



## Análisis Forense : Clonado

Maria Andrea Ugarte Valencia y Marcos Villar Avión

March 21, 2024

### 1

De aquí al tercer ejercicio se han usado dos pendrives: la evidencia y otro para realizar pruebas de automontado. A continuación, se adjuntan unas fotografías de la evidencia:



Figure 1: Parte delantera de la evidencia



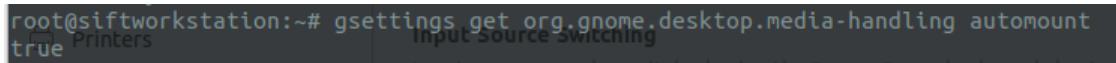
Figure 2: Parte trasera de la evidencia

Como podemos observar, se trata de un pendrive de 8 GB, por lo que los clonados o los cálculos de hashes posteriores no deberían llevar una cantidad exagerada de tiempo.

## 1.a

Después de analizar las opciones disponibles para desactivar el automontado como, por ejemplo, la modificación de las reglas udev, hemos decidido usar la siguiente opción debido a que nos resulta más sencilla:

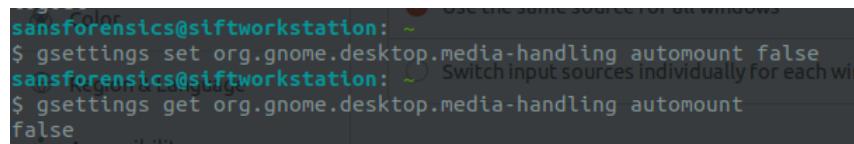
```
gsettings get org.gnome.desktop.media-handling automount
```



```
root@siftworkstation:~# gsettings get org.gnome.desktop.media-handling automount
true
```

Figure 3: Comprobación del estado del automontado

Vemos como está a *true*, esto quiere decir que los dispositivos se montarán automáticamente. Para cambiarlo ejecutamos la versión *set* del comando anterior y verificamos los cambios:



```
sansforensics@siftworkstation: ~
$ gsettings set org.gnome.desktop.media-handling automount false
sansforensics@siftworkstation: ~ Switch input sources individually for each window
$ gsettings get org.gnome.desktop.media-handling automount
false
```

Figure 4: Cambio del estado del automontado

Como nuestra máquina *host* es Windows, vamos a evitar, antes de insertar la evidencia, que se monten automáticamente los USB en nuestra máquina anfitrion. Primero, intentamos hacerlo de forma manual accediendo al **Registry Editor**, pero no tuvimos éxito. Como alternativa, hemos hecho uso de la herramienta **PolicyPlus**<sup>1</sup>. Esta herramienta nos ayudará a modificar los registros importantes que hacen que un USB se monte automáticamente en nuestra máquina Windows.

Una vez instalado **PolicyPlus**, para conseguir deshabilitar el automontado activamos las opciones de denegación de permisos de los discos extraíbles en **Sistema - Acceso de almacenamiento extraíble**.

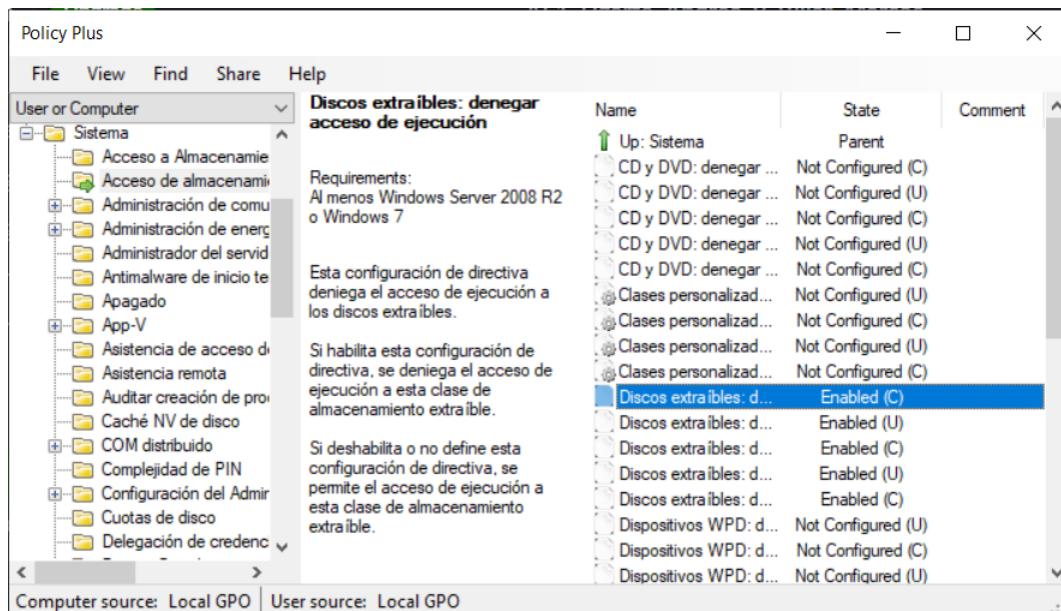


Figure 5: Configuración PolicyPlus

<sup>1</sup>[Github: PolicyPlus](#)

Una vez realizada dicha configuración, podemos comprobar que tras insertar un USB, el SO nos mostrará el siguiente aviso en caso de que intentemos acceder. Tanto en este paso como en el siguiente se ha usado previamente un pendrive de prueba para verificar que efectivamente no se montan los USB y que nuestra evidencia no correría el peligro de un cambio de hash.

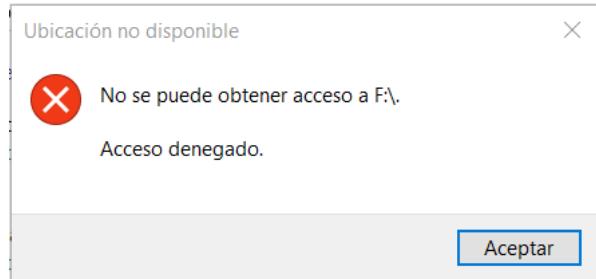


Figure 6: Acceso denegado al dispositivo en Windows

Vamos a mostrar ahora que el USB no se monta automáticamente en la la máquina **SIFT Workstation**. Para verificar que el dispositivo no se ha montado ejecutamos el siguiente comando:

```
lsblk
```

```
sansforensics@siftworkstation: ~
$ lsblk
NAME      MAJ:MIN   RM   SIZE RO TYPE MOUNTPOINTS
loop0      7:0       0    4K  1 loop /snap/bare/5
loop1      7:1       0 159.6M 1 loop /snap/chromium/2738
loop2      7:2       0   62M  1 loop /snap/core20/1587
loop3      7:3       0  74.1M  1 loop /snap/core22/1033
loop4      7:4       0  66.5M  1 loop /snap/cups/1024
loop5      7:5       0 497M  1 loop /snap/gnome-42-2204/141
loop6      7:6       0 91.7M  1 loop /snap/gtk-common-themes/1535
loop7      7:7       0 79.9M  1 loop /snap/lxd/22923
loop8      7:8       0 40.4M  1 loop /snap/snapd/20671
sda
└─sda1     8:0       0 488.3G 0 disk
  └─part1  8:1       0   1M  0 part
  └─part2  8:2       0   2G  0 part /boot
  └─part3  8:3       0 486.3G 0 part ...
  └─ubuntu--vg-ubuntu--lv 253:0  0 100G 0 lvm /
sdb
└─sdb1     8:16      1   7.3G 0 disk
  └─part1  8:17      1   7.3G 0 part
```

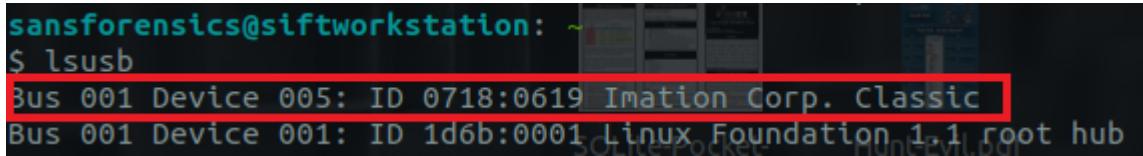
Figure 7: Salida de lsblk

Podemos ver como nos aparece una breve información del dispositivo y, como no está montado, no existe ninguna carpeta en la que podamos acceder a los recursos del mismo.

