



CURSO 2024/25

ADMINISTRACION DE SISTEMAS Y REDES

PRACTICA 2

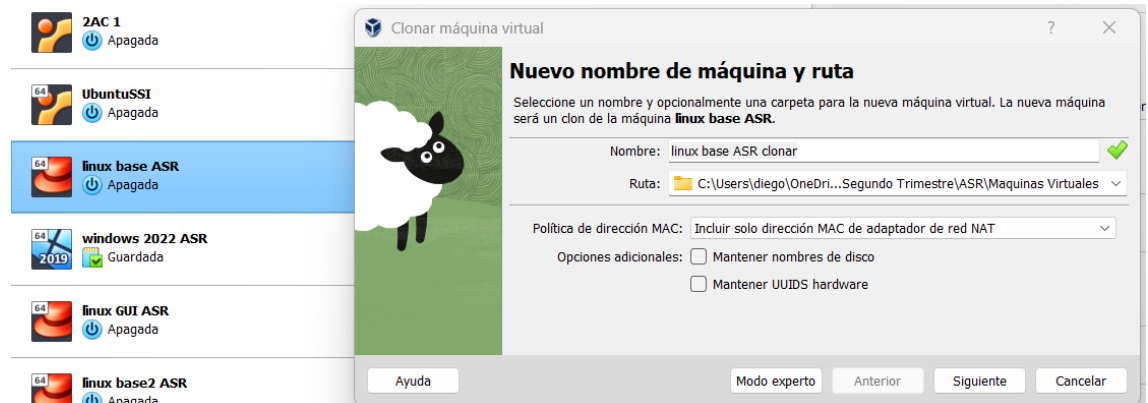
DIEGO GARCÍA GONZÁLEZ



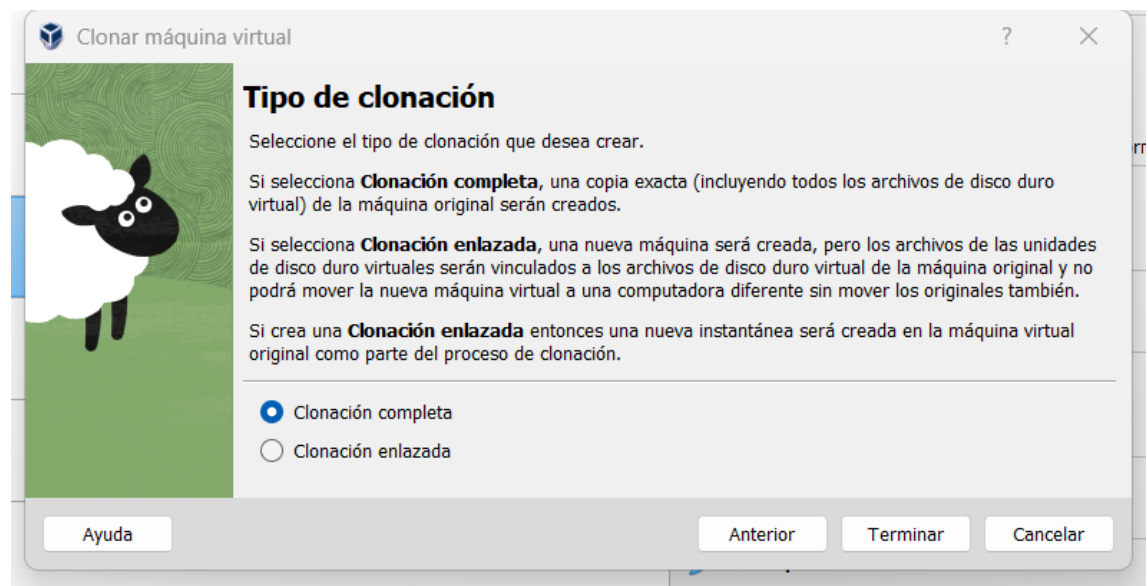
Trabajando con máquinas virtuales y discos

A. Replicación y traslado de máquinas virtuales

Lo primero de todo que se nos pide es la clonación de la maquina virtual, para poder modificarla sin alterar el prototipo inicial. Para ello, con la maquina apagada hacemos click derecho sobre ella, y pulsamos la opción de clonar:



Le damos a siguiente y por ultimo a terminar.

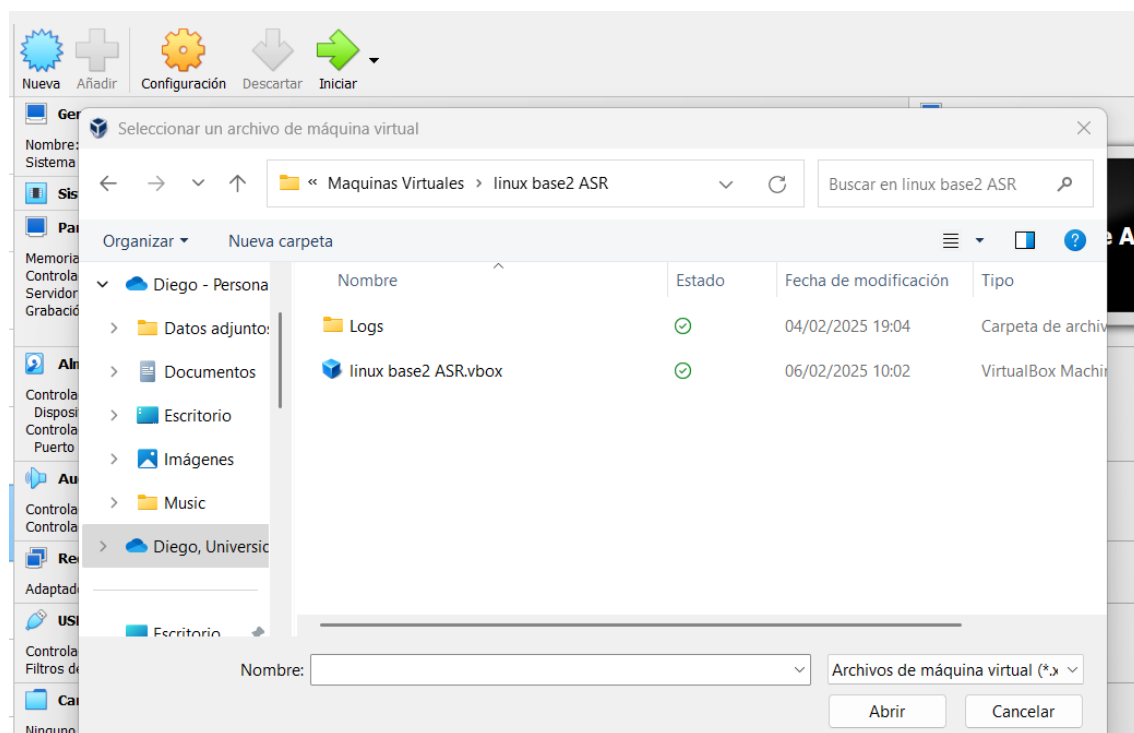


Con ello tendremos la maquina virtual clonada y lista para la realización de esta práctica o cualquier otra.

También, en vez de darle click derecho sobre la maquina apagada, podemos entrar en el apartado de instantáneas y elegir la opción clonar:

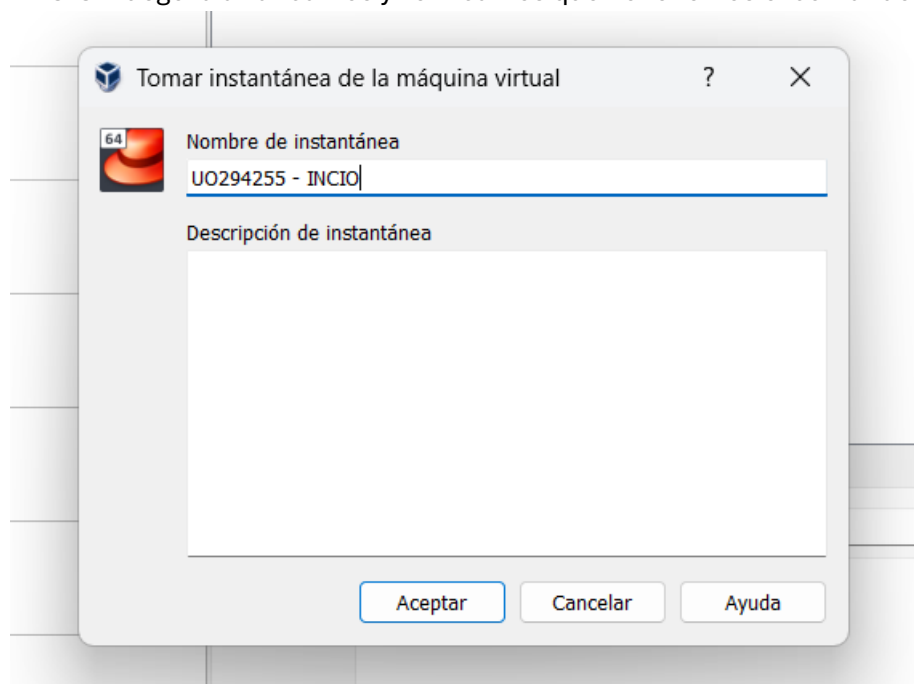


También una alternativa a la clonación es mediante exportado e importado de una máquina. Al ser mismo software y hardware, podemos desde el directorio donde tenemos almacenadas las máquinas virtuales, importar la misma maquina nuevamente para usarla. Esto es a través del menú de VirtualBox, le damos a añadir, y elegiremos en la carpeta de la máquina, el archivo “.vbox”, y con eso estaría:



B. Instantáneas

En primer lugar, tomamos la instantánea de la máquina, llamándola por ejemplo INICIO. Luego la arrancamos y verificamos que no tenemos el comando nslookup



```

[~]# nslookup www.google.com
bash: nslookup: orden no encontrada
[~]# dnf whatprovides nslookup
bind-utils-32:9.16.23-24.el9_5.x86_64 : Utilities for querying DNS name servers
Repositorio : appstream
Resultado de:
Archivo : /usr/bin/nslookup

bind9.18-utils-32:9.18.29-1.el9.x86_64 : Utilities for querying DNS name servers
Repositorio : appstream
Resultado de:
Archivo : /usr/bin/nslookup

[~]# sudo dnf install bind-utils -y_

[~]# nslookup www.google.com
Server:      77.26.11.233
Address:     77.26.11.233#53

Non-authoritative answer:
Name:   www.google.com
Address: 216.58.209.68
Name:   www.google.com
Address: 2a00:1450:4003:801::2004

[~]# _

```

Ahora cargaremos la maquina de nuevo al punto donde tomamos la instantánea. Lo primero es apagar la maquina y nos vamos al menú de instantáneas de Virtual Box:



