

# Análisis Forense

## **ADQUISICIÓN DE EVIDENCIAS EN CALIENTE.**

Jennifer

## Índice

|  |    |
|--|----|
| Índice   | 2  |
| Introducción   | 3  |
| Requisitos para realizar la práctica                       | 3  |
| Preparación del entorno                                    | 4  |
| Máquina virtual window                                     | 4  |
| Máquina virtual Linux                                      | 4  |
| Extracción de evidencias digitales volátiles (RAM) Windows | 5  |
| OSF  | 7  |
| Calcular el hash de las evidencias                         | 10 |
| FTK imager   | 10 |
| Calcular el hash de las evidencias                         | 12 |
| DumpIt   | 12 |
| Calcular el hash de las evidencias                         | 14 |
| Calcular hash usando sha512sum                             | 15 |
| Extracción de evidencias digitales volátiles (RAM) Linux   | 16 |
| Microsoft AVML (linux)                                     | 16 |

## Introducción

aprenderás a realizar la adquisición de evidencias digitales en caliente, es decir, mientras el sistema aún se encuentra encendido y operativo. Se explican diversas herramientas y técnicas para extraer memoria RAM en sistemas Windows y Linux, calcular hashes y preservar la integridad de las evidencias volátiles. Es una guía paso a paso, útil para quienes quieren adentrarse en el análisis forense digital de sistemas activos.

## Requisitos para realizar la práctica

### Entorno virtual:

- VirtualBox u otra herramienta similar.

- Dos máquinas virtuales:

  - Una con Windows.

  - Otra con Linux.

- Disco adicional Debian 12 para almacenar las evidencias.

### Herramientas para Windows:

- OSForensics (OSF)

- FTK Imager

- Dumplt

- Comando: certutil -hashfile

### Herramientas para Linux:

- Kali Linux Live ISO

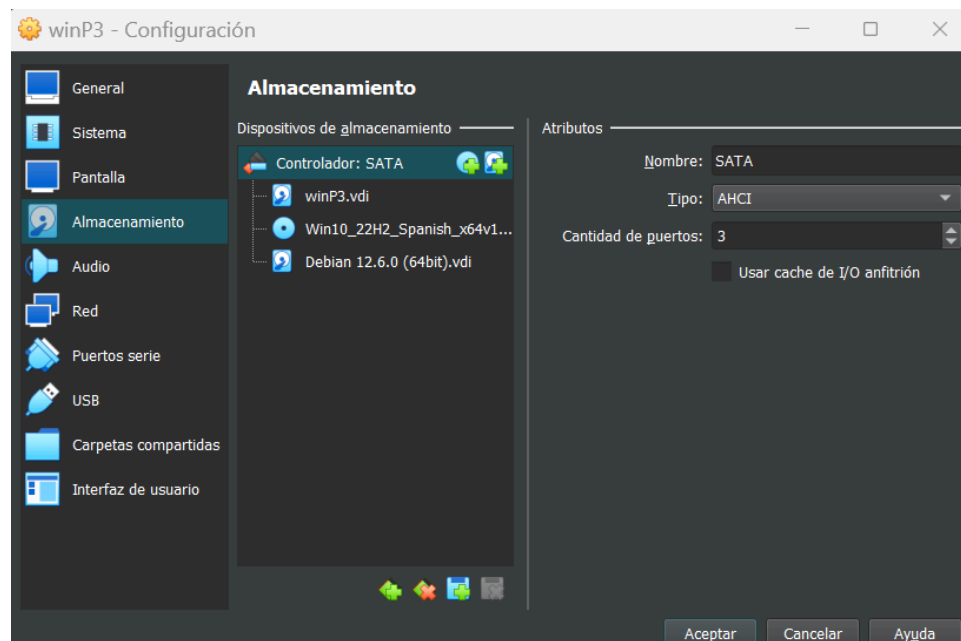
- Microsoft AVML (herramienta en Rust para volcado de RAM)