## 1.b

Vamos ahora a consultar los datos identificativos del dispositivo, para ello podemos usar los comandos:

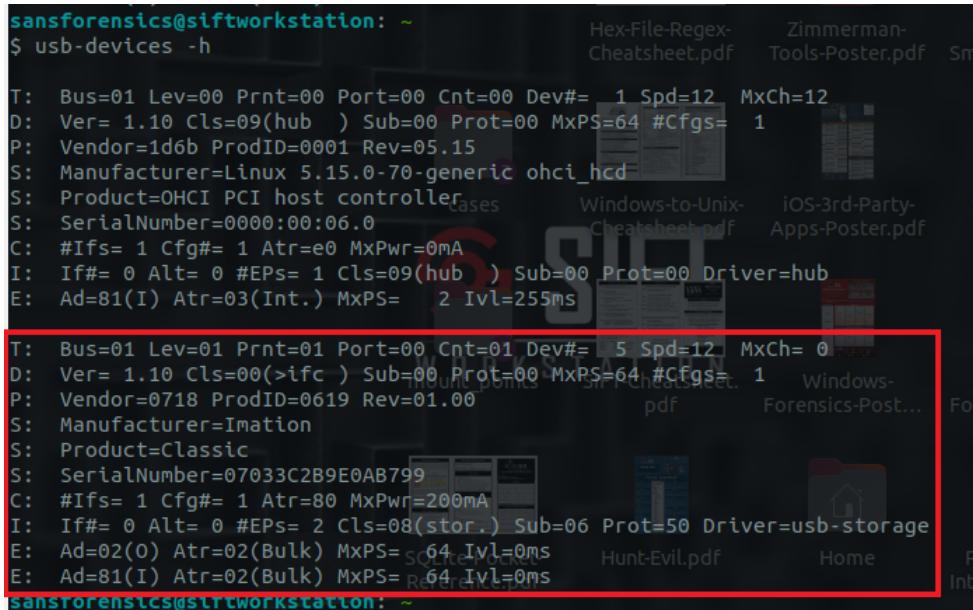
```
lsusb  
usb-devices
```



```
sansforensics@siftworkstation: ~  
$ lsusb  
Bus 001 Device 005: ID 0718:0619 Imation Corp. Classic  
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

Figure 8: Información de la evidencia con lsusb

Como podemos ver, en el caso de **lsusb** se nos muestra el nombre del USB, el ID del fabricante y el ID del producto.



```
sansforensics@siftworkstation: ~  
$ usb-devices -h  
T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh=12  
D: Ver= 1.10 Cls=09(hub ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1  
P: Vendor=1d6b ProdID=0001 Rev=05.15  
S: Manufacturer=Linux 5.15.0-70-generic ohci_hcd  
S: Product=OHCI PCI host controller  
S: SerialNumber=0000:00:06.0  
C: #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA  
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub  
E: Ad=81(I) Atr=03(Int.) MxPS= 2 Ivl=255ms  
  
T: Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#= 5 Spd=12 MxCh= 0  
D: Ver= 1.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1  
P: Vendor=0718 ProdID=0619 Rev=01.00  
S: Manufacturer=Imation  
S: Product=Classic  
S: SerialNumber=07033C2B9E0AB799  
C: #Ifs= 1 Cfg#= 1 Atr=80 MxPwr=200mA  
I: If#= 0 Alt= 0 #EPs= 2 Cls=08(stor.) Sub=06 Prot=50 Driver=usb-storage  
E: Ad=02(0) Atr=02(Bulk) MxPS= 64 Ivl=0ms  
E: Ad=81(I) Atr=02(Bulk) MxPS= 64 Ivl=0ms
```

Figure 9: Información de la evidencia con usb-devices

Con **usb-devices** podemos ver la información anterior. Además, se añaden más detalles como el número de serie o el driver usado para gestionar el dispositivo. En nuestra opinión esta herramienta es mejor ya que es más completa.

Tras una breve investigación, hemos concluido que las opciones recomendadas para realizar el hash del USB son SHA-2 o SHA-3. Por esta razón, hemos optado por utilizar SHA-2, específicamente SHA-256, para calcular el hash de nuestra muestra. Afortunadamente, nuestro dispositivo es compatible con esta función, por lo que no necesitamos recurrir a alternativas menos seguras como SHA-1 o MD5.

Vamos a usar la herramienta sha256sum para poder hacer el hash SHA256:

```
sha256sum /dev/sdX
```

Ahora vamos a esperar a que realice el hash del usb y nos mostrará el siguiente resultado:



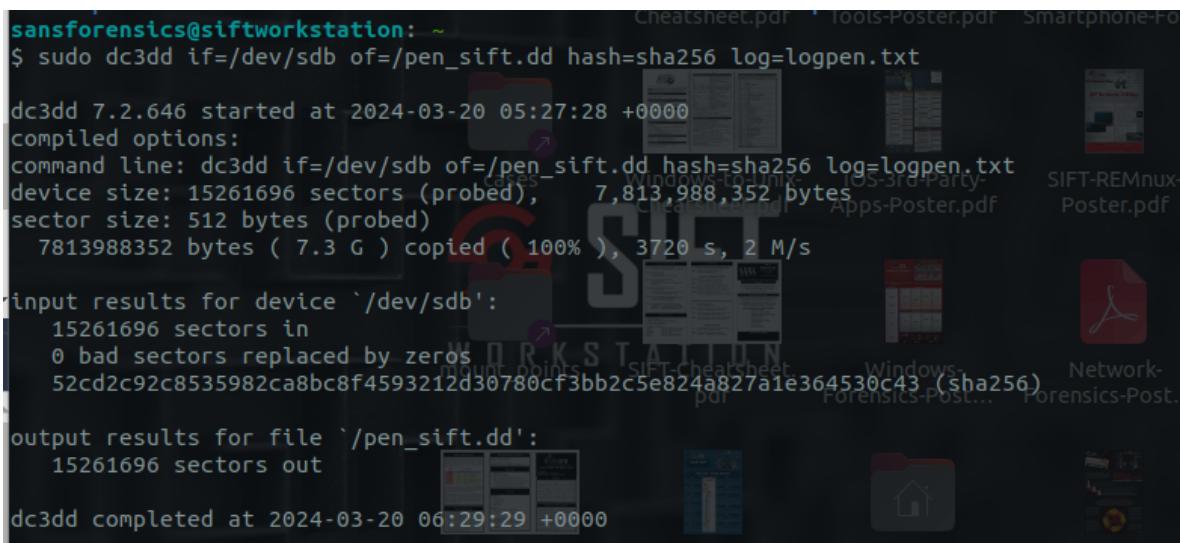
```
sansforensics@siftworkstation: ~
$ sudo sha256sum /dev/sdb
52cd2c92c8535982ca8bc8f4593212d30780cf3bb2c5e824a827a1e364530c43 /dev/sdb
```

Figure 10: Hash de la evidencia

### 1.c

Gracias a la herramienta dc3dd, clonaremos el pendrive a una imagen denominada pen\_sift.dd, indicando que guarde un log y que calcule el hash de la imagen con las siguientes opciones:

```
sudo dc3dd if=/dev/sdb of=/pen_sift.dd hash=sha256 log=logopen.txt
```



```
sansforensics@siftworkstation: ~
$ sudo dc3dd if=/dev/sdb of=/pen_sift.dd hash=sha256 log=logopen.txt
dc3dd 7.2.646 started at 2024-03-20 05:27:28 +0000
compiled options:
command line: dc3dd if=/dev/sdb of=/pen_sift.dd hash=sha256 log=logopen.txt
device size: 15261696 sectors (probed), 7,813,988,352 bytes
sector size: 512 bytes (probed)
    7813988352 bytes ( 7.3 G ) copied ( 100% ), 3720 s, 2 M/s

input results for device `/dev/sdb':
    15261696 sectors in
    0 bad sectors replaced by zeros
    52cd2c92c8535982ca8bc8f4593212d30780cf3bb2c5e824a827a1e364530c43 (sha256)

output results for file `/pen_sift.dd':
    15261696 sectors out

dc3dd completed at 2024-03-20 06:29:29 +0000
```

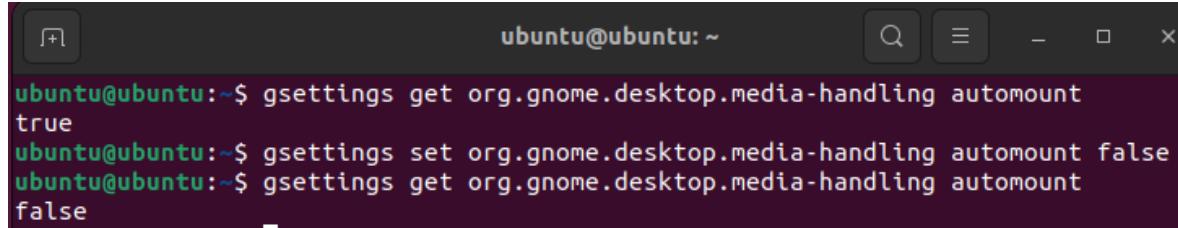
Figure 11: Clonado en SIFT Workstation

Una vez realizado el clonado podemos ver que efectivamente el hash coincide con el calculado en el apartado anterior, lo que significa que ha ido todo bien.

## 2

### 2.a

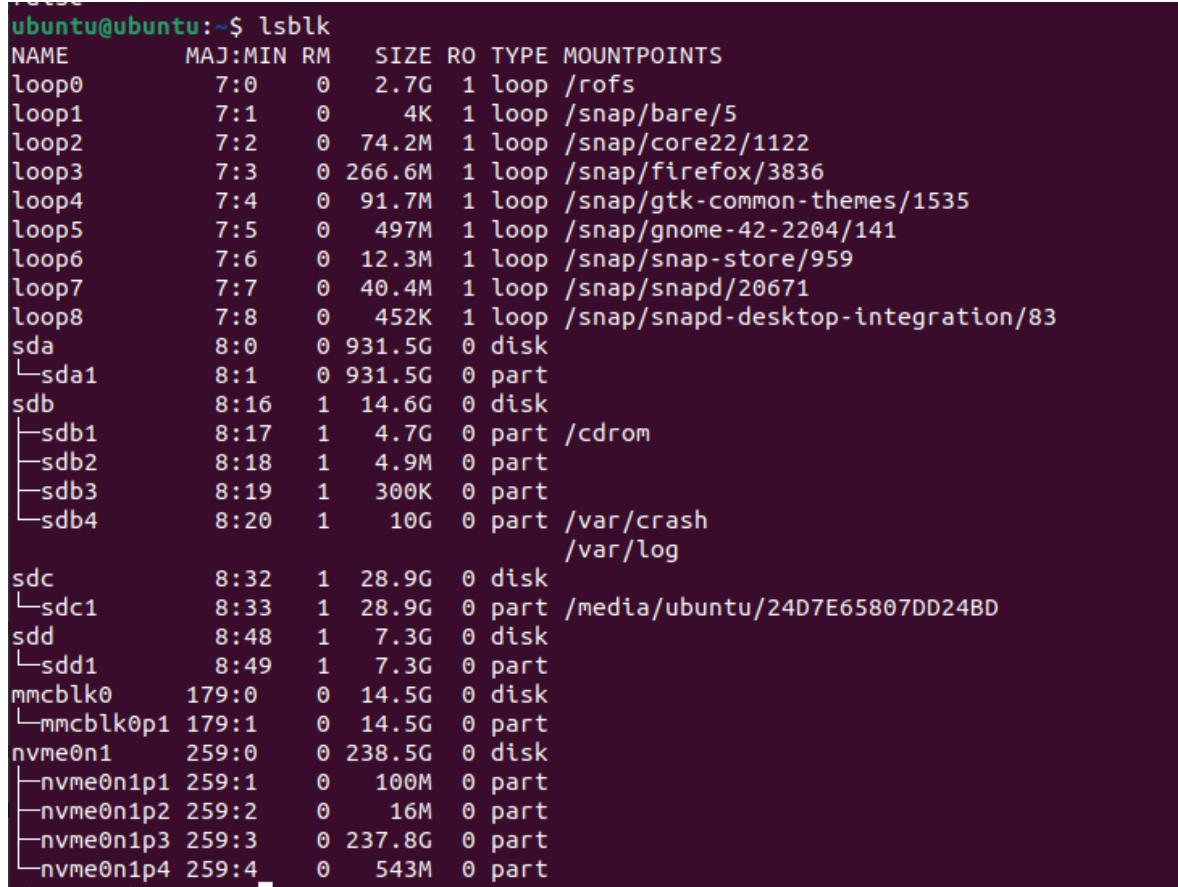
Arrancamos nuestro equipo con la distribución de Ubuntu Live y desactivamos el automontado con los comandos del apartado 1.a.



```
ubuntu@ubuntu:~$ gsettings get org.gnome.desktop.media-handling automount
true
ubuntu@ubuntu:~$ gsettings set org.gnome.desktop.media-handling automount false
ubuntu@ubuntu:~$ gsettings get org.gnome.desktop.media-handling automount
false
```

Figure 12: Comprobación del estado de automontado y desactivación del mismo

Ahora, con **lsblk** comprobamos que efectivamente no se ha montado. Aunque esto se ha hecho antes de los pasos anteriores, esta captura es posterior, por lo que en la imagen sale que se corresponde a sdd pero, en el momento de la verdad, se correspondía a sdc.



```
ubuntu@ubuntu:~$ lsblk
NAME      MAJ:MIN RM    SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0  2.7G  1 loop /rofs
loop1      7:1    0   4K  1 loop /snap/bare/5
loop2      7:2    0 74.2M  1 loop /snap/core22/1122
loop3      7:3    0 266.6M  1 loop /snap/firefox/3836
loop4      7:4    0 91.7M  1 loop /snap/gtk-common-themes/1535
loop5      7:5    0 497M  1 loop /snap/gnome-42-2204/141
loop6      7:6    0 12.3M  1 loop /snap/snap-store/959
loop7      7:7    0 40.4M  1 loop /snap/snapd/20671
loop8      7:8    0  452K  1 loop /snap/snapd-desktop-integration/83
sda        8:0    0 931.5G  0 disk
└─sda1     8:1    0 931.5G  0 part
sdb        8:16   1 14.6G  0 disk
└─sdb1     8:17   1   4.7G  0 part /cdrom
└─sdb2     8:18   1   4.9M  0 part
└─sdb3     8:19   1   300K  0 part
└─sdb4     8:20   1    10G  0 part /var/crash
                           /var/log
sdc        8:32   1 28.9G  0 disk
└─sdc1     8:33   1 28.9G  0 part /media/ubuntu/24D7E65807DD24BD
sdd        8:48   1  7.3G  0 disk
└─sdd1     8:49   1   7.3G  0 part
mmcblk0   179:0  0 14.5G  0 disk
└─mmcblk0p1 179:1  0 14.5G  0 part
nvme0n1   259:0  0 238.5G 0 disk
└─nvme0n1p1 259:1  0 100M  0 part
└─nvme0n1p2 259:2  0   16M  0 part
└─nvme0n1p3 259:3  0 237.8G 0 part
└─nvme0n1p4 259:4  0  543M  0 part
```

Figure 13: Salida de lsblk

Obtenemos información de la evidencia de la misma forma que en el primer ejercicio, solo que esta vez usaremos sólamente el comando **usb-devices** ya que llegamos anteriormente a la conclusión de que es una mejor opción.