Ahora volvemos a arrancar la maquina para verificar que se han eliminado los cambios, volviendo a ejecutar nslookup:

```

[~]# nslookup www.google.com
-bash: nslookup: orden no encontrada
[~]# _

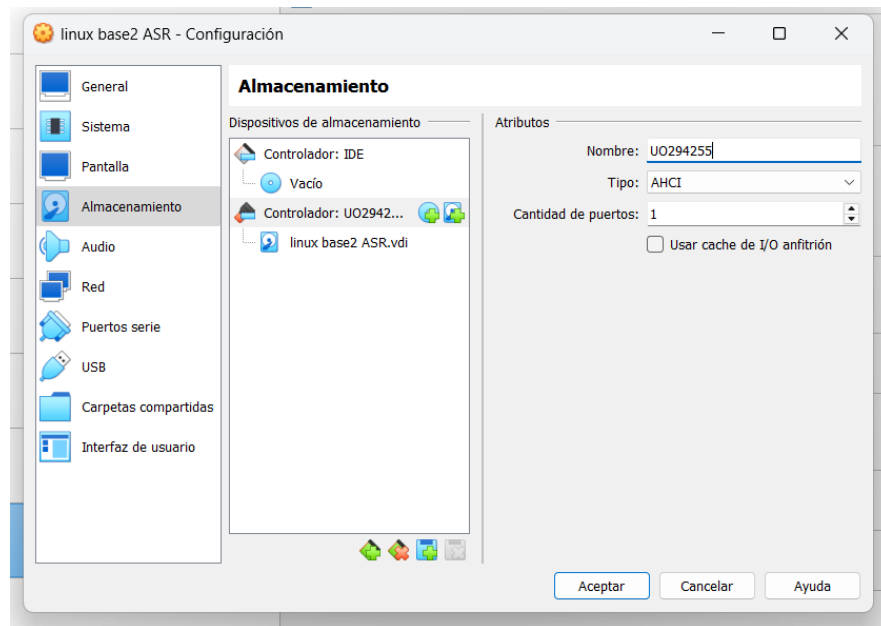
```

C. Añadir un nuevo disco a las MVs

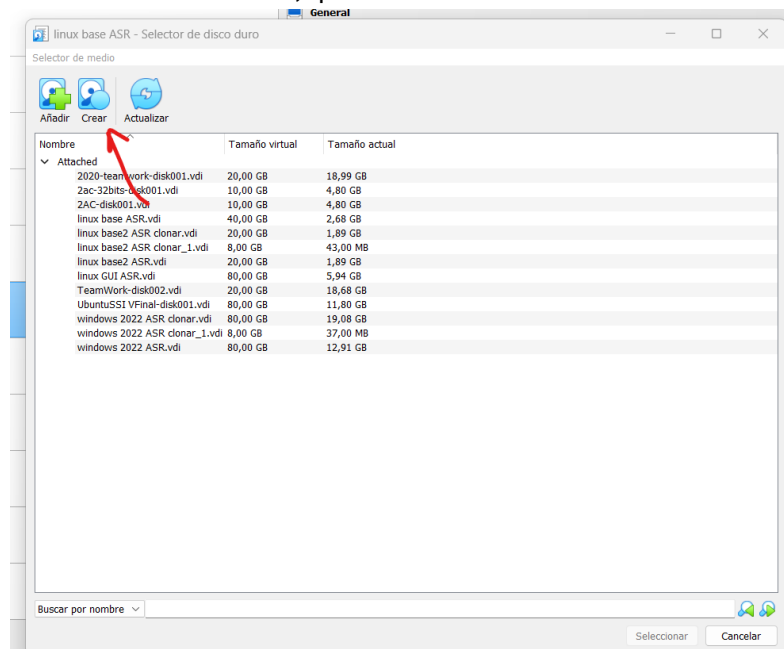
El proceso de añadir un nuevo disco es ligeramente distinto entre Windows y Linux:

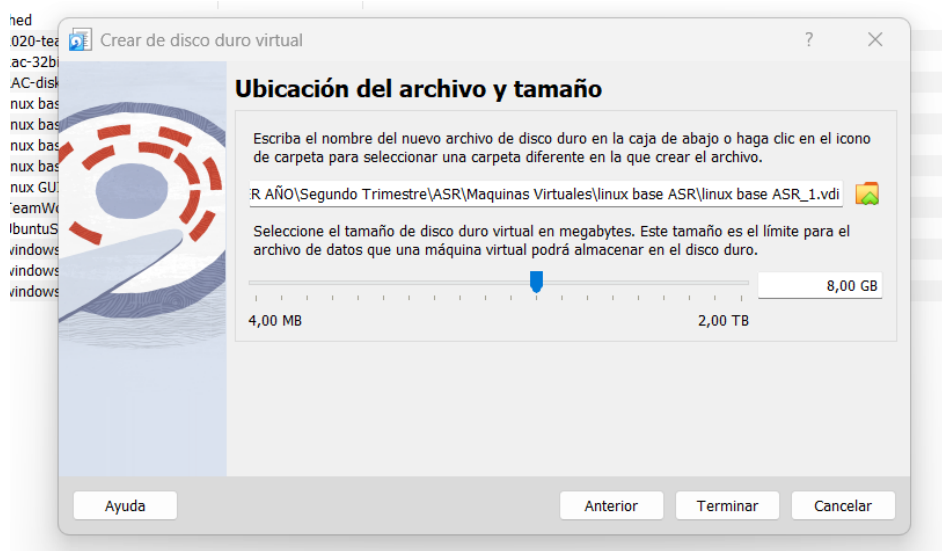
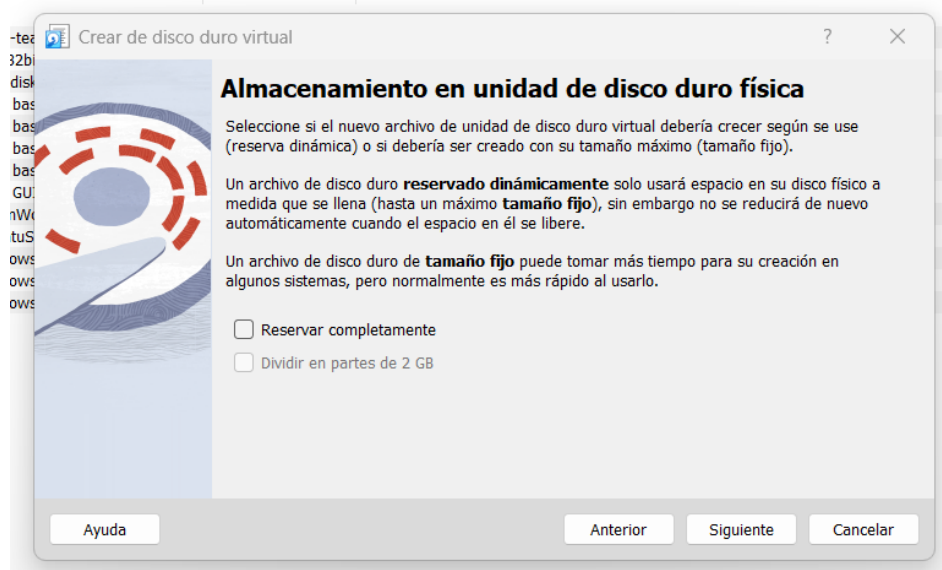
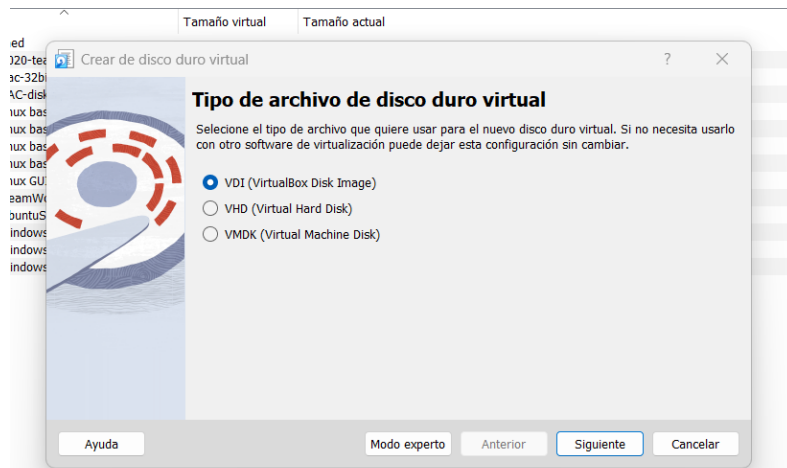
- Adición de un segundo disco a un sistema Linux ya instalado
Para añadir un disco al sistema Linux, el primer paso es desde la misma pestaña de inicio de virtualbox, entrar a la configuración de la maquina a la cual añadiremos el disco. Una vez en la configuración, iremos al apartado

de almacenamiento donde añadiremos un nuevo disco, sobre el controlador SATA.



Una vez hagamos click en la opción, nos saldrá un menú con todos los discos disponibles, nosotros crearemos uno nuevo, modificando únicamente el tamaño, que será de 8GB:





Y pulsando sobre terminar, ya habremos instalado el disco, aunque no está configurado.

Ahora encendemos la máquina virtual, iniciamos sesión como “root” y proseguimos con la configuración.

Una vez iniciada la sesión, podremos ejecutar “ls /dev/sd*” para ver los discos y particiones actuales.:

```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
[\U0294255linux ~]# ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sdb
[\U0294255linux ~]#
```

Como podemos ver, los discos actuales son el “sda”, que es el disco del sistema con 3 particiones (“sda1”, “sda2” y “sda3”), y el disco “sdb” que es el disco que recién acabamos de instalar.

Si usamos la instrucción “gdisk /dev/sda” podremos ver la configuración del disco del sistema:

```
[\U0294255linux ~]# gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help):
```

En ello podemos apreciar la tabla de particiones, junto a la especificación de que usa el tipo de tabla de particiones GPT (la nueva).

Ahora ejecutaremos el mismo comando para inspeccionar el disco nuevo:

```
[\U0294255linux ~]# gdisk /dev/sdb
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.

Command (? for help): _
```

Debido a que esta sin configurar, podemos ver que no hay tabla de particiones creadas, puesto a que no existe ninguna partición, pero hay solución para ello. Usamos la opción “n” para crear una nueva partición, de 3GB, luego veremos cómo se ha creado usando la opción “p”:

```

[U0294255]linux ~]# gdisk /dev/sdb
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.

Command (? for help): n
Partition number (1-128, default 1):
First sector (34-16777182, default = 2048) or {+-}size{KMGT}:
Last sector (2048-16777182, default = 16777182) or {+-}size{KMGT}: +512MB
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

Command (? for help): p
Disk /dev/sdb: 16777216 sectors, 8.0 GiB
Model: VBOX HARDDISK
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): CA77C3A2-CD10-4982-807B-855505C174B4
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 16777182
Partitions will be aligned on 2048-sector boundaries
Total free space is 15728573 sectors (7.5 GiB)

Number  Start (sector)    End (sector)  Size      Code  Name
   1            2048         1050623   512.0 MiB   8300   Linux filesystem

Command (? for help): _

```

Una vez creada esa partición, crearemos una nueva partición de 3GB y otra que ocupe el resto del espacio libre del disco:

```

Command (? for help): p
Disk /dev/sdb: 16777216 sectors, 8.0 GiB
Model: VBOX HARDDISK
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 6DF121BB-51EB-4F7A-A079-5C0EE5906E8D
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 16777182
Partitions will be aligned on 2048-sector boundaries
Total free space is 2014 sectors (1007.0 KiB)

Number  Start (sector)    End (sector)  Size      Code  Name
   1            2048         1050623   512.0 MiB   8300   Linux filesystem
   2          1050624         7342079    3.0 GiB    8300   Linux filesystem
   3          7342080         16777182    4.5 GiB    8300   Linux filesystem

Command (? for help):

```

Ahora que tenemos las particiones creadas, usaremos la opción “w” para que guarde estos cambios en el disco, y podremos salir de gdisk con la opción “q” sin que se pierdan las modificaciones.

```

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/sdb.
[ 195.166464] sdb: sdb1 sdb2 sdb3
[ 196.268871] sdb: sdb1 sdb2 sdb3
The operation has completed successfully.
[U0294255]linux ~]#

```


Ahora volveremos a usar “ls /dev/sd*” para ver los resultados:

```
[U0294255]linux ~]# ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3 /dev/sdb /dev/sdb1 /dev/sdb2 /dev/sdb3
[U0294255]linux ~]# ls /dev/sdb*
/dev/sdb /dev/sdb1 /dev/sdb2 /dev/sdb3
[U0294255]linux ~]# _
```

También ejecutaremos el comando “parted /dev/sdb unit MB print” para mostrar la tabla de particiones del disco que creamos:

```
[U0294255]linux ~]# parted /dev/sdb unit MB print
Modelo: ATA VBOX HARDISK (scsi)
Disco /dev/sdb: 8590MB
Tamaño de sector (lógico/físico): 512B/512B
Tabla de particiones: gpt
Banderas de disco:

Número  Inicio  Fin      Tamaño  Sistema de ficheros  Nombre              Banderas
1        1,05MB  538MB    537MB    Linux filesystem     Linux filesystem
2        538MB   3759MB   3221MB    Linux filesystem     Linux filesystem
3        3759MB  8590MB   4831MB    Linux filesystem     Linux filesystem
```

Ahora que ya hemos particionado el disco, vamos a crear un sistema de archivos para las particiones.