- Comando: sha512sum, mount, fdisk, ntfs-3g

## Preparación del entorno

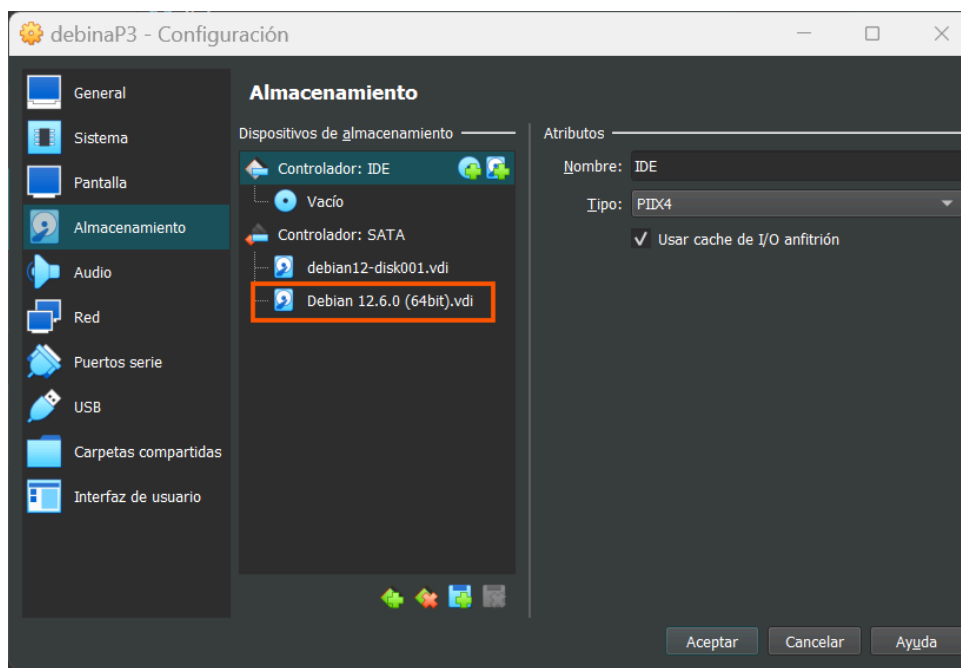
### Máquina virtual window

Agregue un disco duro debían 12 para guardar la evidencia



### Máquina virtual Linux

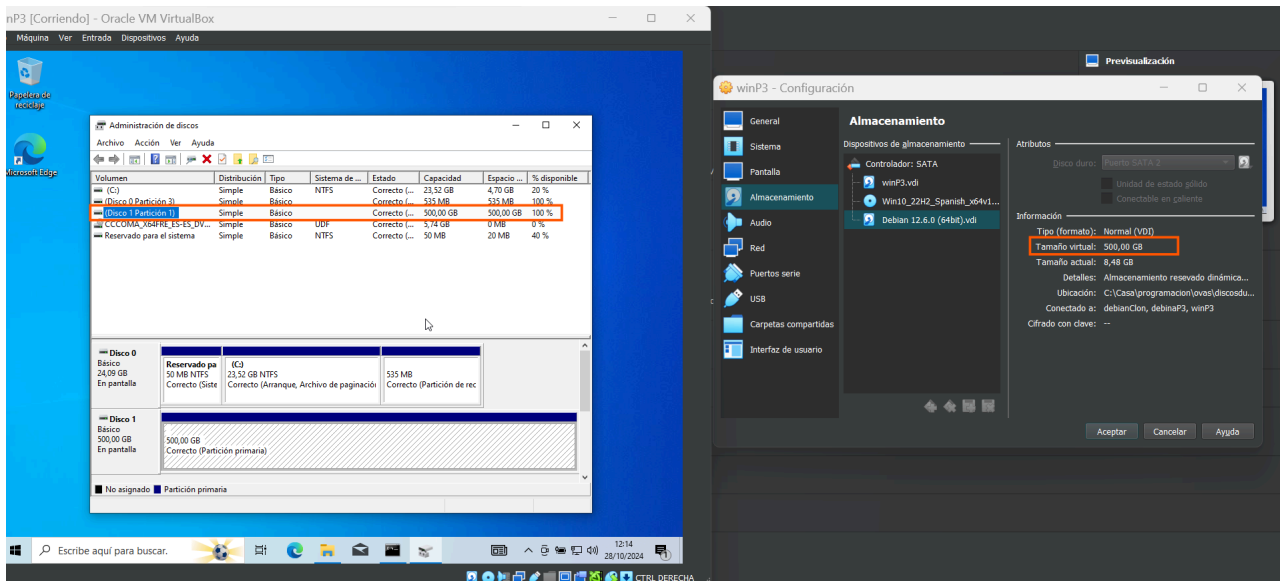
Agregue un disco duro debían 12 para guardar la evidencia



[índice](#)

## Extracción de evidencias digitales volátiles (RAM) Windows

Comprobar que tenga el disco duro que le agregue para agregar ahí las evidencias.



Asignar una letra al disco que agregamos

1. Abrimos la terminal de windows con permisos de administrador y escribimos **diskpart**, listamos los discos que tenemos disponibles **list disk**

```

Administrador: Símbolo del sistema - diskpart
C:\Windows\system32>diskpart

Microsoft DiskPart versión 10.0.19041.964

Copyright (C) Microsoft Corporation.
En el equipo: DESKTOP-3TLHLBI

DISKPART> list disk

Núm Disco  Estado      Tamaño  Disp  Din  Gpt
-----
Disco 0    En línea    24 GB   1024 KB
Disco 1    En línea    500 GB   0 B

DISKPART>
  
```

2. Seleccionar el disco que se le asignará una letra:
  - a. Eliminamos la partición existente **clean**
  - b. Creamos una nueva partition **create partition primary**
  - c. Formateamos la partición con NTFS **format fs=ntfs quick**
  - d. Asignamos la letra **assign letter=E**

Administrador: Símbolo del sistema - diskpart

La partición 1 es ahora la partición seleccionada.  
DISKPART> format fs=ntfs quick  
No hay un volumen seleccionado.  
Seleccione un volumen e inténtelo de nuevo.  
DISKPART> select disk 1  
El disco 1 es ahora el disco seleccionado.  
DISKPART> clean  
DiskPart ha limpiado el disco satisfactoriamente.  
DISKPART> create partition primary  
DiskPart ha creado satisfactoriamente la partición especificada.  
DISKPART> format fs=ntfs quick  
100 por ciento completado  
DiskPart formateó el volumen correctamente.  
DISKPART> assign letter=E  
DiskPart asignó correctamente una letra de unidad o punto de montaje.  
DISKPART>

Administración de discos

Archivo Acción Ver Ayuda

| Volumen                       | Distribución | Tipo   | Sistema de ... | Estado         | Capacidad | Espacio ... | % disp |
|-------------------------------|--------------|--------|----------------|----------------|-----------|-------------|--------|
| (C:)                          | Simple       | Básico | NTFS           | Correcto (...) | 23,52 GB  | 3,49 GB     | 15 %   |
| (E:)                          | Simple       | Básico | NTFS           | Correcto (...) | 500,00 GB | 499,89 GB   | 100 %  |
| (Disco 0 Partición 3)         | Simple       | Básico |                | Correcto (...) | 535 MB    | 535 MB      | 100 %  |
| CCCCOMA_X64FRE_ES-ES_DV9 (D:) | Simple       | Básico | UDF            | Correcto (...) | 5,74 GB   | 0 MB        | 0 %    |
| Reservado para el sistema     | Simple       | Básico | NTFS           | Correcto (...) | 50 MB     | 20 MB       | 40 %   |

Disco 0

Básico  
24,09 GB  
En pantalla

Reservado pa

50 MB NTFS  
Correcto (Siste

(C:)

23,52 GB NTFS  
Correcto (Arranque, Archivo de paginació

535 MB  
Correcto (Partición de rec

Disco 1

Básico  
500,00 GB  
En pantalla

(E:)  
500,00 GB NTFS  
Correcto (Partición primaria)