```

ubuntu@ubuntu:~$ usb-devices

T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=480 MxCh=16
D: Ver= 2.00 Cls=09(hub ) Sub=00 Prot=01 MxPS=64 #Cfgs= 1
P: Vendor=1d6b ProdID=0002 Rev=06.05
S: Manufacturer=Linux 6.5.0-18-generic xhci-hcd
S: Product=xHCI Host Controller
S: SerialNumber=0000:00:14.0
C: #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 4 Ivl=256ms

T: Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#= 6 Spd=480 MxCh= 0
D: Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1
P: Vendor=0718 ProdID=0619 Rev=01.00
S: Manufacturer=Imation
S: Product=Classic
S: SerialNumber=07033C2B9E0AB799
C: #Ifs= 1 Cfg#= 1 Atr=80 MxPwr=200mA
I: If#= 0 Alt= 0 #EPs= 2 Cls=08(stor.) Sub=06 Prot=50 Driver=usb-storage
E: Ad=02(0) Atr=02(Bulk) MxPS= 512 Ivl=0ms
E: Ad=81(I) Atr=02(Bulk) MxPS= 512 Ivl=0ms

T: Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=10000 MxCh= 8
D: Ver= 3.10 Cls=09(hub ) Sub=00 Prot=03 MxPS= 9 #Cfgs= 1
P: Vendor=1d6b ProdID=0003 Rev=06.05
S: Manufacturer=Linux 6.5.0-18-generic xhci-hcd
S: Product=xHCI Host Controller
S: SerialNumber=0000:00:14.0
C: #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 4 Ivl=256ms

```

Figure 14: Salida usb-devices

Una vez hecho todo esto, realizamos el hash de la evidencia de la misma forma que antes, obteniendo el mismo resultado y garantizando su integridad.

```

ubuntu@ubuntu:~$ sudo sha256sum /dev/sdc
52cd2c92c8535982ca8bc8f4593212d30780cf3bb2c5e824a827a1e364530c43  /dev/sdc

```

Figure 15: Cálculo del hash del pendrive

## 2.b

Ahora, realizaremos el clonado de la imagen. Para este apartado usaremos la herramienta ***dd*** en vez de ***dc3dd***, utilizada previamente. Para esto, emplearemos el siguiente comando:

```
sudo dd if=/dev/sdc of=/media/ubuntu/24D7E65807DD24BD/forense/pen_ubuntu.dd
```

```

ubuntu@ubuntu:~$ sudo dd if=/dev/sdc of=/media/ubuntu/24D7E65807DD24BD/forense/pen_ubuntu.dd
15261696+0 registros leídos
15261696+0 registros escritos
7813988352 bytes (7,8 GB, 7,3 GiB) copied, 864,248 s, 9,0 MB/s
ubuntu@ubuntu:~$ sudo sha256sum /media/ubuntu/24D7E65807DD24BD/forense/pen_ubuntu.dd

```

Figure 16: Creación de la imagen.

Una vez conseguida la imagen, realizaremos el hash de la misma:

```
ubuntu@ubuntu:~$ sudo sha256sum /media/ubuntu/24D7E65807DD24BD/forense/pen_ubuntu.dd
52cd2c92c8535982ca8bc8f4593212d30780cf3bb2c5e824a827a1e364530c43  /media/ubuntu/24D7E65807DD24BD/forense/pen_ubuntu.dd
```

Figure 17: Cálculo del hash de la imagen.

El hash se sigue manteniendo intacto.

## 2.c

Las diferencias que hemos podido observar entre estas dos herramientas son:

- **Finalidad:** Mientras que el comando dd se utiliza para copiar y convertir datos en Linux, dc3dd es una versión mejorada de dd que aporta más herramientas enfocadas al análisis forense.
- **Velocidad:** La herramienta dc3dd tarda más en ejecutarse que la herramienta dd, probablemente esto se deba a que incluye más funcionalidades.
- **Opciones:** Al ser una versión mejorada, dc3dd incluye más opciones, algunas de estas son:
  - **hash:** Puedes indicar el algoritmo con el que quieras calcular el hash de la imagen que estás creando.
  - **log:** Puedes generar un log e indicar su destino en la máquina.
  - **hashlog:** Genera un log del cálculo de hash durante el proceso de clonado.
  - **progress:** Muestra el progreso del clonado.

Tanto la opción hash como la opción log han sido usadas en esta práctica.

## 3

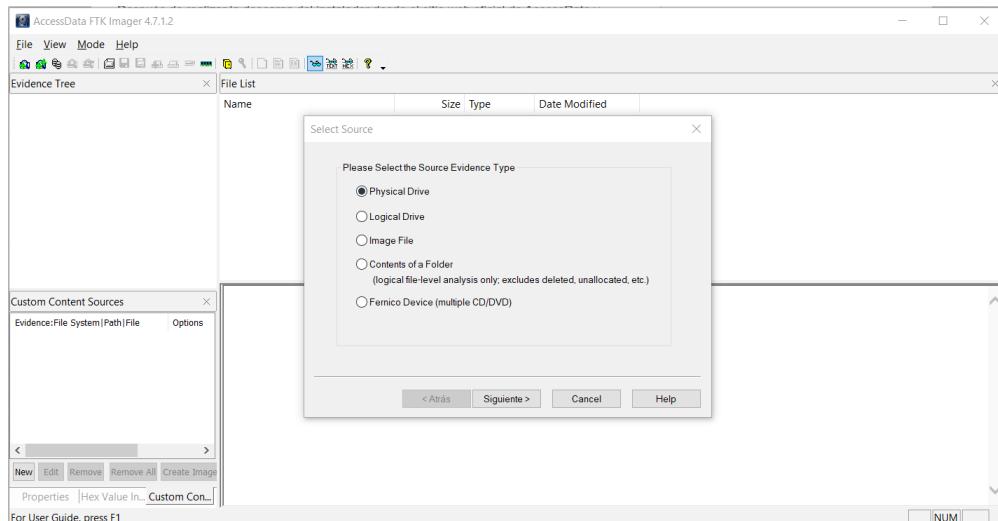
### 3.a

Estos pasos ya han sido realizados y documentados en profundidad en el apartado indicado

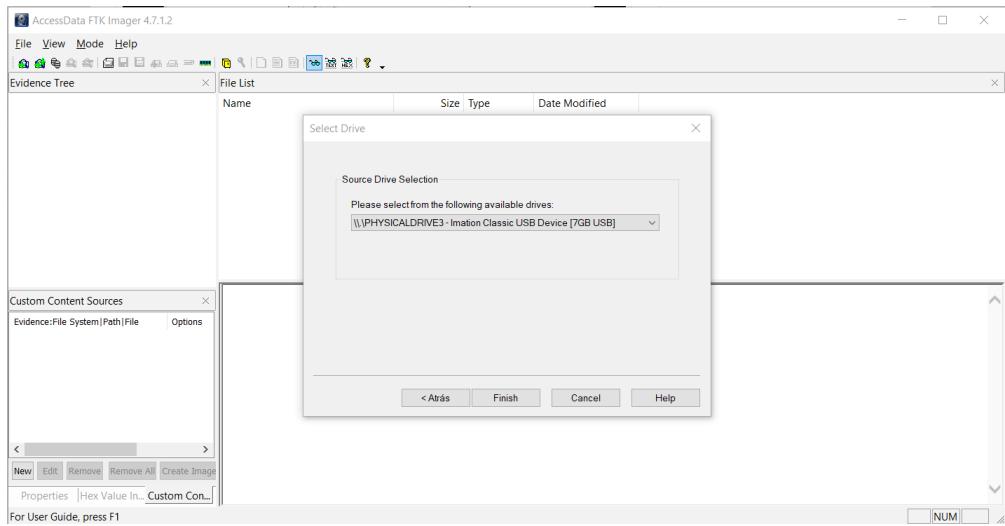
### 3.b

Usaremos la herramienta FTK Imager para llevar a cabo el clonado en Windows. A continuación, ilustraremos los pasos a seguir.

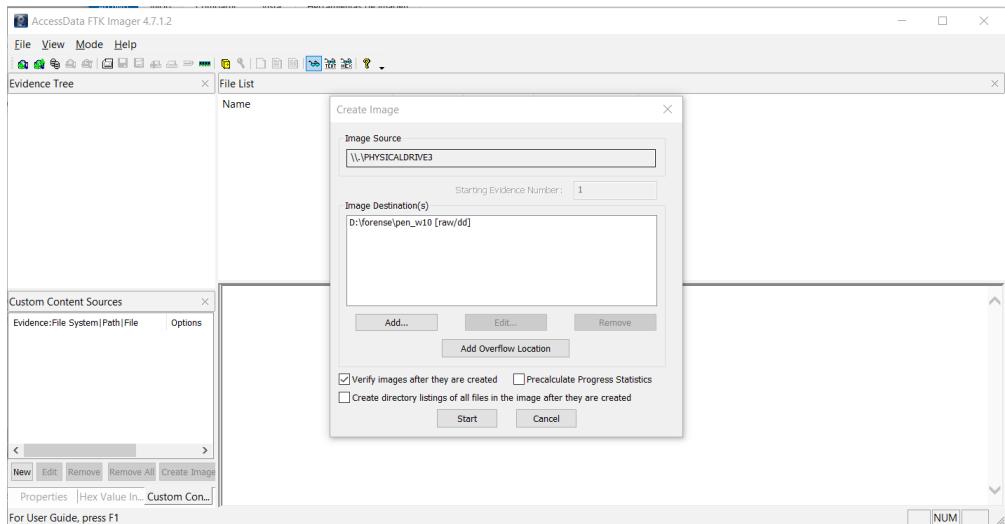
Creamos una imagen de disco en **File - Create Disk Image**:



Y seleccionamos la evidencia:



Seleccionamos el tipo de la imagen y el destino:



Le damos a start y se creará la imagen. Una vez terminado el proceso, se nos mostrará el hash de la imagen en MD5 Y SHA1.

Drive/Image Verify Results	
Sector count	15261696
MD5 Hash	
Computed hash	794f17e49b37a078628a3422541c8a78
Report Hash	794f17e49b37a078628a3422541c8a78
Verify result	Match
SHA1 Hash	
Computed hash	5122a4f93e9d10f8f105ff88ced0ed621c
Report Hash	5122a4f93e9d10f8f105ff88ced0ed621c
Verify result	Match
Bad Blocks List	
Bad block(s) in image	No bad blocks found in image
Close	

Figure 18: Hash MD5 y SHA1

Ahora solo falta calcular el hash en SHA256, como lo llevamos haciendo durante toda la práctica. Para ello, en PowerShell ejecutaremos:

```
Get-FileHash -Algorithm SHA256 F:\forense\pen_w10
```

PS C:\Users\Andrea> Get-FileHash -Algorithm SHA256 F:\forense\pen_w10		
Algorithm	Hash	Path
SHA256	52CD2C92C8535982CA8BC8F4593212D30780CF3BB2C5E824A827A1E364530C43	F:\forense\pen_w10

Figure 19: Hash de la imagen en Windows

### 3.c

#### Ventajas:

- **Interfaz gráfica:** La herramienta FTK ofrece una interfaz gráfica que hace que su uso sea más cómodo para el usuario.
- **Especializada en análisis forense:** Esta herramienta está especializada en análisis forense, a diferencia de dd que simplemente está dedicada a la copia de datos.
- **Sencillez:** El enfoque en análisis forense junto a la interfaz gráfica hace que esta herramienta sea más sencilla de usar para las personas interesadas en el análisis forense que otras opciones.
- **Compatibilidad nativa con Windows:** FTK está diseñado específicamente para sistemas Windows, lo que significa que puede integrarse más fácilmente con el entorno de trabajo de los usuarios de Windows

#### Inconvenientes:

- **Dependencia de la interfaz gráfica:** Aunque la interfaz gráfica puede suponer una ventaja para muchos, para otros supone un inconveniente ya que no hay opción de línea de comandos, preferida por algunos usuarios.
- **Limitaciones en la ejecución en entornos no Windows:** FTK está diseñado principalmente para sistemas Windows y puede tener limitaciones o problemas de compatibilidad al ejecutarse en entornos no Windows.

## 4

Para este ejercicio, se ha empleado un pendrive distinto al utilizado en los ejercicios previos. Adjuntamos en el documento algunas fotografías del dispositivo utilizado:



Figure 20: Parte delantera del pendrive



Figure 21: Parte trasera del pendrive

En el exterior del pendrive no se nos muestra su tamaño. Sin embargo, sabemos que se trata de un pendrive de 32 GB. Este pendrive es de un tamaño considerable para el análisis en nuestros ordenadores, por lo que se trata de un perfecto ejemplo para ilustrar casos en los que carecemos de tiempo o espacio de almacenamiento adecuado, lo que nos llevaría a optar por un clonado parcial.

#### 4.a

Primero ejecutamos `lsblk` para ver las particiones:

```
sansforensics@siftworkstation: ~
$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0   62M  1 loop /snap/core20/1587
loop1      7:1    0    4K  1 loop /snap/bare/5
loop2      7:2    0 159.6M 1 loop /snap/chromium/2738
loop3      7:3    0  74.1M 1 loop /snap/core22/1033
loop4      7:4    0  66.5M 1 loop /snap/cups/1024
loop5      7:5    0  497M 1 loop /snap/gnome-42-2204/141
loop6      7:6    0  91.7M 1 loop /snap/gtk-common-themes/1535
loop7      7:7    0  79.9M 1 loop /snap/lxd/22923
loop8      7:8    0  40.4M 1 loop /snap/snapd/20671
sda
└─sda1     8:0    0   1M  0 part
└─sda2     8:1    0    2G  0 part /boot
└─sda3     8:2    0  486.3G 0 part
  └─ubuntu--vg-ubuntu--lv 253:0  0 100G 0 lvm /
```

Figure 22: Salida de `lsblk`

Después, mediante el comando

```
sudo fdisk /dev/sdb
```

eliminamos la partición existente con la opción **d** y creamos las nuevas con la opción **n**.

```
Command (m for help): d
Selected partition 1
Partition 1 has been deleted.
```

Figure 23: Borrado de la partición existente

```
Command (m for help): n
Partition type
  p  primary (0 primary, 0 extended, 4 free)
  e  extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
```

Figure 24: Creación de una partición

Formateamos las particiones para poder hacer uso de ellas.

```
sudo mkfs.ext4 /dev/sdb1
sudo mkfs.ext4 /dev/sdb2
```

```
sansforensics@siftworkstation: ~
$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 5120 4k blocks and 5120 inodes
Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

sansforensics@siftworkstation: ~
$ sudo mkfs.ext4 /dev/sdb2
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 5120 4k blocks and 5120 inodes
Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done
```

Figure 25: Formateo de las particiones

Podemos ver las particiones:

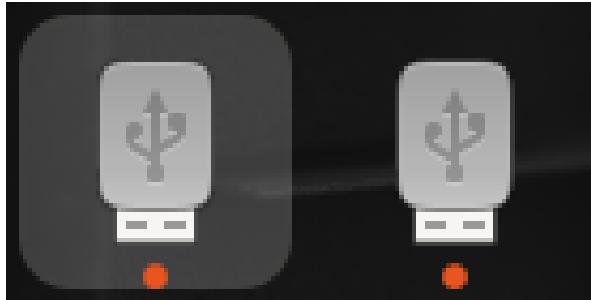


Figure 26: Particiones

Añadimos nuevos ficheros a ambos:

```
sansforensics@siftworkstation: /media/sansforensics/f46255d1-6fb3-4974-9586-feefaa2499ae
e
$ sudo nano fichero1
sansforensics@siftworkstation: /media/sansforensics/f46255d1-6fb3-4974-9586-feefaa2499ae
e
$ Documents
$ sudo nano fichero2
sansforensics@siftworkstation: /media/sansforensics/f46255d1-6fb3-4974-9586-feefaa2499ae
e
$ sudo nano fichero3
```

Figure 27: Añadido de ficheros

```
sansforensics@siftworkstation: /media/sansforensics/3915fe12-3d44-4baa-ad09-19b718d092d
7
$ sudo nano fichero4
sansforensics@siftworkstation: /media/sansforensics/3915fe12-3d44-4baa-ad09-19b718d092d
7
$ Documents
$ sudo nano fichero5
sansforensics@siftworkstation: /media/sansforensics/3915fe12-3d44-4baa-ad09-19b718d092d
7
$ sudo nano fichero6
```

Figure 28: Añadido de ficheros

#### 4.b

Para calcular el hash tendremos que utilizar los primeros 41 MB de /dev/sdb, 1 de la tabla de particiones y 40 de las particiones que acabamos de crear. Para ello, emplearemos el siguiente comando indicándole con -c que muestre los primeros 41 megabytes.