Lo primero a hacer será crear sistema de archivos ext2 en la primera partición:

```
[U0294255]linux ~]# mkfs.ext2 /dev/sdb1
mke2fs 1.46.5 (30-Dec-2021)
Se está creando un sistema de ficheros con 131072 bloques de 4k y 32768 nodos-i
UUID del sistema de ficheros: 5bafc4fe-d0fb-4058-ab8a-b2ad1f0fe174
Respalos del superbloque guardados en los bloques:
    32768, 98304

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho
```

Ahora una vez creado su sistema, lo nombraremos como “disco2a”:

```
[U0294255]linux ~]# e2label /dev/sdb1 disco2a
```

Ahora, se nos pide crear el archivo de journal de la primera partición y hacer que el sistema de ficheros sea de tipo ext3. Para ello ejecutaremos la instrucción siguiente “tune2fs -j /dev/sdb1”:

```
[U0294255]linux ~]# tune2fs -j /dev/sdb1
tune2fs 1.46.5 (30-Dec-2021)
Creando el nodo-i del fichero de transacciones: hecho
```

Ahora crearemos un sistema de archivos **xfs** en la segunda partición, poniéndole como etiqueta disco2b con el comando **xfs_admin**.

Para ello lo primero será crear el sistema con la instrucción de antes de esta manera “**mkfs.xfs /dev/sdb2**”, para posteriormente modificar la etiqueta con “**xfs_admin -L disco2b /dev/sdb2**”:

```
[U0294255]linux ~]# mkfs.xfs /dev/sdb2
meta-data=/dev/sdb2             isize=512    agcount=4, agsize=196608 blks
=                               sectsz=512    attr=2, projid32bit=1
=                               crc=1        finobt=1, sparse=1, rmapbt=0
=                               reflink=1    bigtime=1 inobtcount=1 nrext64=0
data      =                     bsize=4096    blocks=786432, imaxpct=25
=                               sunit=0       swidth=0 blks
naming    =version 2           bsize=4096    ascii-ci=0, ftype=1
log       =internal log       bsize=4096    blocks=16384, version=2
=                               sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none               extsz=4096    blocks=0, rtextents=0
[U0294255]linux ~]# xfs_admin -L disco2b /dev/sdb2
writing all SBs
new label = "disco2b"
[U0294255]linux ~]# _
```

Por último, crearemos un sistema de archivos **fat32** en la tercera partición del disco, usando esta vez **fatlabel** para etiquetar la partición con DISCO2C:

Para ello ambos comandos serán “**mkfs.vfat -F 32 /dev/sdb3**” y “**fatlabel /dev/sdb3 DISCO2C**” correspondientemente:

```
[U0294255linux ~]# mkfs.vfat -F 32 /dev/sdb3
mkfs.fat 4.2 (2021-01-31)
[U0294255linux ~]# fatlabel /dev/sdb3 DISCO2C
[U0294255linux ~]#
```

Con esto concluimos la parte de asignar un “filesystem” a las particiones. Ahora para finalizar el proceso, es hora del montaje del sistema de ficheros.

Para ello empezaremos creando un directorio para cada partición, con el siguiente formato: “/mnt/pruebaX”

```
[U0294255linux ~]# mkdir /mnt/prueba1
[U0294255linux ~]# mkdir /mnt/prueba2
[U0294255linux ~]# mkdir /mnt/prueba3
[U0294255linux ~]# _
```

Ahora montamos cada partición sobre su respectivo directorio:

```
[U0294255linux ~]# mount /dev/sdb1 /mnt/prueba1
[ 1813.283352] EXT4-fs (sdb1): mounting ext3 file system using the ext4 subsystem
[ 1813.298519] EXT4-fs (sdb1): mounted filesystem 5bafc4fe-d0fb-4058-ab8a-b2ad1f0fe174 r/w with ordered data mode. Quota mode: none.
[U0294255linux ~]# mount /dev/sdb2 /mnt/prueba2
[ 1824.697537] XFS (sdb2): Mounting U5 Filesystem c120bf85-757d-4ccf-9533-30fc87139317
[ 1824.704225] XFS (sdb2): Ending clean mount
[U0294255linux ~]# mount /dev/sdb3 /mnt/prueba3
[U0294255linux ~]# ls /mnt/prueba*
/mnt/prueba1:
lost+found
/mnt/prueba2:
/mnt/prueba3:
[U0294255linux ~]# _
```

Ahora crearemos un fichero dentro del directorio montado de la primera partición con “vi”:

```
"/mnt/prueba1/ejemplo.txt" [New] 2L, 14B written
[U0294255linux ~]# cat /mnt/prueba1/ejemplo.txt
Hola mundo!

[U0294255linux ~]# ls /mnt/prueba*
/mnt/prueba1:
ejemplo.txt  lost+found
/mnt/prueba2:
/mnt/prueba3:
[U0294255linux ~]#
```

Por último, anotaremos el resultado del comando “**lsblk -f**”:

```
[U0294255linux ~]# lsblk -f
NAME        FSTYPE     FSVER    LABEL    UUID                                  FSAVAIL FSUSE%  MOUNTPOINTS
sda
├─sda1      vfat       FAT32    7EAF-147B 95154805-adf4-4b3c-a9d3-06e5b05abc4a  591,7M   1%     /boot/efi
├─sda2      xfs        4.20     disco2a  5bafc4fe-d0fb-4058-ab8a-b2ad1f0fe174  736,6M   23%    /boot
├─sda3      LVM2_member LVM2 001    jDk01E-Tdm0-Udfp-i6B4-msk0-zm20-rx06yd 4790896d-a10e-499a-9004-44d41e9cc6cc  14,9G    9%     /
├─┌─almalinux-root xfs
│   └─almalinux-swap swap        1        023de030-220d-4522-99fb-c0eabc7e6070
sdb
├─sdb1      ext3       1.0      disco2a  5bafc4fe-d0fb-4058-ab8a-b2ad1f0fe174  461,9M   0%     /mnt/prueba1
├─sdb2      xfs        4.20     disco2b  c120bf85-757d-4ccf-9533-30fc87139317   2,9G    2%     /mnt/prueba2
├─sdb3      vfat       FAT32    DISCO2C  622A-53F7 4,5G    0%     /mnt/prueba3
sr0
```

Todo esto que hemos hecho no persistirá debido a las configuraciones por defecto de las maquinas Linux, pero podemos modificarlo para que si lo haga. Para ello añadiremos lo siguiente en el fichero “**/etc/fstab**”:

```

# /etc/fstab
# Created by anaconda on Tue Feb  4 18:09:49 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/almalinux-root /                    xfs     defaults        0 0
UUID=95154805-adf4-4b3c-a9d3-06e5b05abc4a /boot        xfs     defaults        0 0
UUID=7EAF-147B /boot/efi    vfat    umask=0077,shortname=winnt 0 2
/dev/mapper/almalinux-swap none          swap    defaults        0 0
/dev/sdb1 /mnt/prueba1 ext 3 defaults 0 0
/dev/sdb2 /mnt/prueba2 xfs defaults 0 0
/dev/sdb3 /mnt/prueba3 vfat defaults 0 0