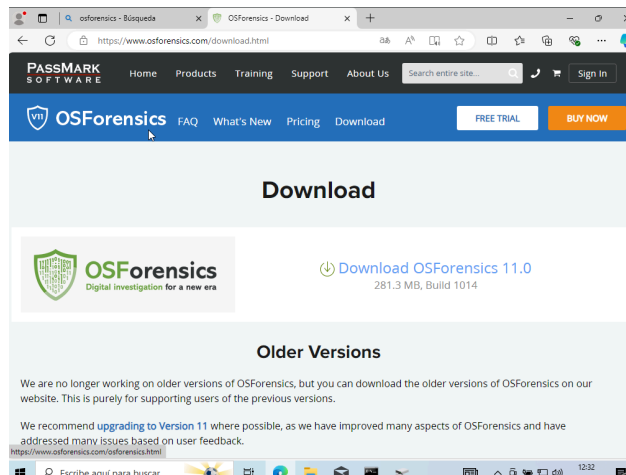
No asignado

Partición primaria

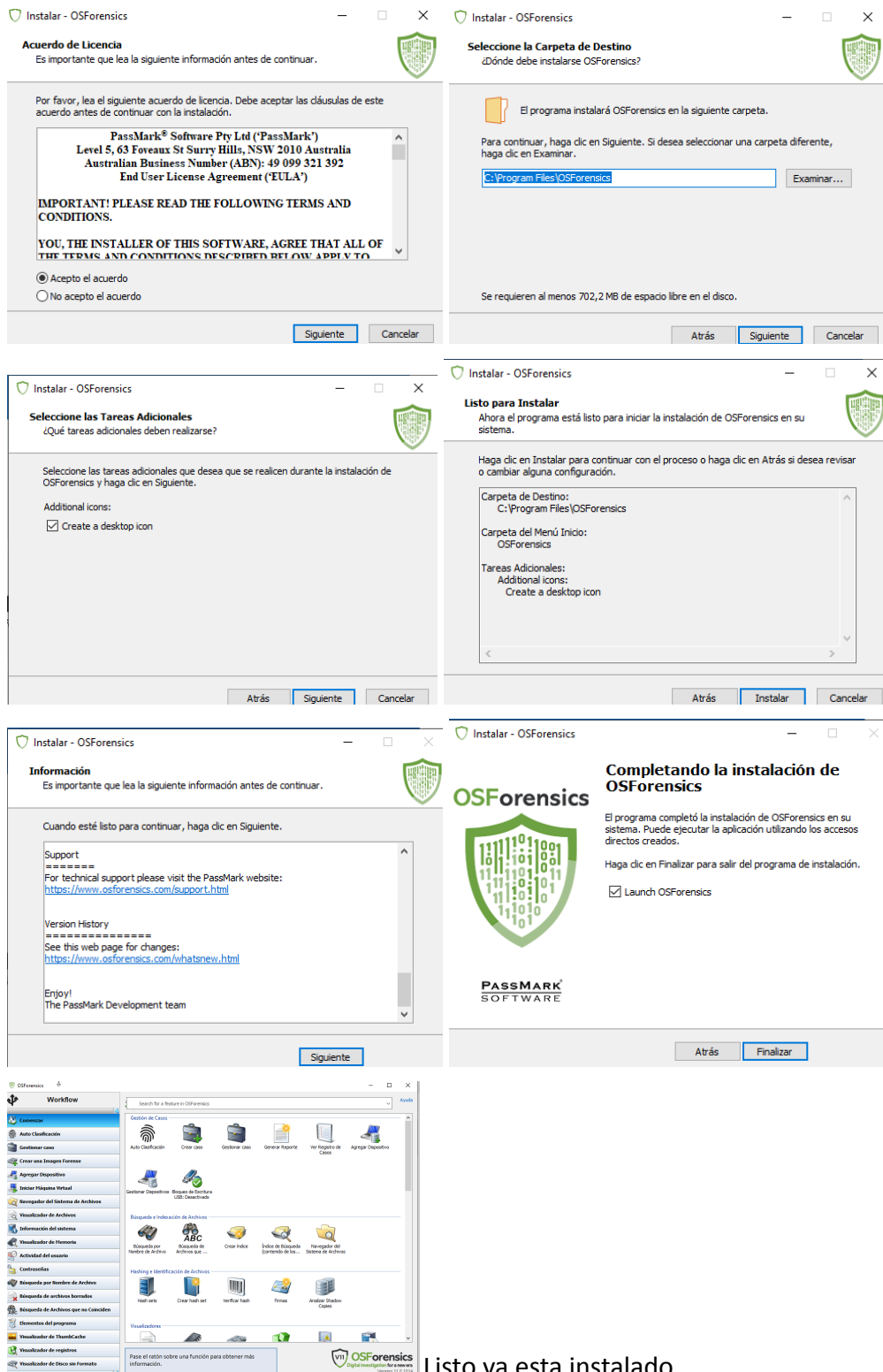
## OSF

Es una herramienta de análisis forense digital desarrollada por PassMark Software. Está diseñada para ayudar a investigadores forenses a recopilar, analizar y gestionar evidencia digital en computadoras y dispositivos de almacenamiento.