```
sudo head -c 41M /dev/sdb | sha256sum
```

```
sansforensics@siftworkstation: /
$ sudo head -c 41M /dev/sdb | sha256sum
d3d00572c0be759c7b9d136d0073e2cf4681048bc614f38e1f52d0ba5ea73186
```

Figure 29: Cálculo del hash sobre la evidencia

#### 4.c

Realizamos la imagen parcial del tamaño calculado con el comando:

```
sudo dd if=/dev/sdb of=pen_parcial.dd bs=1M count=41
```

```
sansforensics@siftworkstation: /  
$ sudo dd if=/dev/sdb of=pen_parcial.dd bs=1M count=41  
41+0 records in  
41+0 records out  
42991616 bytes (43 MB, 41 MiB) copied, 0.148819 s, 289 MB/s
```

Figure 30: Realización de la imagen parcial

Calculamos el hash de la imagen, y podemos ver como coincide con el obtenido anteriormente:

```
sansforensics@siftworkstation: /  
$ sudo sha256sum pen_parcial.dd  
d3d00572c0be759c7b9d136d0073e2cf4681048bc614f38e1f52d0ba5ea73186 pen_parcial.dd
```

Figure 31: Cálculo del hash de la imagen parcial

#### 4.d

Borramos un fichero en ambas particiones:

```
sansforensics@siftworkstation: /media/sansforensics/f46255d1-6fb3-4974-9586-feefa2499ae  
e  
$ sudo rm fichero3
```

Figure 32: Borrado de un fichero en la primera partición

```
sansforensics@siftworkstation: /media/sansforensics/3915fe12-3d44-4baa-ad09-19b718d092d  
7  
$ sudo rm fichero6
```

Figure 33: Borrado de un fichero en la segunda partición

Calculamos el hash de la misma forma que antes:

```
sansforensics@siftworkstation: /  
$ sudo head -c 41M /dev/sdc | sha256sum  
5d1e14d3f74ba034e6f43001904052b5a4012751fabe1cf13472d4d3a1a2334a -
```

Figure 34: Cálculo del hash sobre la evidencia

Como podemos ver ha cambiado, esto se debe a que hemos alterado el dispositivo eliminando archivos en su interior y alterando así el estado en el que realizamos previamente el hash.

#### 4.e

Realizamos una imagen parcial después de haber eliminado ficheros en ambas particiones:

```
sansforensics@siftworkstation: /  
$ sudo dd if=/dev/sdc of=pen_borrado_ficheros.dd bs=1M count=41  
41+0 records in  
41+0 records out  
42991616 bytes (43 MB, 41 MiB) copied, 11.3003 s, 3.8 MB/s
```

Figure 35: Realización de la imagen parcial

Y calculamos el hash de dicha imagen:

```
sansforensics@siftworkstation: /  
$ sudo sha256sum pen_borrado_ficheros.dd  
5d1e14d3f74ba034e6f43001904052b5a4012751fabe1cf13472d4d3a1a2334a pen_borrado_ficheros.  
dd
```

Figure 36: Cálculo del hash de la imagen parcial

#### 4.f

Eliminamos la segunda partición creada anteriormente:

```
sansforensics@siftworkstation: /  
$ sudo fdisk /dev/sdc  
  
Welcome to fdisk (util-linux 2.37.2).  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.  
  
Command (m for help): d  
Partition number (1,2, default 2): 2  
Deleting partition 2  
Writing superblocks and filesystem accounting information: done  
Partition 2 has been deleted.
```

Figure 37: Eliminación de la segunda partición

Calculamos de nuevo el hash:

```
sansforensics@siftworkstation: /  
$ sudo head -c 21M /dev/sdc | sha256sum  
d6184cb4efe0b22e000672608aaae2dadd28fff2a69a174c63f32022298594e7 -
```

Figure 38: Cálculo del hash sobre la evidencia

Al haber alterado de nuevo el dispositivo, el hash ha vuelto a cambiar.

#### 4.g

Realizamos la imagen después de haber borrado la partición:

```
sansforensics@siftworkstation: /  
$ sudo dd if=/dev/sdc of=pen_borrado_particion.dd bs=1M count=21  
21+0 records in  
21+0 records out  
22020096 bytes (22 MB, 21 MiB) copied, 5.94868 s, 3.7 MB/s
```

Figure 39: Realización de la imagen parcial

Calculamos el hash de la imagen:

```
sansforensics@siftworkstation: /  
$ sudo sha256sum pen_borrado_particion.dd  
d6184cb4efe0b22e000672608aaae2dadd28fff2a69a174c63f32022298594e7 pen_borrado_particion  
.dd
```

Figure 40: Cálculo del hash de la imagen

Podemos ver que es el mismo hash que el calculado anteriormente.

#### 4.h

Para poder montar las particiones, primero tenemos que asociar dispositivos de bucle para poder acceder a las imágenes como si fueran dispositivos. Esto lo haremos gracias a la herramienta **losetup**. Con la opción **-o** indicaremos desde donde leer el archivo. En este caso será a partir de 1 MB, ya que ese espacio pertenece a la tabla del sistema de particiones. Lo tendremos que indicar en bytes.

```
sudo losetup -o 1048576 /dev/loop15 pen_parcial.dd  
sudo losetup -o 1048576 /dev/loop16 pen_borrado_ficheros.dd  
sudo losetup -o 1048576 /dev/loop17 pen_borrado_particion.dd
```

```
$ sudo losetup -o 1048576 /dev/loop15 pen_parcial.dd  
sansforensics@siftworkstation: /  
$ sudo losetup -o 1048576 /dev/loop16 pen_borrado_ficheros.dd  
sansforensics@siftworkstation: /  
$ sudo losetup -o 1048576 /dev/loop17 pen_borrado_particion.dd
```

Figure 41: Asociación de los dispositivos de bucle

Para las segundas particiones, tendremos que indicarle a la opción **-o** que empiece a leer a partir de los 21 MB, que se corresponden con el primer megabyte de la tabla del sistema de particiones y los veinte restantes de la primera partición.

```
sansforensics@siftworkstation: /  
$ sudo losetup -o 22020096 /dev/loop18 pen_parcial.dd
```

Figure 42: Asociación de los dispositivos de bucle

```
sansforensics@siftworkstation: /  
$ sudo losetup -o 22020096 /dev/loop19 pen_borrado_ficheros.dd
```

Figure 43: Asociación de los dispositivos de bucle

Ahora, montaremos las particiones con el comando **mount**

```
sudo mount /dev/loop15 /mnt/pen_parcial
sudo mount /dev/loop16 /mnt/pen_borrado_ficheros
sudo mount /dev/loop17 /mnt/pen_borrado_particion
sudo mount /dev/loop18 /mnt/pen_parcial2
sudo mount /dev/loop19 /mnt/pen_borrado_ficheros2
```