```

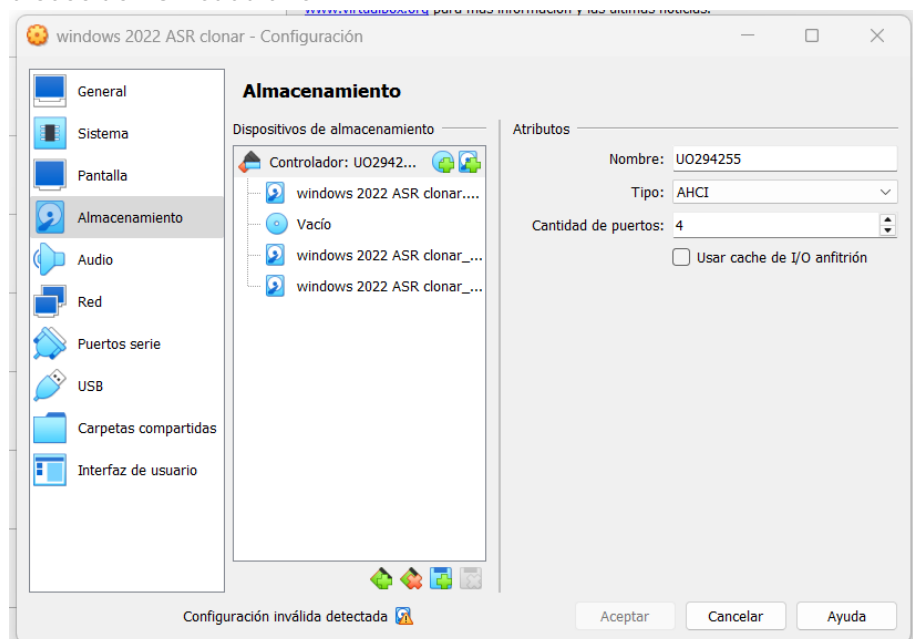
Y con eso bastaría. Guardamos los cambios y habríamos acabado el proceso.

```

[\U0294255linux ~]# df
S.ficheros          bloques de 1K  Usados  Disponibles  Uso%  Montado en
devtmpfs             4096         0        4096         0% /dev
tmpfs                900324         0       900324         0% /dev/shm
tmpfs               360132      5052       355080         2% /run
efivarfs             256         27         225        11% /sys/firmware/efi/efivars
/dev/mapper/almalinux-root 17141760 1492480 15649280         9% /
/dev/sda2            983040     228768     754272        24% /boot
/dev/sda1            613160      7220      605940         2% /boot/efi
tmpfs               180064         0       180064         0% /run/user/0
/dev/sdb1            499204         48     473024         1% /mnt/prueba1
/dev/sdb2            3080192     54516     3025676         2% /mnt/prueba2
/dev/sdb3            4708316         4     4708312         1% /mnt/prueba3
[\U0294255linux ~]#

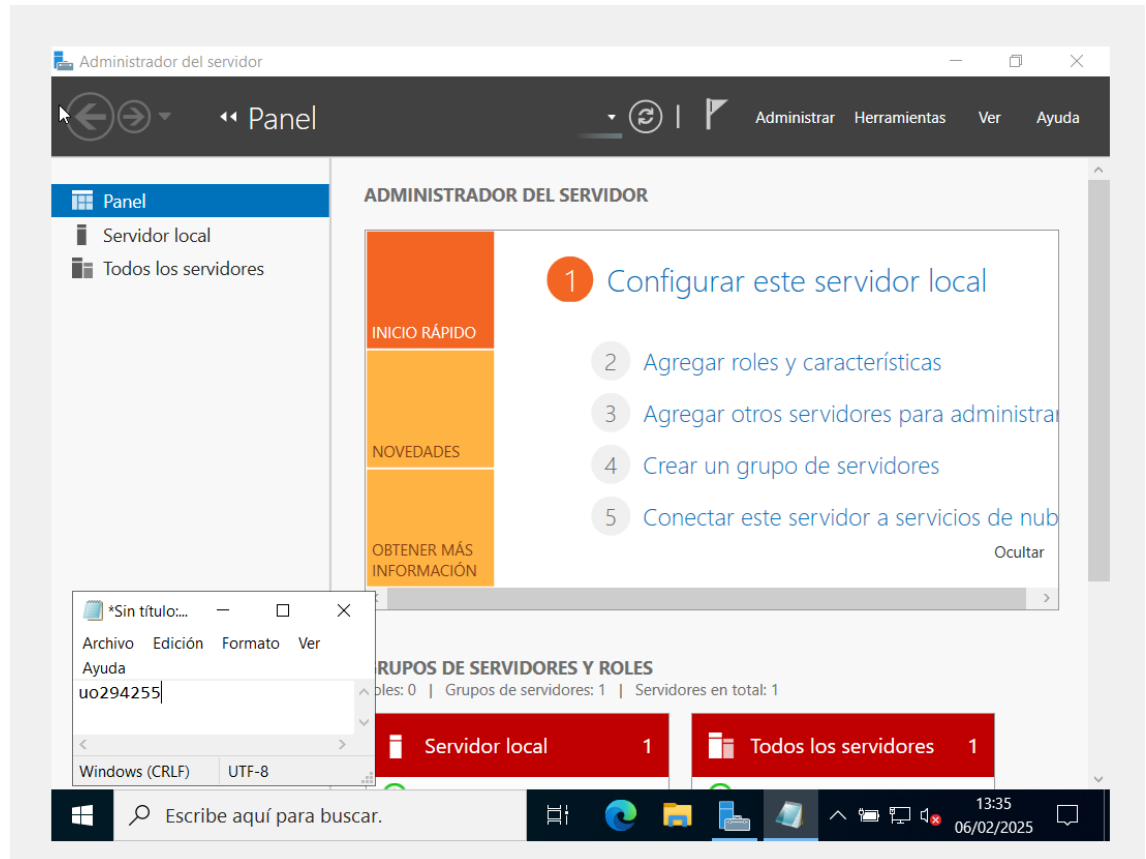
```

- Adición de nuevos discos a un sistema Windows ya instalado.
Ahora se nos pide el mismo proceso, pero en un sistema Windows 2022 ya instalado y con interfaz de usuario.
Para ello lo primero que haremos será nuevamente, desde la pestaña inicial de virtualbox, entrar a la configuración de la maquina virtual que hemos previamente clonado de la de Windows inicial, y añadiremos dos discos de 4GB cada uno.

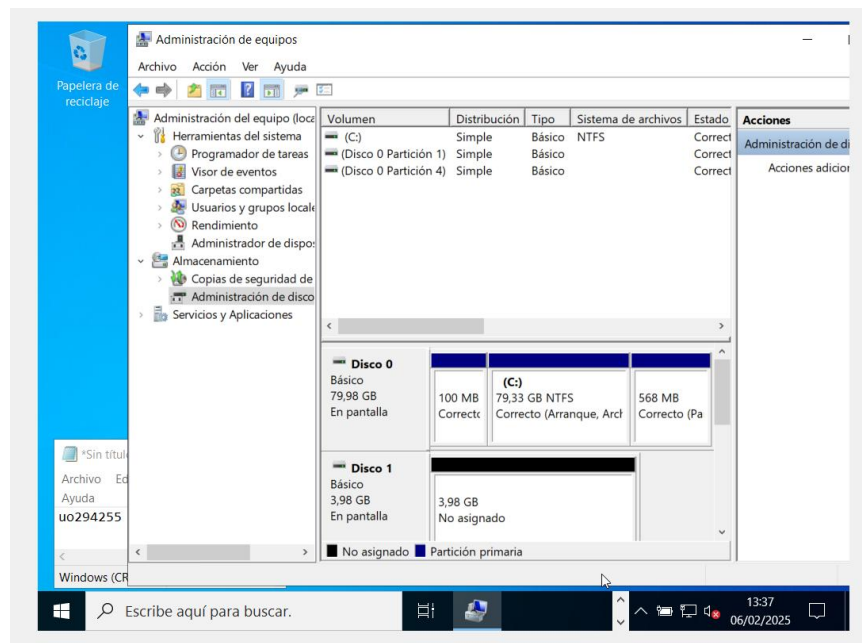


Ahora arrancaremos la máquina virtual.

Una vez dentro, abriremos el “Administrador del servidor”. Clickamos encima de “Herramientas” >> “Administración de equipos” (la primera opción).

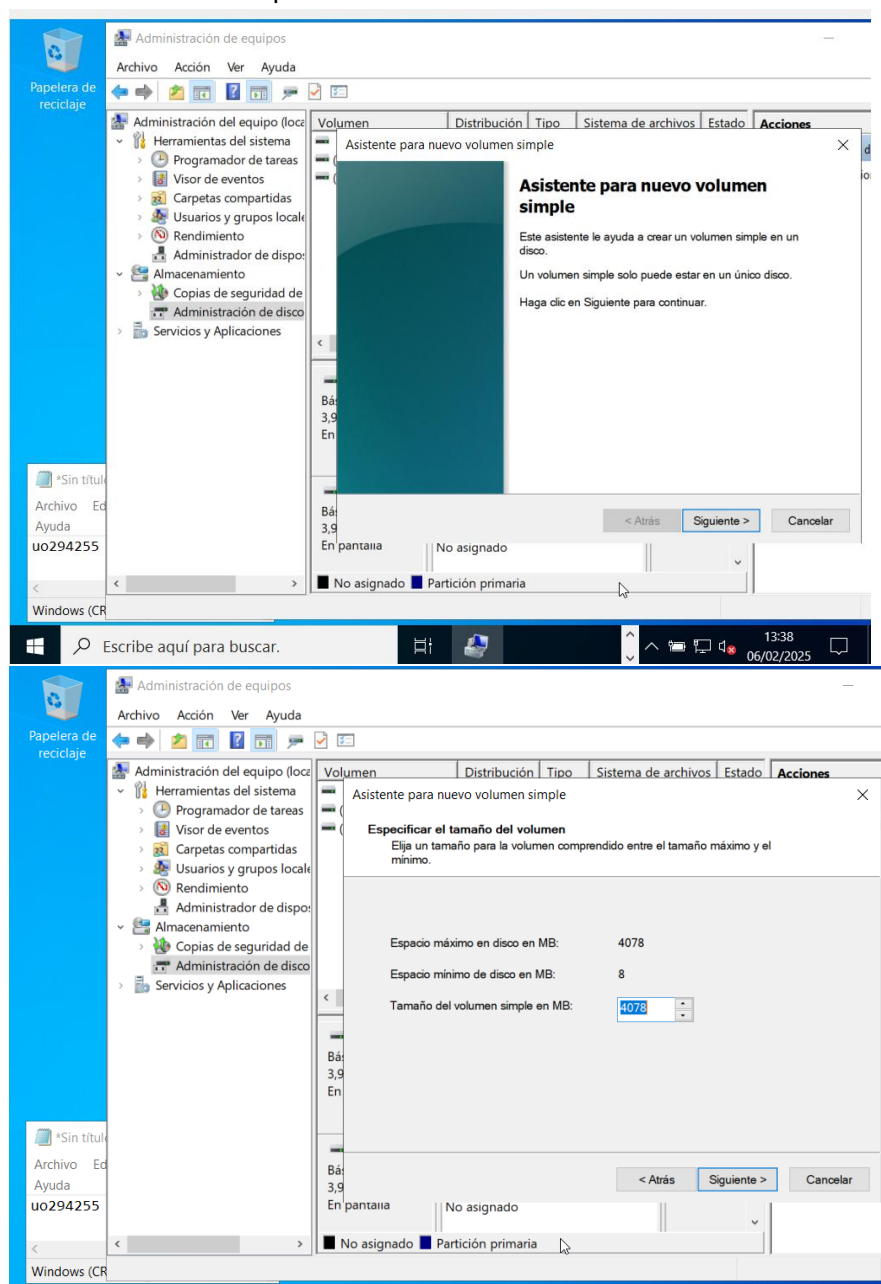


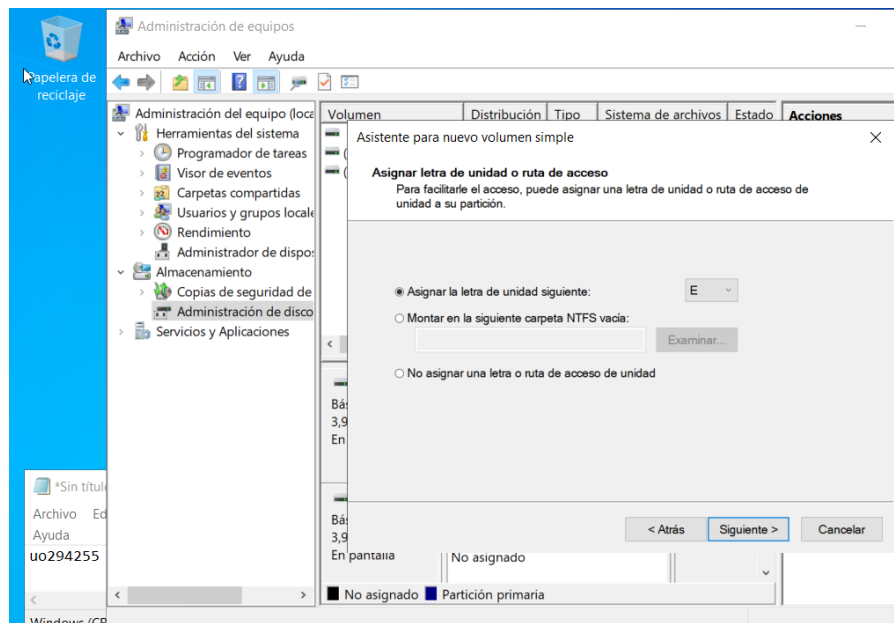
Una vez dentro, pulsaremos sobre Almacenamiento >> Administración de discos:



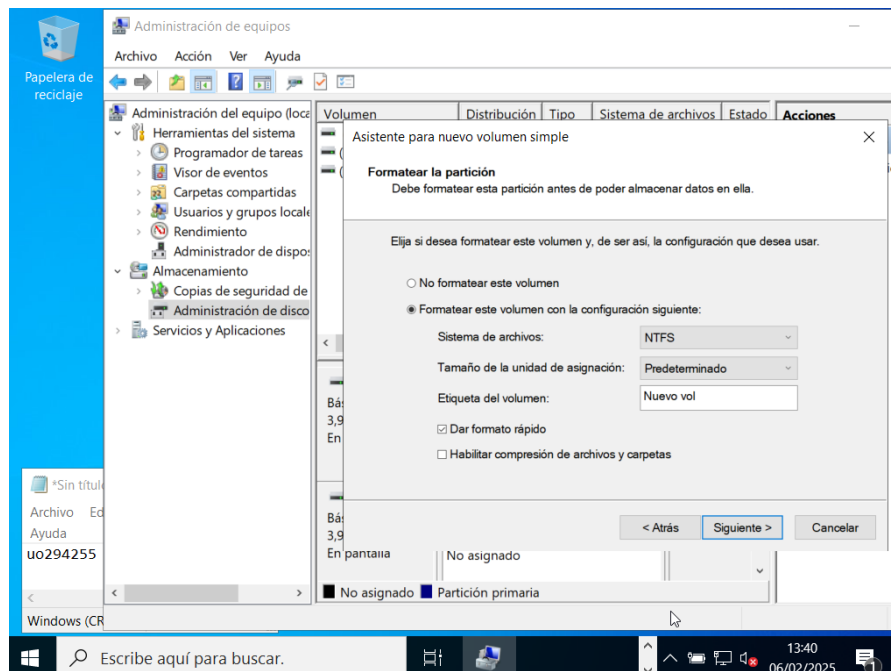
Aquí, tendremos visibles los discos disponibles en el sistema.