### 1. Descargar la versión gratuita



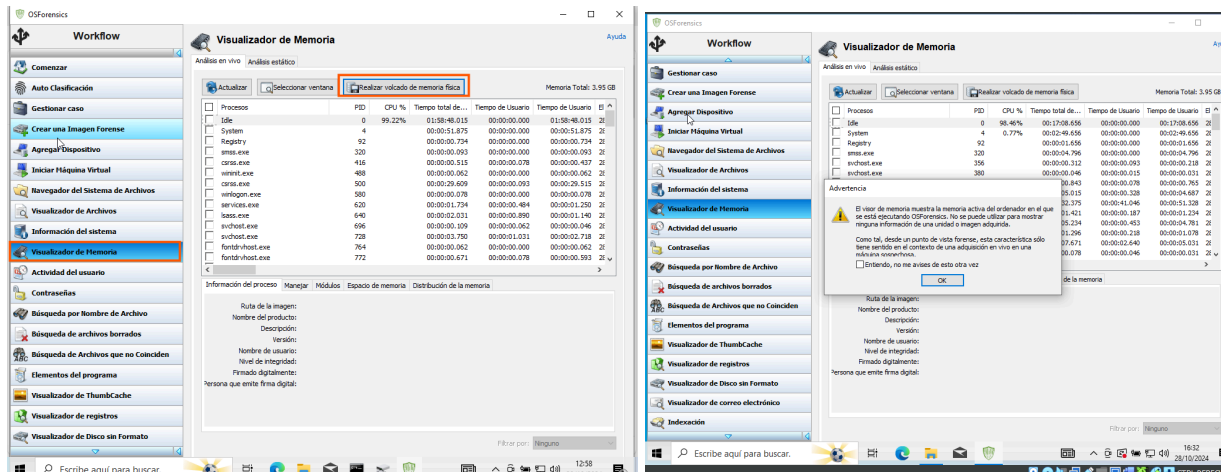
## 2. Instalamos la herramienta



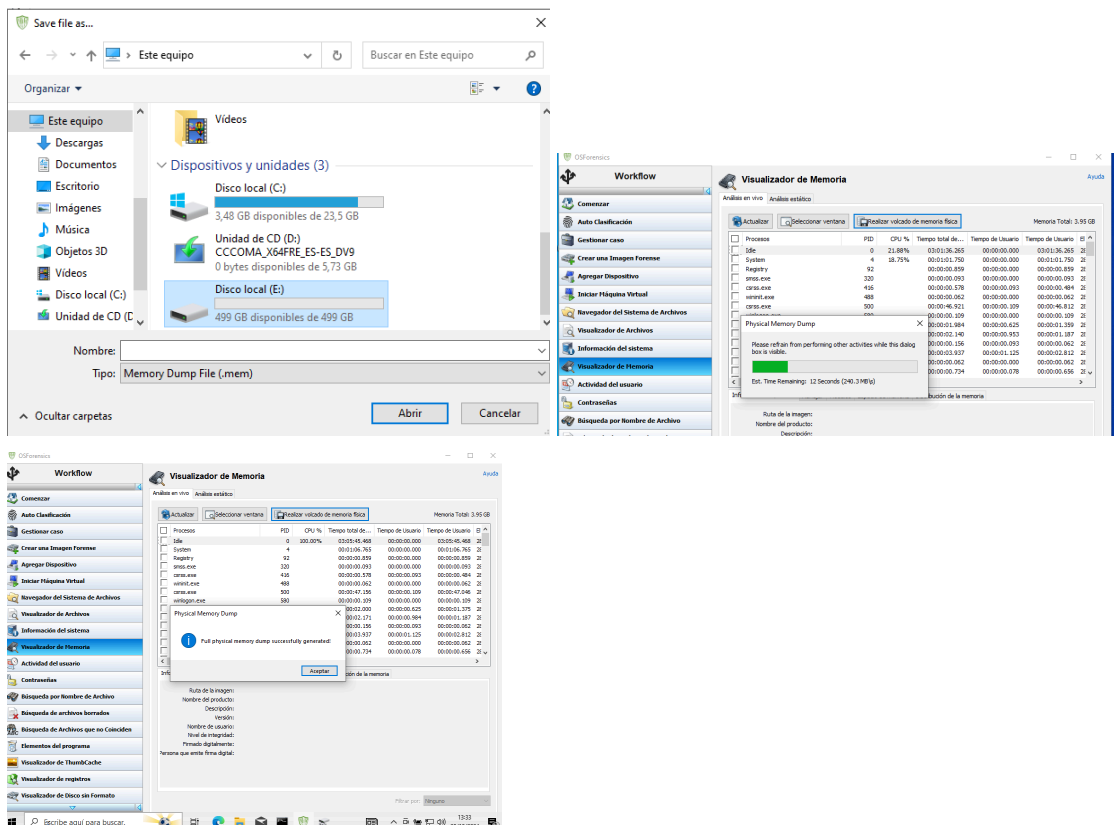
Listo ya esta instalado



### 3. Seleccionar Visualizador de memoria y Realizar volcado de memoria física



### 4. Seleccionar en donde se va a guardar la evidencia



## Calcular el hash de las evidencias

En window utilice el siguiente comando, mencionó que el hash es distinto ya que la memoria RAM es volátil y cambia constantemente mientras el sistema está en funcionamiento

certutil -hashfile

```

Simbolo del sistema
El número de serie del volumen es: E848-0177

Directorio de E:\

8/10/2024 13:31      22 copramosf.cfg
8/10/2024 13:31    4.781.506.560 copramosf.mem
8/10/2024 16:35      22 copraosf2.cfg
8/10/2024 16:35    4.781.506.560 copraosf2.mem
                4 archivos 9.563.013.164 bytes
                0 dirs  527.188.586.496 bytes libres

:\>certutil -hashfile copramosf.mem
HA1 hash de copramosf.mem:
3cc46cd3434bd30b1b9f9aee52b9fcb301f74dc
ertUtil: -hashfile comando completado correctamente.

:\>certutil -hashfile copramosf.mem sha512sum
ertUtil: -hashfile error del comando: 0xd000225 (NT: 0xc0000225 STATUS_NOT_FOUND)
ertUtil: No se ha encontrado el objeto.

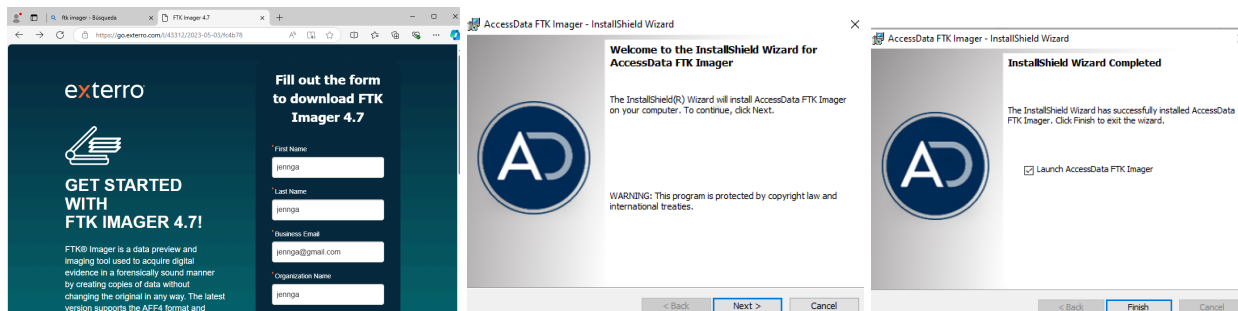
:\>certutil -hashfile copramosf.mem sha512
HA512 hash de copramosf.mem:
55e9d7ca419bb2b102aba8f1cb5656c5b4d18e38be3e42cc06e21a907acb61743cd64e46e1f0c3646c476292e63d58dc5b35bfc922311467224287
485bda6
ertUtil: -hashfile comando completado correctamente.

:\>certutil -hashfile copraosf2.mem sha512
HA512 hash de copraosf2.mem:
f5618e4d45985c4e153a809644d9ab7c3a7c1436eb8d6cd085e3bf4530a0b14ae9d06c4d86e0cc8aa0d62a3c8f925e53138e046407e12ca8bcaca
bb16bbf
  