```
sansforensics@siftworkstation: /
$ sudo mount /dev/loop15 /mnt/pen_parcial
sansforensics@siftworkstation: /
$ sudo mount /dev/loop16 /mnt/pen_borrado_ficheros
sansforensics@siftworkstation: /
$ sudo mount /dev/loop17 /mnt/pen_borrado_particion
sansforensics@siftworkstation: /
$ sudo mount /dev/loop18 /mnt/pen_parcial2
sansforensics@siftworkstation: /
$ sudo mount /dev/loop19 /mnt/pen_borrado_ficheros2
```

Figure 44: Montado de las particiones

Después de montar las particiones, tendremos acceso a su contenido. Vamos a mostrar el contenido que hemos conseguido de cada una de ellas:

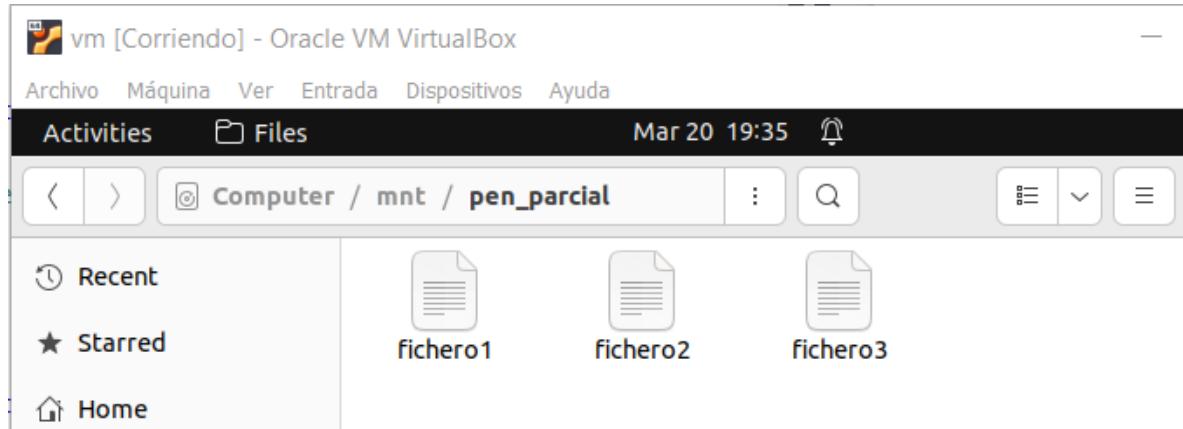


Figure 45: Enter Caption

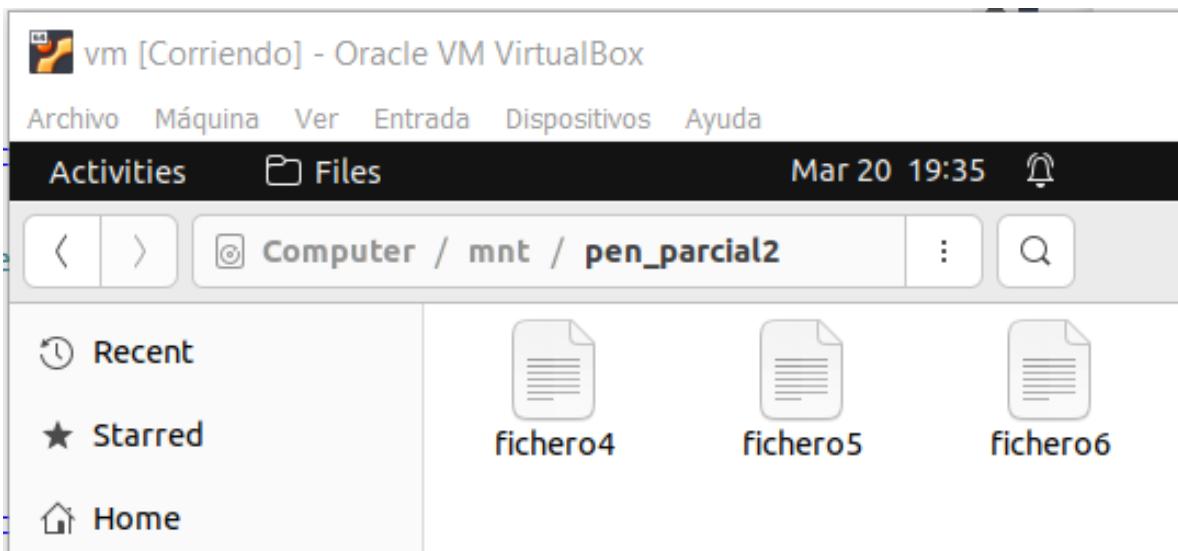


Figure 46: Enter Caption

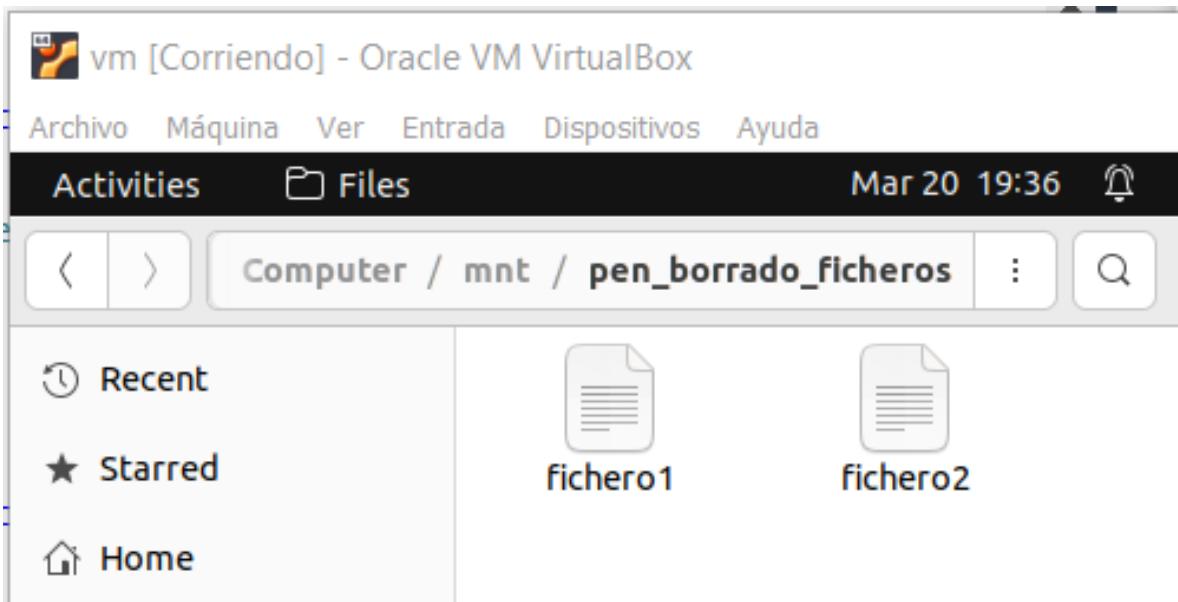


Figure 47: Enter Caption



Figure 48: Enter Caption



Figure 49: Enter Caption

Como podemos observar, el contenido coincide con el que había justo en el momento de realizar las imágenes, por lo que lo hemos recuperado con éxito.

## 5

### 5.a

En clase hemos visto 3 modelos importantes respecto al clonado hardware.

#### StarTech SATDOCK2REU3

Este dispositivo<sup>2</sup> es un duplicador y borrador de disco duro de 2 bahías de discos duros SATA HD-D/SSD. Es una de los más baratos del mercado



- 2 discos duros SATA
- SATA I/II/III de 2,5
- independencia del sistema operativo
- Tasa de clonado: 11 GB/min
- USB 3.2 (5 Gbps)
- Precio: 119,78 EUR
- HASH?¿?¿?¿?

---

<sup>2</sup>StarTech: SATDOCK2REU3

## Tableau Forensic Duplicator TD2u

Este dispositivo<sup>3</sup> es de una gama superior al anterior como se puede ver en sus características



- 2 discos duros SATA
- SATA I/II/III
- Hash (MD5 or SHA-1)
- Imagen: hasta 15 GB/minute
- Clonado: 25GB/minute
- USB 3.0, 2.0 y 1.1

Nosotros hemos podido ver esta clonadora montada en el taller "El inicio del procedimiento forense informático: adquisición de evidencias en PCs, móviles y pendrives USB", impartido por Rafael López García. Adjuntamos una imagen a continuación:

---

<sup>3</sup>[Opentext security: Tableau Forensic Duplicator TD2u](#)

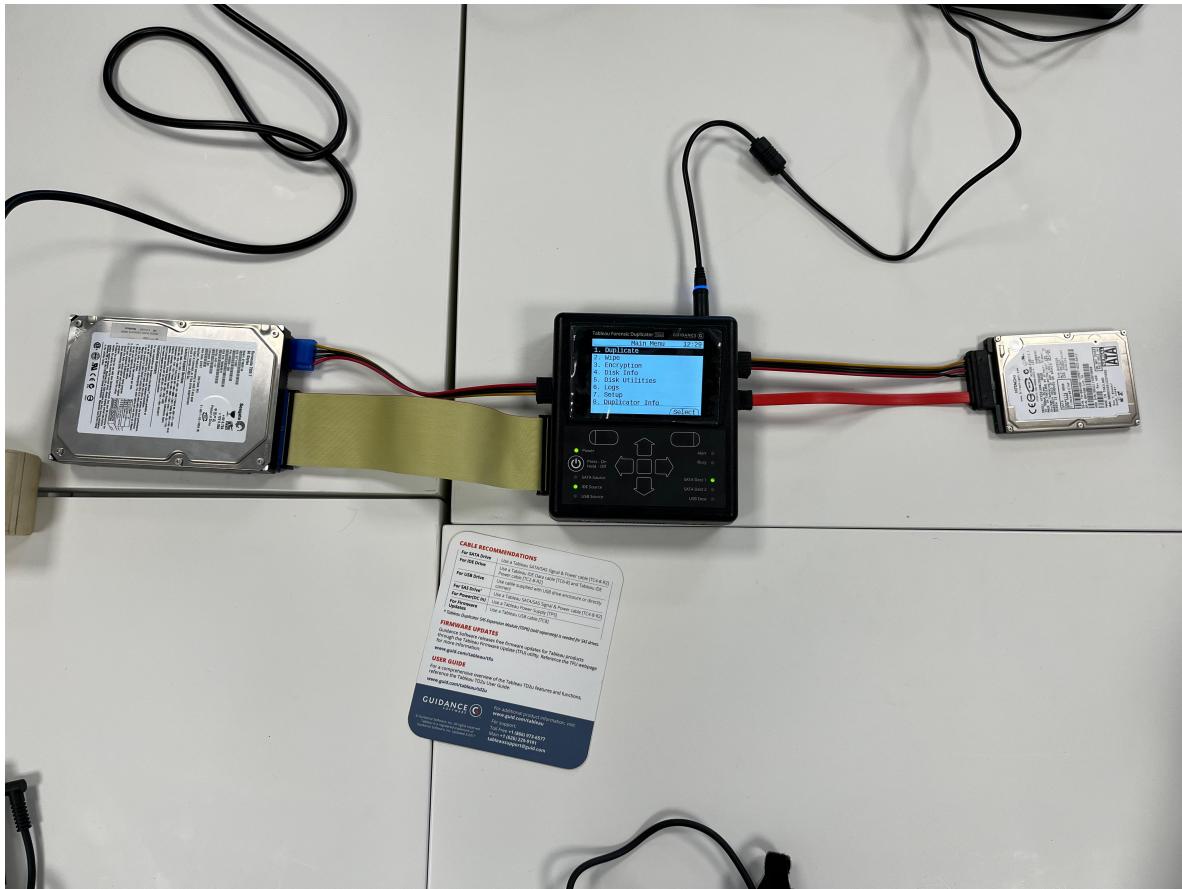


Figure 50: Clonadora montada

### Atola Taskforce

Esta clonadora<sup>4</sup> es una de los dispositivos mas potentes del mercado como se puede intuir.



- 4 NVMe M.2/U.2 PCIe 4.0

---

<sup>4</sup> Atola Taskforce

- 8 SATA
- 8 SATA/SAS
- 4 USB
- IDE
- Extension slot (for Atola Thunderbolt, Apple PCIe SSD and M.2 NVMe/PCIe/SATA SSD extension modules)
- Hash: MD5, SHA-1, SHA-256, SHA-512
- Tipo de hash: lineal (ideal para buenos discos duros) o segmentado (ideal para discos duros dañados)
- Con interfaz de usuario
- 25 TB/hora

### Ventajas e inconvenientes frente a soluciones software

#### Ventajas:

- **Velocidad:** Las clonadoras hardware suelen ser más rápidas ya que operan directamente en el nivel de hardware, evitando la sobrecarga del sistema operativo y el software adicional.
- **Independencia del sistema operativo:** No dependen del sistema operativo en el que se estén ejecutando, lo que las hace útiles para clonar sistemas operativos diferentes o para sistemas en los que el software no puede ser instalado.