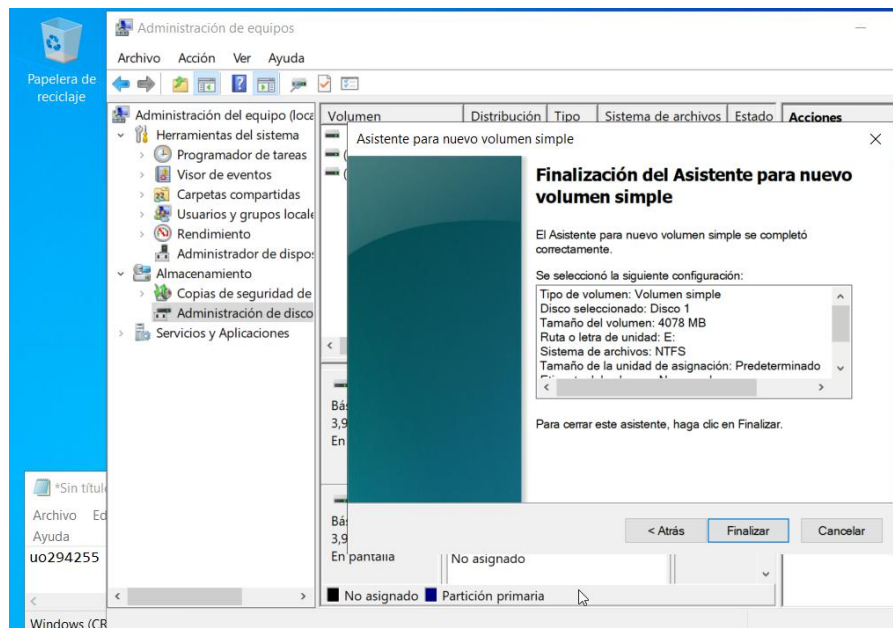
Hacemos click derecho sobre “Disco 1”, y seleccionamos la opción de “Nuevo volumen simple...”



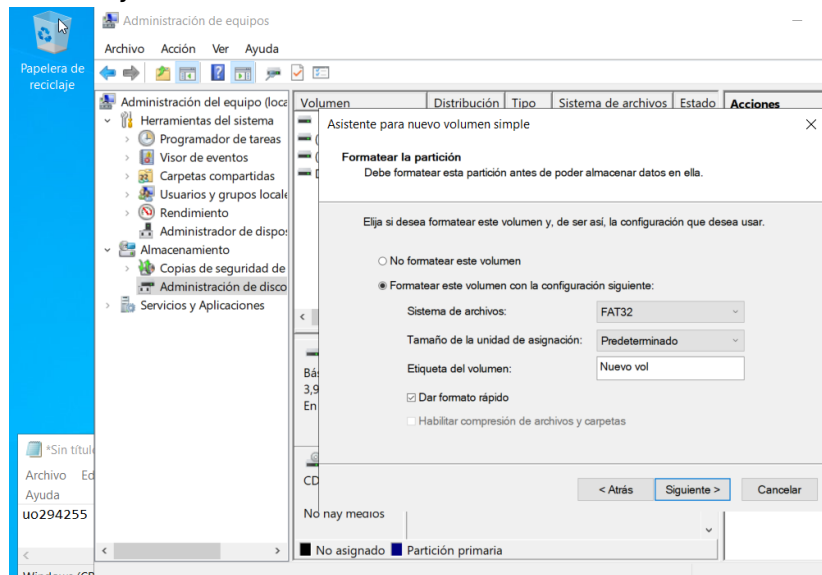


Y lo formatearemos como NTFS:

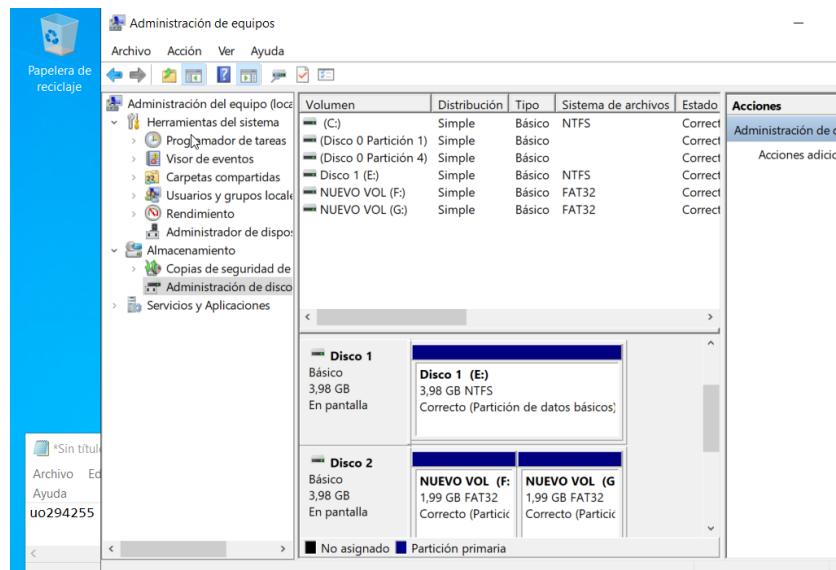




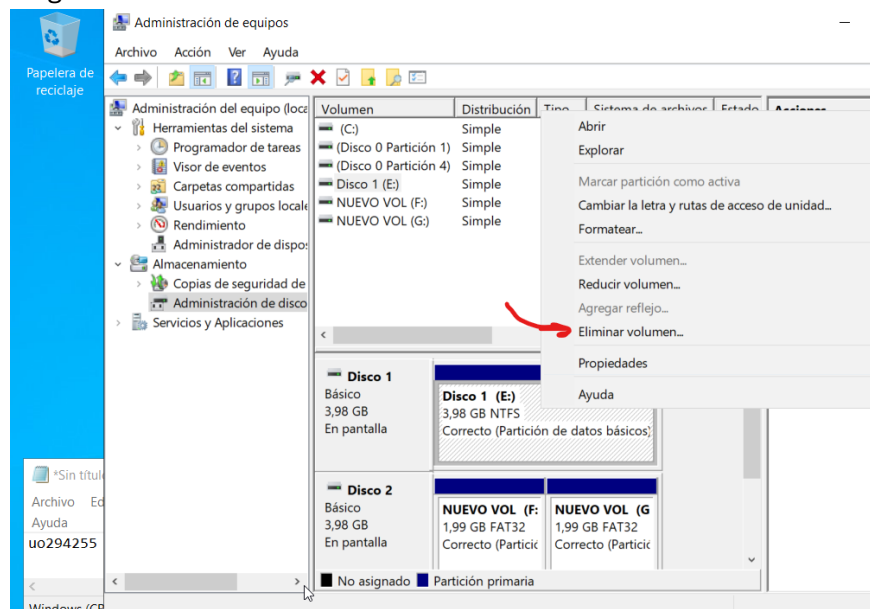
Con el segundo disco hacemos lo mismo, pero haciendo dos particiones internas y formateándolas a FAT32:



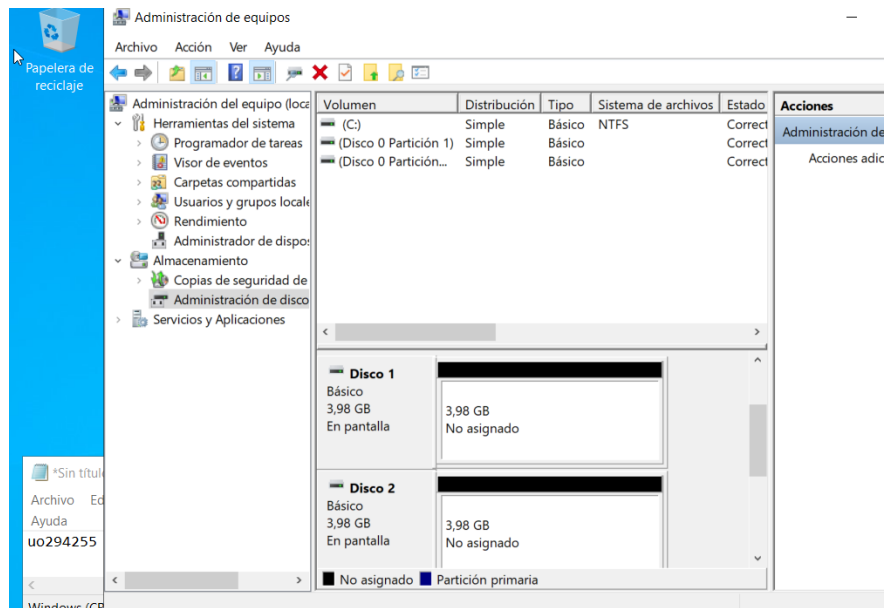
El resultado de ambas operaciones es:



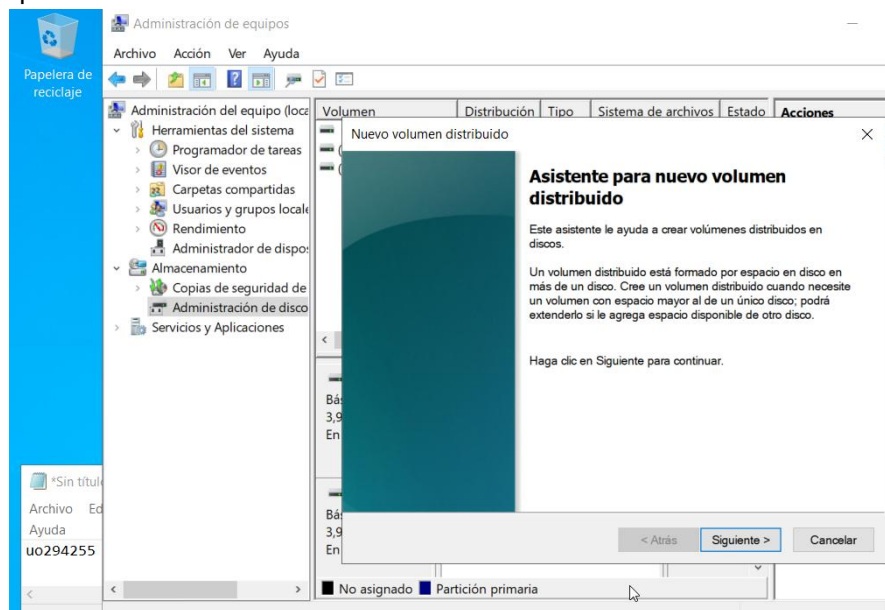
Ahora vamos a eliminar las 3 particiones, dejando todo en el estado “No asignado”:

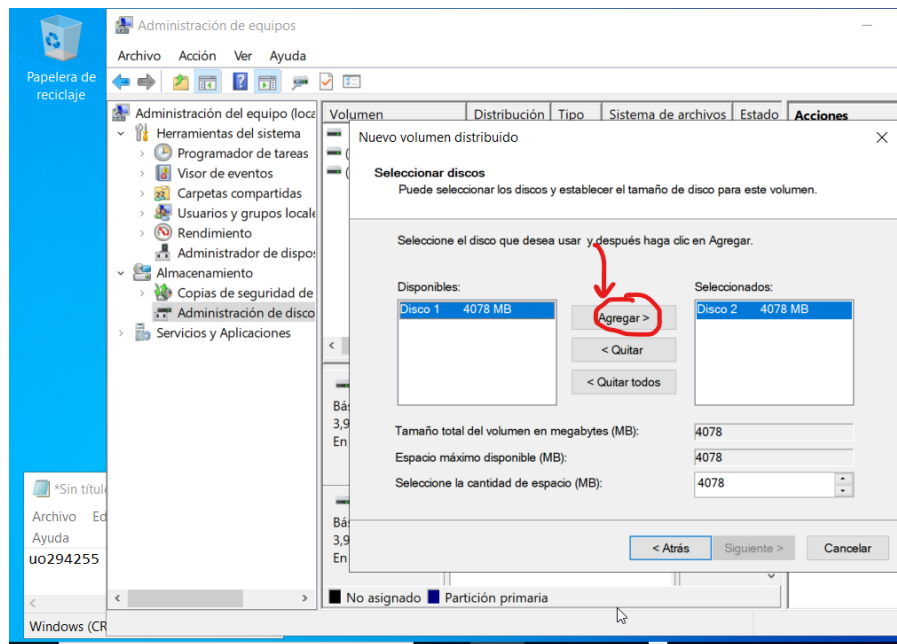


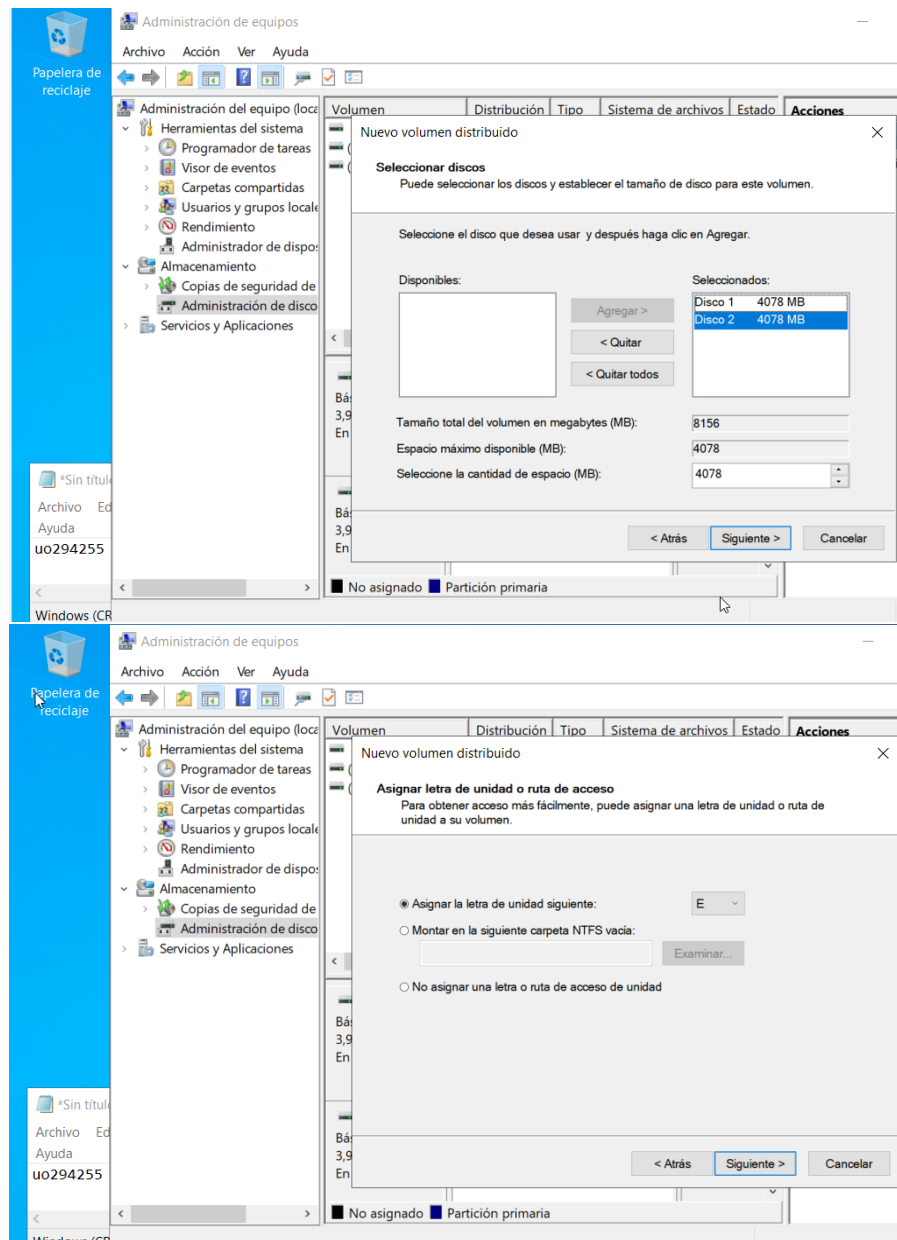
Al eliminar las 3 particiones volveremos a este estado:



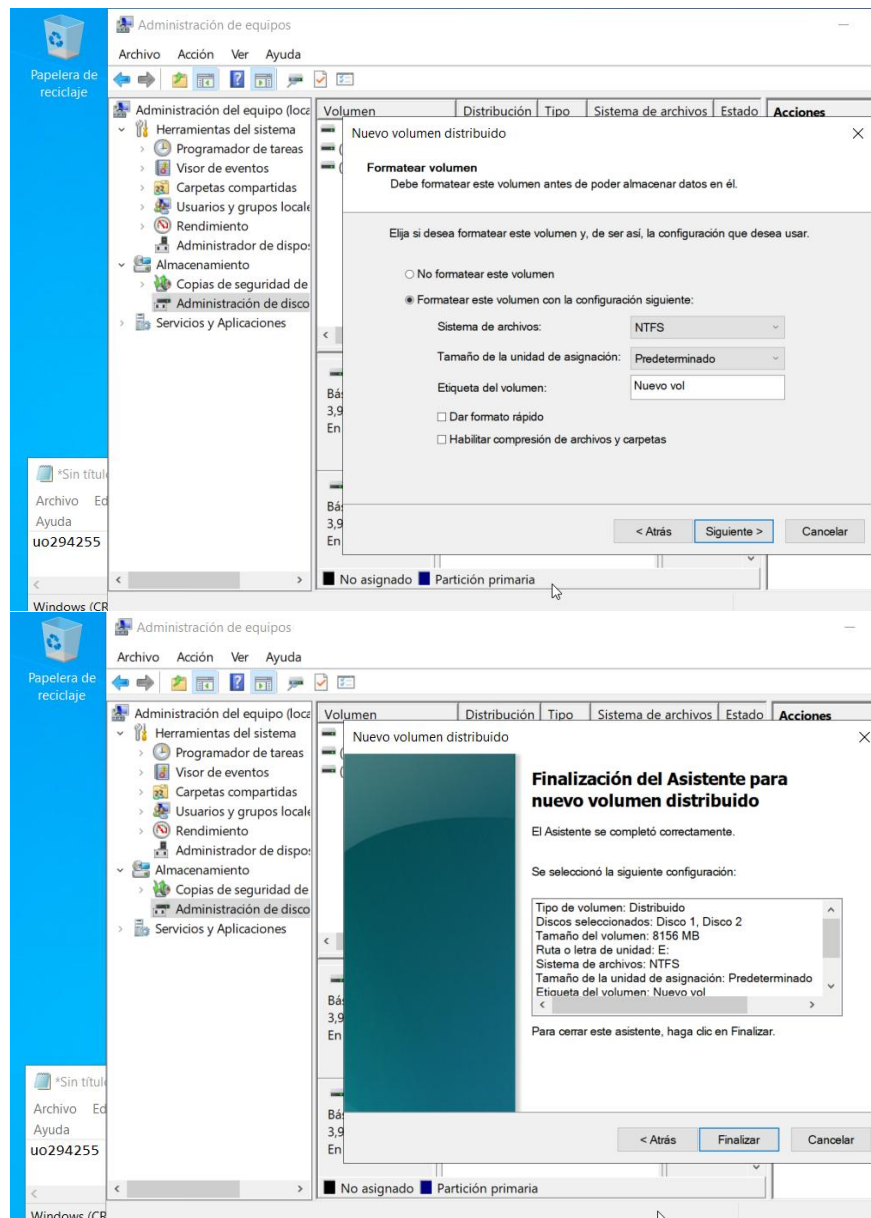
Ahora uniremos ambos discos con un volumen NTFS distribuido, de forma que exista una única unidad E:



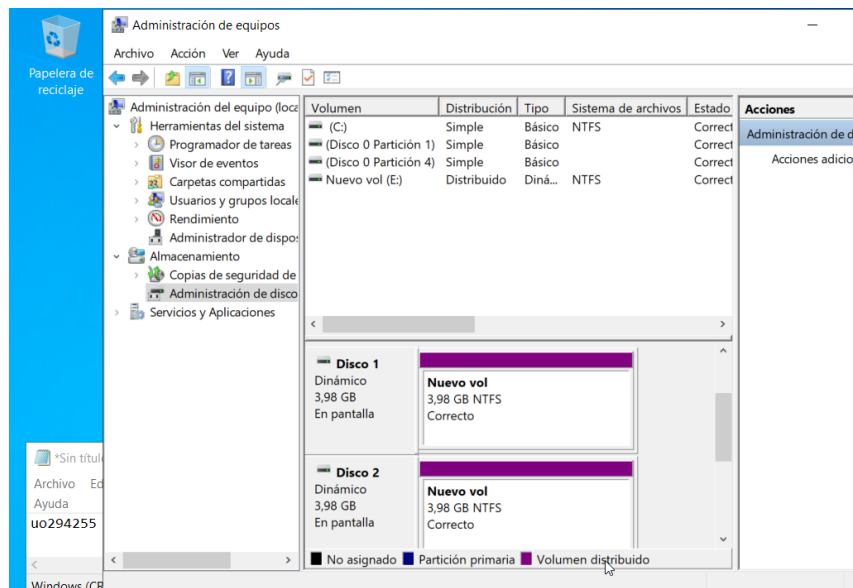




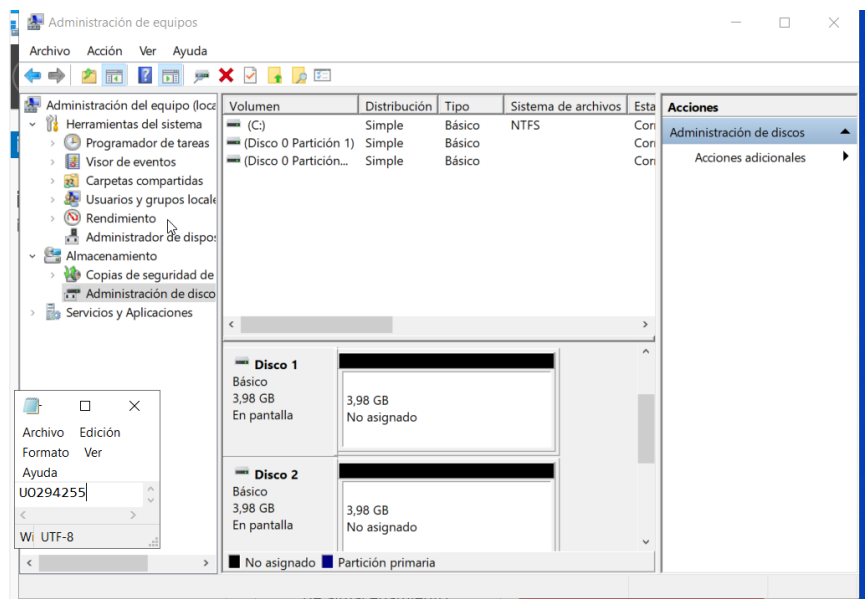
Formatearemos a NTFS también:

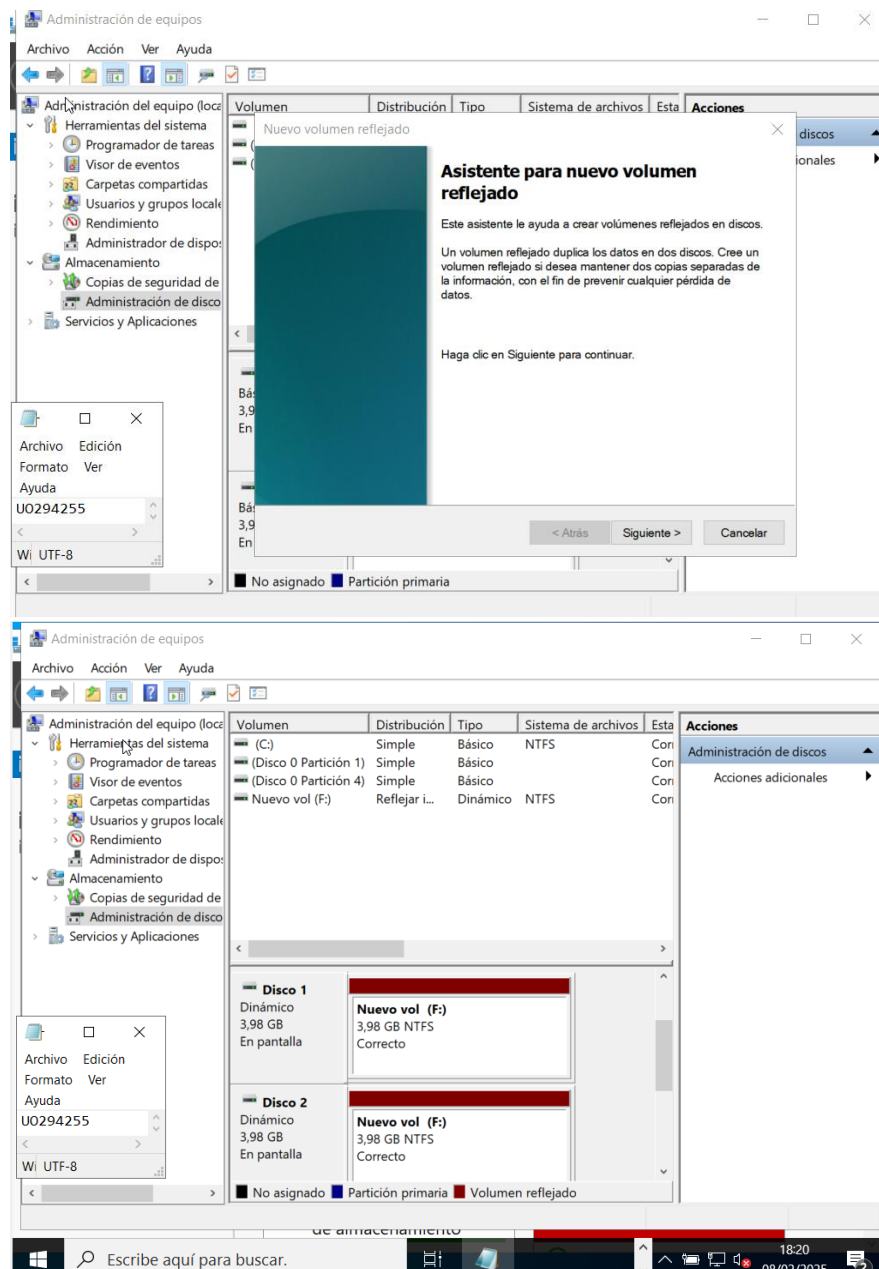


El resultado es:



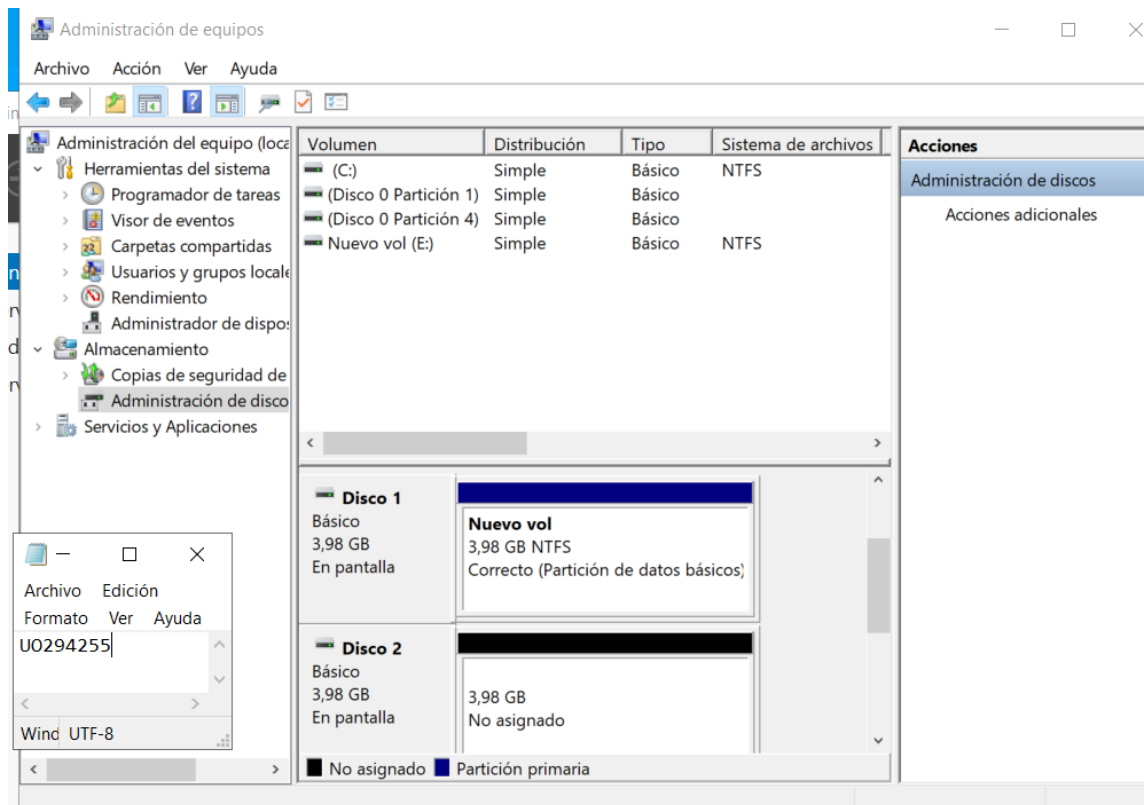
Por último, vamos a eliminar el volumen distribuido y unir de nuevo ambos discos del volumen distribuido NTFS en un volumen reflejado (RAID1).
¿Qué tamaño tiene la nueva unidad?



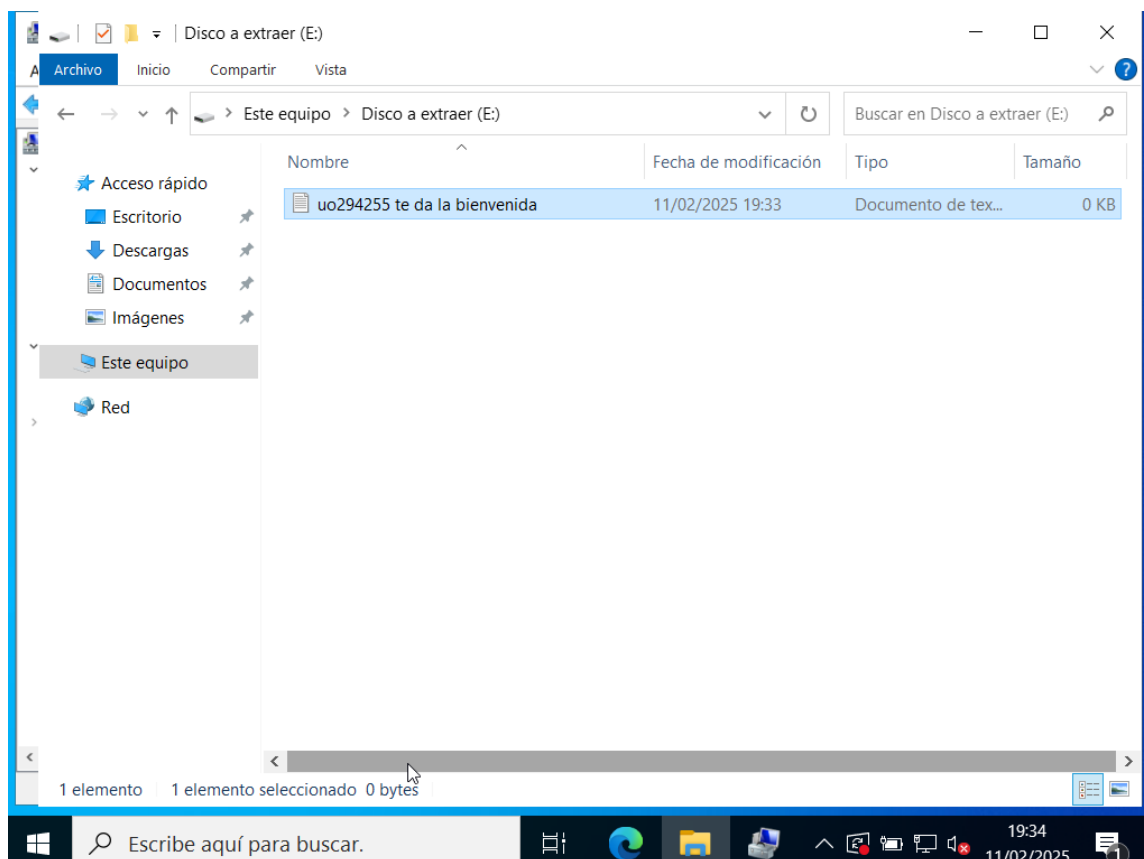


D. Traspase de discos entre máquinas con distintos operativos

Lo primero de todo será ver a colocar un disco simple NTFS en el Disco 1 de la maquina Windows:



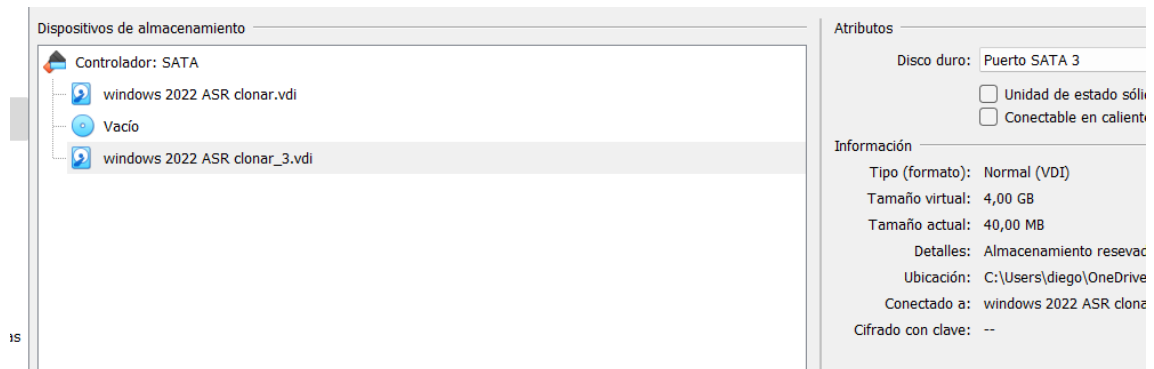
Para luego añadir un fichero txt en ese mismo disco:



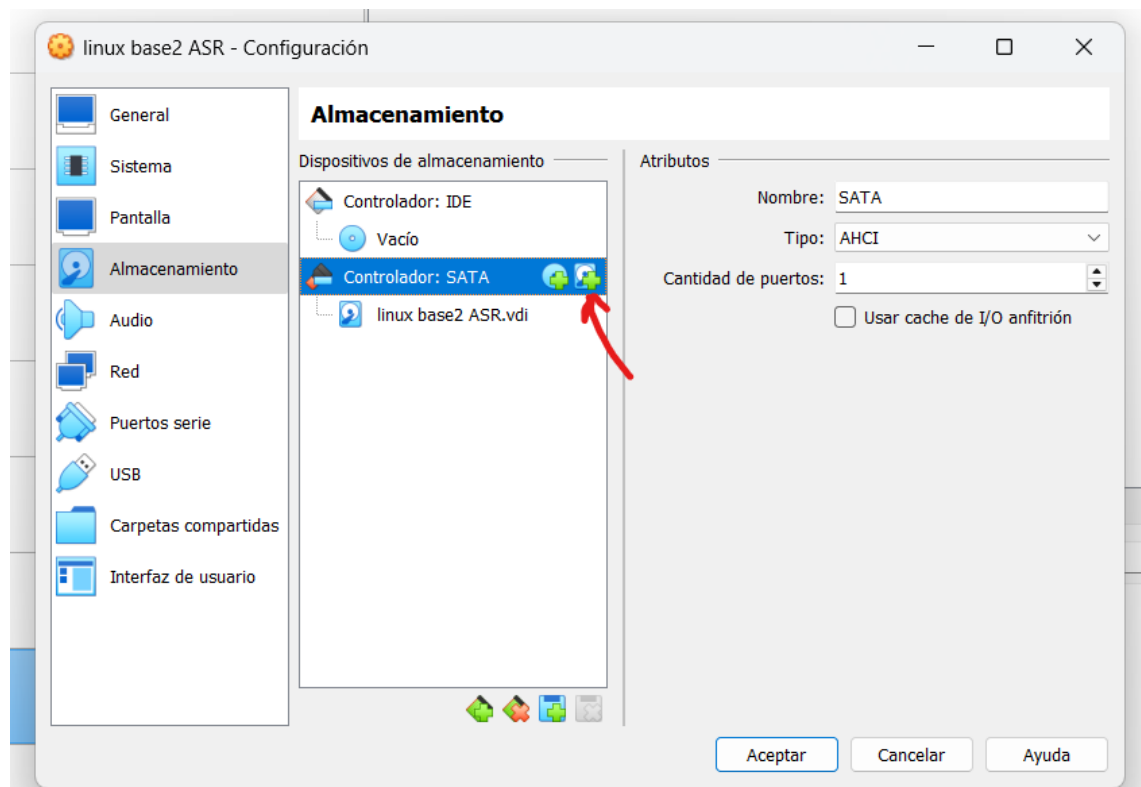
Por último, apagamos la máquina.

Ahora pasamos a virtual box, donde simularemos el trasvase de disco de una maquina a otra.

Primero eliminamos el disco en cuestión de la maquina Windows:



Y luego lo instalamos en la maquina Linux:



<div> <div>Añadir</div> <div>Crear</div> <div>Actualizar</div> </div>		
Nombre	Tamaño virtual	Tamaño actual
✓ Attached		
2020-teamwork-disk001.vdi	20,00 GB	19,09 GB
2ac-32bits-disk001.vdi	10,00 GB	4,80 GB
2AC-disk001.vdi	10,00 GB	4,80 GB
linux base ASR.vdi	40,00 GB	2,68 GB
linux base2 ASR.vdi	20,00 GB	2,39 GB
linux GUI ASR.vdi	80,00 GB	5,94 GB
TeamWork-disk002.vdi	20,00 GB	18,68 GB
UbuntuSSI VFinal-disk001.vdi	80,00 GB	13,10 GB
windows 2022 ASR clonar.vdi	80,00 GB	19,52 GB
windows 2022 ASR clonar_3.vdi	4,00 GB	40,00 MB
windows 2022 ASR.vdi	80,00 GB	12,91 GB
✓ Not Attached		
windows 2022 ASR clonar_1.vdi	8,00 GB	37,00 MB
windows 2022 ASR clonar_2.vdi	4,00 GB	37,00 MB

Una vez hecho esto, encendemos la maquina Linux e instalamos el driver correspondiente para que la maquina acepte particiones NTFS:

```
[U0294255linux ~]# dnf -y install epel-release
AlmaLinux 9 - AppStream
AlmaLinux 9 - AppStream
AlmaLinux 9 - BaseOS
AlmaLinux 9 - BaseOS
AlmaLinux 9 - Extras
AlmaLinux 9 - Extras
Última comprobación de caducidad de metadatos hecha hace 0:00:01, el mar 11 feb 2025 19:42:27.
Dependencias resueltas.
=====
Paquete                                Arquitectura  Versión
=====
Instalando:
epel-release                           noarch       9-9.e19
=====
Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 18 k
Tamaño instalado: 26 k
Descargando paquetes:
epel-release-9-9.e19.noarch.rpm
=====
Total
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando      :
Instalando      : epel-release-9-9.e19.noarch
Ejecutando scriptlet: epel-release-9-9.e19.noarch
Many EPEL packages require the CodeReady Builder (CRB) repository.
It is recommended that you run /usr/bin/crb enable to enable the CRB repository.
[ 95.917422] systemd-rc-local-generator[14511]: /etc/rc.d/rc.local is not marked executable, skipping.
Verificando     : epel-release-9-9.e19.noarch

Instalado:
epel-release-9-9.e19.noarch

¡Listo!

[U0294255linux ~]# dnf -y upgrade
```

```

[U0294255]linux ~]# dnf -y install ntfs-3g
Última comprobación de caducidad de metadatos hecha hace 0:02:24, el mar 11 feb 2025 19:43:16.
Dependencias resueltas.
=====
Paquete                Arquitectura      Versión           Repositorio
-----
Instalando:
ntfs-3g                x86_64            2:2022.10.3-1.el9 epel
Instalando dependencias:
ntfs-3g-libs           x86_64            2:2022.10.3-1.el9 epel
Resumen de la transacción
-----
Instalar 2 Paquetes

Tamaño total de la descarga: 302 k
Tamaño instalado: 696 k
Descargando paquetes:
(1/2): ntfs-3g-libs-2022.10.3-1.el9.x86_64.rpm                953 kB/s | 174
(2/2): ntfs-3g-2022.10.3-1.el9.x86_64.rpm                    607 kB/s | 128
-----
Total                                                         348 kB/s | 302
Extra Packages for Enterprise Linux 9 - x86_64                1.6 MB/s | 1.6
Importando llave GPG 0x3220467C:
ID usuario: "Fedora (epel9) <epel@fedoraproject.org>"
Huella      : FF0A D194 4597 186E CE01 3B91 0A38 72BF 3220 467C
Desde       : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-9
La llave ha sido importada exitosamente
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando :
Instalando : ntfs-3g-libs-2:2022.10.3-1.el9.x86_64
Instalando : ntfs-3g-2:2022.10.3-1.el9.x86_64
Ejecutando scriptlet: ntfs-3g-2:2022.10.3-1.el9.x86_64
Verificando : ntfs-3g-2:2022.10.3-1.el9.x86_64
Verificando : ntfs-3g-libs-2:2022.10.3-1.el9.x86_64

Instalado:
ntfs-3g-2:2022.10.3-1.el9.x86_64          ntfs-3g-libs-2:2022.10.3-1.el9.x86_64

¡Listo!
[U0294255]linux ~]#

```

Ahí vemos nuestro disco:

```

[U0294255]linux ~]# lsblk -f
NAME        FSTYPE     FSUVER  LABEL      UUID                                  FSAVAIL FSUSE% MOUNTPOINTS
sda
├─sda1      vfat       FAT32   7EAF-147B   95154805-adf4-4b3c-a9d3-06e5b05abc4a 591,7M  1%  /boot/efi
├─sda2      xfs
├─sda3      LVM2_member LVM2 001     jDk01E-Tdm0-Udfp-i6B4-msk0-zm20-rxD6yd 646,9M  33%  /boot
│   └─almalinux-root xfs
│       └─almalinux-swap swap      1       4790896d-a10e-499a-9004-44d41e9cc6cc 14,2G  13%  /
sdb
├─sdb1
├─sdb2      ntfs       Disco a extraer B206F79206F75637
sr0

```

```

[U0294255]linux ~]# ls -l /mnt/prueba
total 0
drwxrwxrwx. 1 root root 0 feb 11 19:33 '$RECYCLE.BIN'
drwxrwxrwx. 1 root root 0 feb 11 19:30 'System Volume Information'
-rwxrwxrwx. 1 root root 0 feb 11 19:33 'uo294255 te da la bienvenida.txt'

```

En el disco 1 se encuentra otra partición, que puede ser una partición residual, la cual no es necesaria. Luego tiene la partición buena, que es donde tenemos el fichero .txt, que si es necesaria.