```

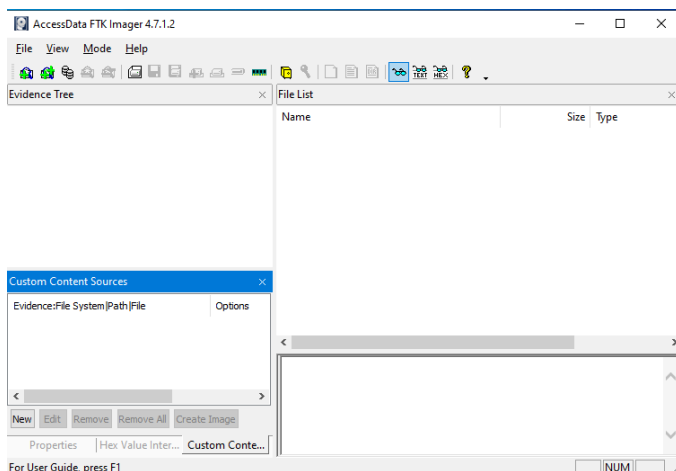
## FTK imager

Es una herramienta forense digital gratuita desarrollada por AccessData

1. Descargar e instalar la herramienta. Se tiene que descargar de su página oficial

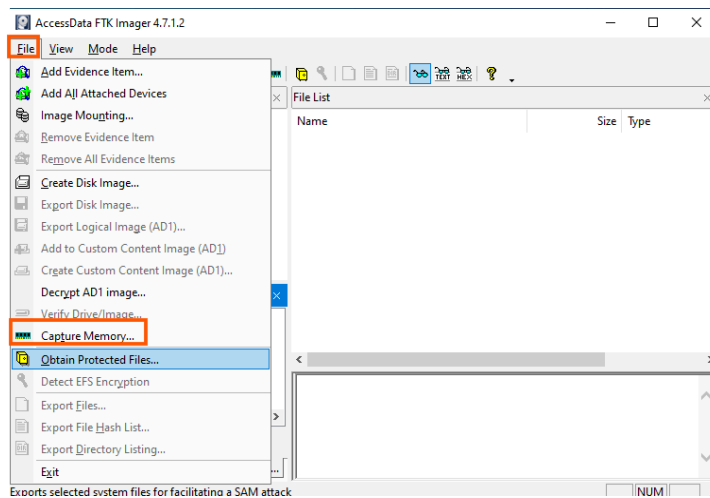


2. Se abre la aplicación

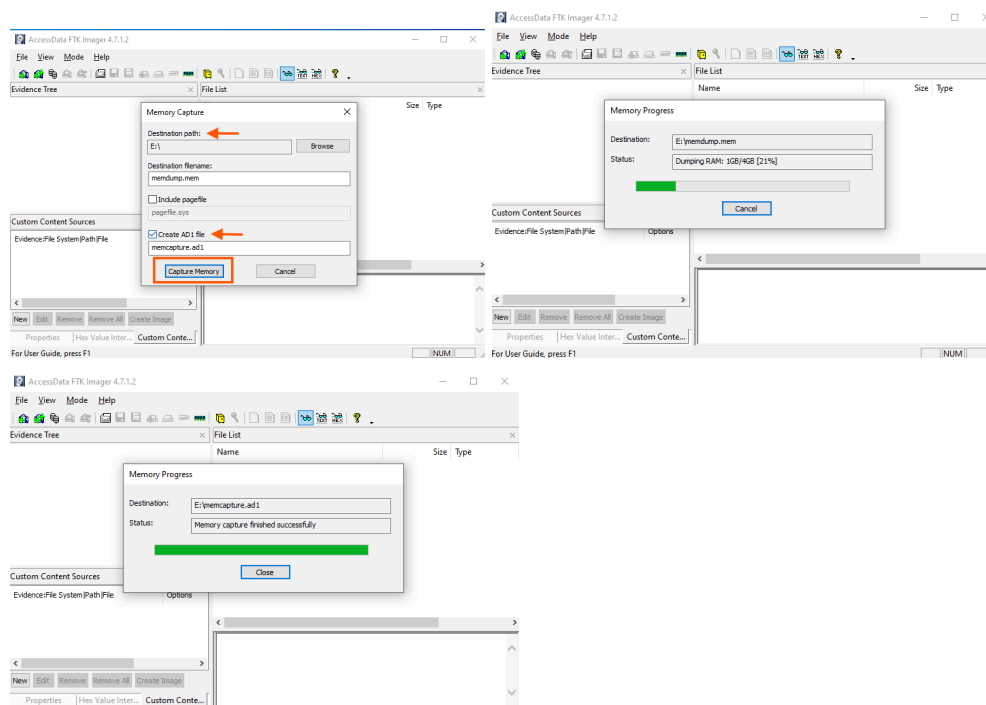


### 3. Capturamos la Memoria RAM.

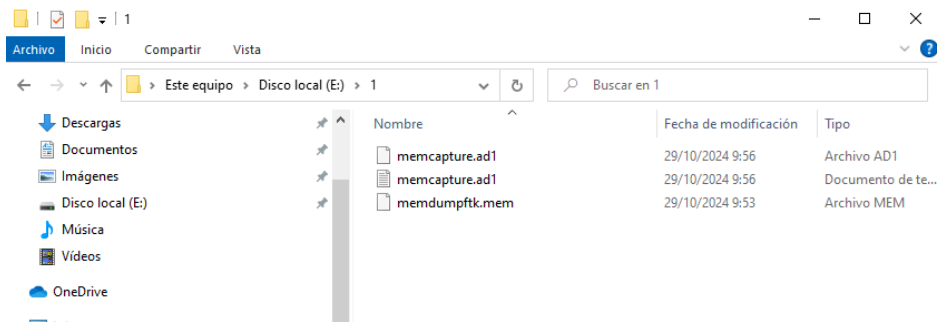
#### a. Seleccionar File/Capture Memory



#### b. Seleccionar el destino en mi caso lo pondré en el disco duro que agregue a la máquina virtual, y seleccionar los distintos formatos que se ofrecen, y seleccionar Capture Memory