- **Facilidad de uso:** Algunas clonadoras hardware vienen con interfaces simples y fáciles de usar que no requieren conocimientos técnicos profundos.

#### Inconvenientes:

- **Costo:** Las clonadoras hardware suelen ser más caras que las soluciones de software.
- **Limitaciones de compatibilidad:** Pueden no ser compatibles con todos los tipos de dispositivos o sistemas de archivos.

## 5.b

### Guardonix

Este dispositivo<sup>5</sup> es una de los bloqueadores de escritura más conocidos del mercado.




---

<sup>5</sup> Guardonix USB 3.0

- Logging
- Ayuda a evitar que Windows se congele, se bloquee y/o deje caer dispositivos de almacenamiento USB debido a sectores defectuosos u otros problemas de inestabilidad de lectura.
- Works bus powered with all storage devices except the most power hungry HDDs
- Permite conectar la unidad fuente a un puerto USB sin alimentación, lo que reduce el riesgo de daños por cortocircuitos y descargas electrostáticas.
- Premium: Funciones avanzadas de registro, desbloqueo de contraseñas ATA y WD Smartware, acceso a áreas de datos ocultos HPA y DCO, visualización gráfica de la velocidad, informes de errores USB-SCSI, información completa sobre el estado de los sectores y mucho más.
- Premium: Permite buscar palabras clave específicas en los datos que pasan por el dispositivo en tiempo real y, en la mayoría de los casos, sin reducir la velocidad de procesamiento de imágenes.
- Precio: 320 €
- Precio Premium: +470 €
- Precio set de adaptadores: 200 €

### **Tableau Forensic SATA/IDE Bridge**

Este bloqueadora de escritura<sup>6</sup>



- USB 3.0 (compatible con versiones anteriores USB de alta/alta/baja velocidad)
- Microsoft Windows versión 7, 8, 10
- Dispositivos de disco duro ATA paralelos compatibles con el direccionamiento lógico de bloques (LBA)
- Dispositivos de disco duro SATA 1 o SATA 2

---

<sup>6</sup>[Tableau Forensic SATA/IDE Bridge](#)

### USB 3.1 WriteBlocker

Este dispositivo<sup>7</sup> es otra bloqueadora de escritura muy conocida y empleada por forenses.



- Windows 10 y 11 (nota: no se admiten máquinas virtuales)
- Dos USB 3.2 Gen 2 Tipo-C, un USB 3.0 Tipo-A
- Dos USB 3.2 Gen 2 Tipo-C: hasta 10 Gbps

### Ventajas e inconvenientes frente a soluciones software

#### Ventajas:

- **Seguridad Física:** Ofrecen una protección física contra la modificación o eliminación accidental de datos.
- **Independencia del Sistema Operativo:** Funcionan independientemente del sistema operativo, lo que las hace compatibles con una amplia gama de dispositivos.
- **Facilidad de Uso:** Por lo general, son fáciles de usar y no requieren instalación ni configuración compleja.
- **Protección Contra Malware:** Al ser dispositivos físicos, son menos susceptibles a los ataques de malware que las soluciones de software.

#### Inconvenientes:

- **Costo:** Suelen ser más caras que las soluciones de software.
- **Portabilidad Limitada:** Pueden ser menos convenientes para usuarios que necesitan moverse frecuentemente entre dispositivos.
- **Limitaciones de Funcionalidad:** Algunas pueden carecer de funciones avanzadas disponibles en soluciones de software, como la colaboración en tiempo real o la integración con otros programas.

---

<sup>7</sup>Wiebe Tech: USB 3.1 WriteBlocker