1 loop /snap/core22/864
loop11                              7:11     0   68,8M  1 loop /snap/powershell/242
sda                                 8:0      0   450G  0 disk
├─sda1                              8:1      0    1M  0 part
├─sda2                              8:2      0    2G  0 part /boot
├─sda3                              8:3      0   448G  0 part
│   └─ubuntu--vg-ubuntu--lv 253:0  0   100G  0 lvm  /
sdb                                 8:16     0   450G  0 disk
├─sdb1                              8:17     0    1M  0 part
├─sdb2                              8:18     0    2G  0 part
├─sdb3                              8:19     0   448G  0 part
sdc                                 8:32     0   50,7G  0 disk
├─sdc1                              8:33     0    1,9G  0 part
└─sdc2                              8:34     0   48,8G  0 part
root@server2:/home/stiven1920#
```

- 18) Luego formateamos el disco y hacemos el siguiente comando para agregar el ext4
Mkfs.ext4 /dev/sdc2

```
root@server2:/home/stiven1920# mkfs.ext4 /dev/sdc2
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 12790536 4k blocks and 3203072 inodes
Filesystem UUID: 493ce9a9-b55f-44c0-bce4-4a01555e45e3
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (65536 blocks): done
Writing superblocks and filesystem accounting information: done

root@server2:/home/stiven1920#
```

- 19) Creamos la carpeta para poner en el disco con el siguiente comando

Mkdir /mnt/ext3

```
root@server2:/home/stiven1920# mkdir /mnt/ext3
root@server2:/home/stiven1920#
```

- 18) luego montamos en el disco la carpeta creada


```

root@server2:/home/stiven1920# mount /dev/sdc2 /mnt/ext3
root@server2:/home/stiven1920# lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0                              7:0      0    73,9M 1 loop /snap/core22/858
loop1                              7:1      0    63,5M 1 loop /snap/core20/2015
loop2                              7:2      0   173,5M 1 loop /snap/lxd/25112
loop3                              7:3      0    53,3M 1 loop /snap/snapd/19457
loop4                              7:4      0     71M 1 loop /snap/prometheus/86
loop5                              7:5      0    68,8M 1 loop /snap/powershell/244
loop6                              7:6      0    63,4M 1 loop /snap/core20/1974
loop7                              7:7      0   128,9M 1 loop /snap/docker/2893
loop8                              7:8      0   175,5M 1 loop /snap/lxd/25505
loop9                              7:9      0    40,8M 1 loop /snap/snapd/19993
loop10                             7:10     0    73,9M 1 loop /snap/core22/864
loop11                             7:11     0    68,8M 1 loop /snap/powershell/242
sda                                8:0      0   450G  0 disk
├─sda1                             8:1      0      1M  0 part
├─sda2                             8:2      0      2G  0 part /boot
├─sda3                             8:3      0   448G  0 part
└─ubuntu--vg-ubuntu--lv 253:0     0   100G  0 lvm  /
sdb                                8:16     0   450G  0 disk
├─sdb1                             8:17     0      1M  0 part
├─sdb2                             8:18     0      2G  0 part
├─sdb3                             8:19     0   448G  0 part
sdc                                8:32     0   50,7G  0 disk
├─sdc1                             8:33     0    1,9G  0 part
└─sdc2                             8:34     0   48,8G  0 part /mnt/ext3
root@server2:/home/stiven1920#

```

20) Con el comando umount y la dirección del disco podemos desmontar todo, ejemplo

```

root@server2:/home/stiven1920# umount /dev/sdc2
root@server2:/home/stiven1920# lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0                              7:0      0    73,9M 1 loop /snap/core22/858
loop1                              7:1      0    63,5M 1 loop /snap/core20/2015
loop2                              7:2      0   173,5M 1 loop /snap/lxd/25112
loop3                              7:3      0    53,3M 1 loop /snap/snapd/19457
loop4                              7:4      0     71M 1 loop /snap/prometheus/86
loop5                              7:5      0    68,8M 1 loop /snap/powershell/244
loop6                              7:6      0    63,4M 1 loop /snap/core20/1974
loop7                              7:7      0   128,9M 1 loop /snap/docker/2893
loop8                              7:8      0   175,5M 1 loop /snap/lxd/25505
loop9                              7:9      0    40,8M 1 loop /snap/snapd/19993
loop10                             7:10     0    73,9M 1 loop /snap/core22/864
loop11                             7:11     0    68,8M 1 loop /snap/powershell/242
sda                                8:0      0   450G  0 disk
├─sda1                             8:1      0      1M  0 part
├─sda2                             8:2      0      2G  0 part /boot
├─sda3                             8:3      0   448G  0 part
└─ubuntu--vg-ubuntu--lv 253:0     0   100G  0 lvm  /
sdb                                8:16     0   450G  0 disk
├─sdb1                             8:17     0      1M  0 part
├─sdb2                             8:18     0      2G  0 part
├─sdb3                             8:19     0   448G  0 part
sdc                                8:32     0   50,7G  0 disk
├─sdc1                             8:33     0    1,9G  0 part
└─sdc2                             8:34     0   48,8G  0 part
root@server2:/home/stiven1920#

```

Recopilación de comando Linux

ls listar los archivos

ls -l: mostrar los archivos en forma de lista pero con los permisos |

d: directorio | carpeta

/: archivos

clear: limpiar terminal

cd: movernos entre carpetas

cat: leer archivos | mostrar archivos

yum update: actualizar los paquetes debian

apt-get: actualizar los paquetes de linux

-lrt: ejecuta de forma ascendente

Cat: mostrar un archivo

vi (nombre del archivo): crea un archivo con vi

:wq! escribir y salir

:q! salir del archivo

chmod u-r nombreArchivo: quita los permisos de lectura al usuario

chown: agregar un usuario al sistema

id nombreUsuario: verificar los grupos de un usuario.

touch: crear un archivo vacío

addUser: agrega un usuario

chown usuario:usuario archivo: le asigna un archivo a un usuario específico

sudo su - test: cambiar el usuario

uptime: revisa el tiempo del servidor

kill -9 identificador del proceso: mata los procesos por el id

ctrl + c: mata un proceso en ejecución

ctrl + z: pone el proceso en segundo plano

jobs: revisa los procesos en segundo plano

fg nombre | identificador: pone a trabajar los procesos de segundo plano

more archivo

less archivo

tail -f head

systemctl { enable, start, stop, restart, status } gnixn

fdisk: permite tener 4 particiones y una extendida

gdisk: 128 primarias y el resto extendidas.

partprobe -s: actualiza las particiones del grupo

umount ruta: desmonta un disco