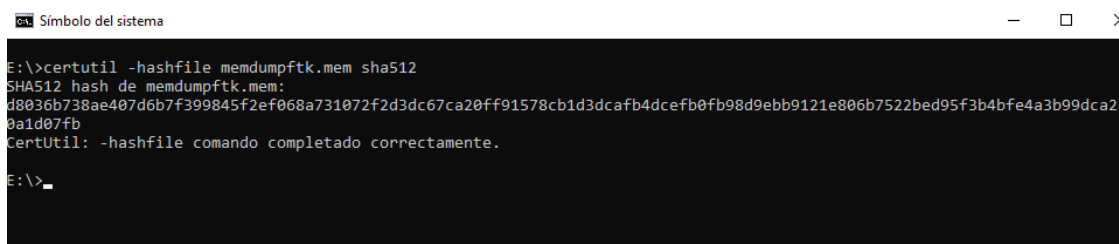
- c. Se crean tres archivos cada uno se puede analizar más tarde con la misma aplicación



### Calcular el hash de las evidencias

En window utilice el siguiente comando, mencionó que el hash es distinto ya que la memoria RAM es volátil y cambia constantemente mientras el sistema está en funcionamiento.

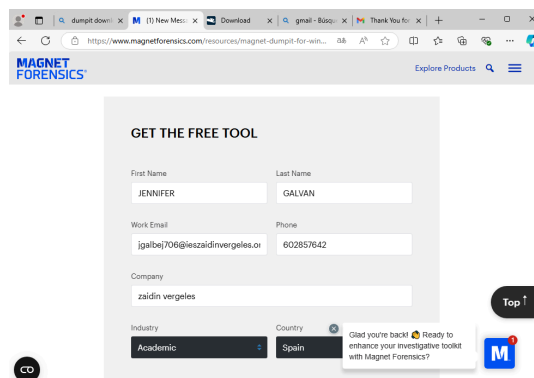
certutil -hashfile



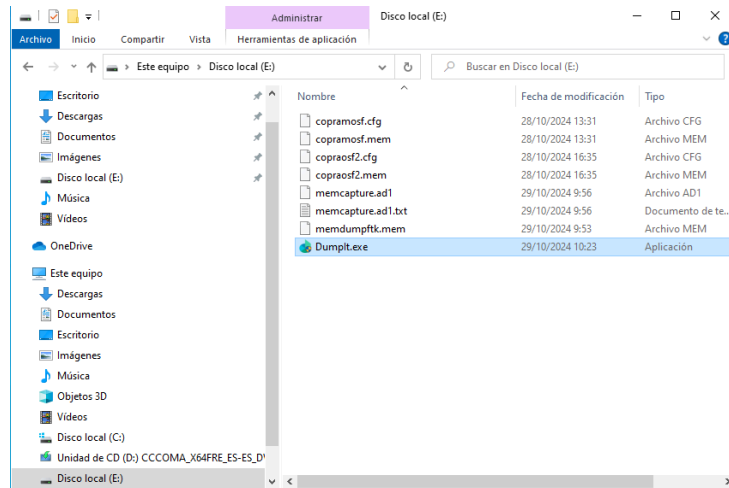
### Dumplt

Es una herramienta forense que nos permite extraer el contenido completo de la memoria RAM

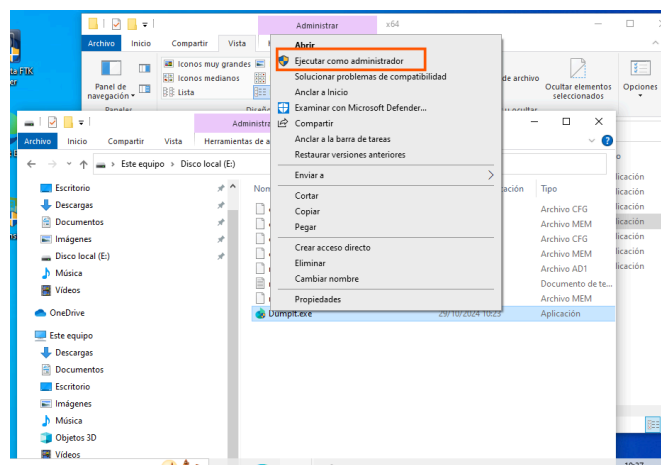
1. Descargar la herramienta, elegimos la versión gratuita, al ser una herramienta portal no requiere instalación.



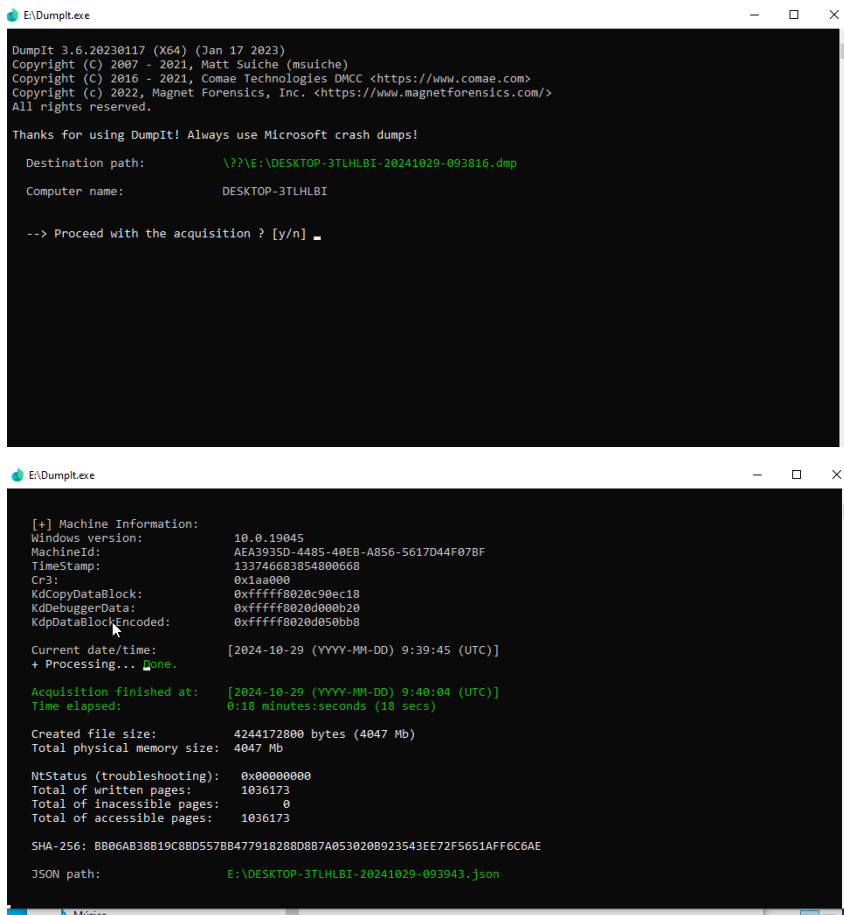
2. Ejecutamos la herramienta y automáticamente se realiza el volcado de memoria al descargar la herramienta se descarga una carpeta comprimida la descomprimos y ejecutamos como se captura en el instante debe ejecutarse en el lugar donde queremos la evidencia así que copie el archivo dumplt.exe y lo pegue en el disco duro que agregue



### 3. Capturar la memoria RAM, lo ejecutamos como administrador



### 4. Se abre una ventana y le damos yes



```
DumpIt 3.6.20230117 (X64) (Jan 17 2023)
Copyright (c) 2007 - 2021, Matt Suiche (msuiche)
Copyright (c) 2016 - 2021, Comae Technologies DMCC <https://www.comae.com>
Copyright (c) 2022, Magnet Forensics, Inc. <https://www.magnetforensics.com/>
All rights reserved.

Thanks for using DumpIt! Always use Microsoft crash dumps!

Destination path:      \\??E:\DESKTOP-3TLHLBI-20241029-093816.dmp
Computer name:         DESKTOP-3TLHLBI

--> Proceed with the acquisition ? [y/n] _

[+] Machine Information:
Windows version:      10.0.19045
MachineId:            AFA3935D-4485-40E8-A856-5617D44F07BF
TimeStamp:            133746683854800668
Cr3:                  0x1aa000
KdCopyDataBlock:      0xffffffff8020c90ec18
KdDebuggerData:        0xffffffff8020d000b20
KdpDataBlockEncoded:  0xffffffff8020d050bb8

Current date/time:     [2024-10-29 (YYYY-MM-DD) 9:39:45 (UTC)]
+ Processing... Done.

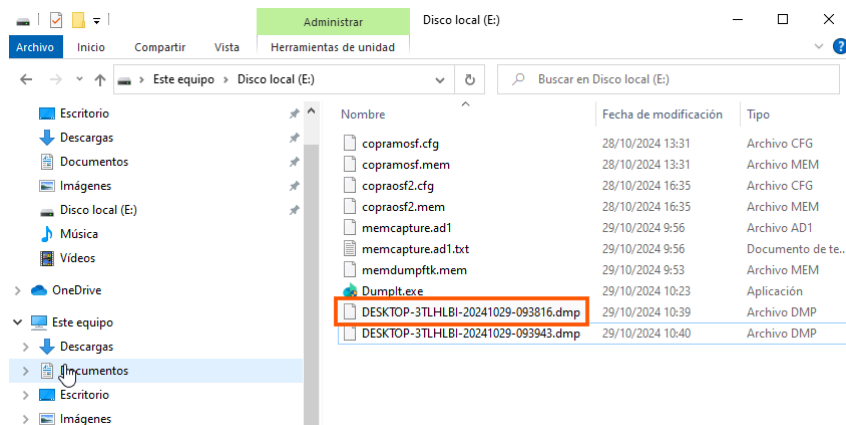
Acquisition finished at: [2024-10-29 (YYYY-MM-DD) 9:40:04 (UTC)]
Time elapsed:          0:18 minutes:seconds (18 secs)

Created file size:     4244172800 bytes (4047 Mb)
Total physical memory size: 4047 Mb

NtStatus (troubleshooting): 0x00000000
Total of written pages: 1036173
Total of inaccessible pages: 0
Total of accessible pages: 1036173

SHA-256: BB06AB38B19C8BD557BB47791828D8B7A053020B923543EE72F5651AFF6C6AE
JSON path:            E:\DESKTOP-3TLHLBI-20241029-093943.json
```

## 5. Se crean un archivo con extensión .dmp



## Calcular el hash de las evidencias

En window utilice el siguiente comando, mencionó que el hash es distinto ya que la memoria RAM es volátil y cambia constantemente mientras el sistema está en funcionamiento.

certutil -hasfile

```

C:\> Símbolo del sistema

E:\>certutil -hashfile memdumpftk.mem sha512
SHA512 hash de memdumpftk.mem:
d8036b738ae407d6b7f399845f2ef068a731072f2d3dc67ca20ff91578cb1d3dcafb4dcefb0fb98d9ebb9121e806b7522bed95f3b4bfe4a3b99dca2c9a1d07fb
CertUtil: -hashfile comando completado correctamente.

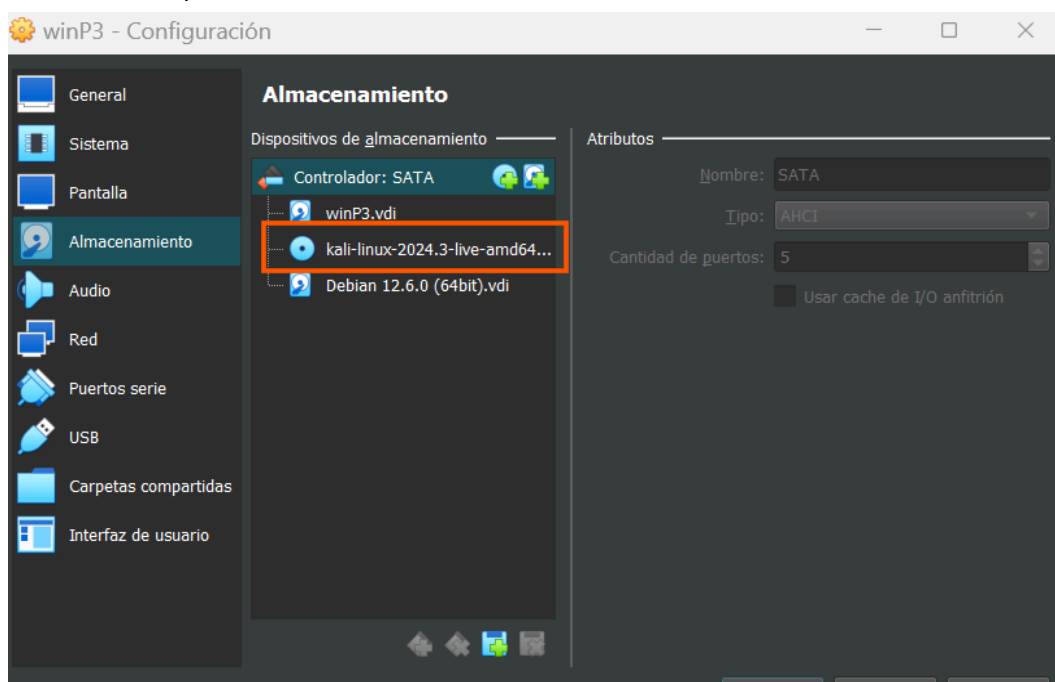
E:\>certutil -hashfile DESKTOP-3TLHLBI-20241029-093816.dmp sha512
SHA512 hash de DESKTOP-3TLHLBI-20241029-093816.dmp:
69547873a74d357cd0103af50ce60870883f2d4b3c470556c9910dc35582652e54d16fa655acb47d2a3debdeb272b55facac27a4cb5d50751abe73d5844141c6
CertUtil: -hashfile comando completado correctamente.

E:\>

```

## Calcular hash usando sha512sum

Utilice un disco de arranque de kali live en virtual box



Escribir `sudo fdisk -l` para ver cual disco vamos a utilizar, crear una carpeta llamada `mnt/window` y montamos con

`sudo mount -t ntfs-3g /dev/sdb1 /mnt/windows`

```

(kali㉿kali)-[~]
$ sudo mount -t ntfs-3g /dev/sdb1 /mnt/windows

(kali㉿kali)-[~]
$ cd /mnt/windows

(kali㉿kali)-[/mnt/windows]
$ ls
$RECYCLE.BIN  copraosf2.mem  DumpStack.log.tmp  pagefile.sys
copramosf.cfg  DESKTOP-3TLHLBI-20241029-093816.dmp  memcapture.ad1  'System Volume Information'
copramosf.mem  DESKTOP-3TLHLBI-20241029-093943.dmp  memcapture.ad1.txt
copraosf2.cfg  DumpIt.exe     memdumpftk.mem

```

Calcular el hash de cada una de las imágenes que realice en los apartados anteriores

```

(kali㉿kali)-[/mnt/windows]
$ sha512sum copramosf.mem
055e9d7ca419bb2b102aba8f1cb5656c5b4d18e38be3e42cc06e21a907acb61743cd64e46e1f0c3646c476292e63d58dc5b35bfc922311
467224287c485bda6  copramosf.mem

(kali㉿kali)-[/mnt/windows]
$ sha512sum memdumpftk.mem
d8036b738ae407d6b7f399845f2ef068a731072f2d3dc67ca20ff91578cb1d3dcafb4dcefb0fb98d9ebb9121e806b7522bed95f3b4bfe4a
3b99dca2c0a1d07fb  memdumpftk.mem

(kali㉿kali)-[/mnt/windows]
$ sha512sum DESKTOP-3TLHLBI-20241029-093816.dmp
69547873a74d357cd0103af50ce60870883f2d4b3c470556c9910dc35582652e54d16fa655acb47d2a3debdeb272b55facac27a4cb5d507
51abe73d5344141c6  DESKTOP-3TLHLBI-20241029-093816.dmp

(kali㉿kali)-[/mnt/windows]
$

```

Es una buena práctica sacar el hash en el sistema operativo donde se encuentra ya que si se cambia puede alterar un poco la evidencia.

[índice](#)

## Extracción de evidencias digitales volátiles (RAM) Linux

### Microsoft AVML (linux)

Es una herramienta escrita en rust sirve para obtener el contenido de la memoria volátil de un sistema Linux; una de las ventajas de esta herramienta, es que no necesita conocer de entrada la distribución basada en Linux o kernel que se está utilizando.

1. Descargue el fichero de los documentos compartidos por drive y lo descomprimi

**tar -xzvf avml.tgz**

```

jenny@debian:~$ tar -xzvf avml.tgz
x86_64-unknown-linux-gnu/
x86_64-unknown-linux-gnu/CACHEDIR.TAG
x86_64-unknown-linux-gnu/release/
x86_64-unknown-linux-gnu/release/examples/

```

2. Revistas los discos que tenemos (**sudo fdisk -l**)

```

jenny@debian:~$ sudo fdisk -l
[sudo] contraseña para jenny:
Disco /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectores
Modelo de disco: VBOX HARDDISK
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x17df26d8

Disposit.  Inicio Comienzo      Final Sectores Tamaño Id Tipo
/dev/sda1  *          2048 39942143 39940096    19G 83 Linux
/dev/sda2          39944190 41940991  1996802    975M  5 Extendida
/dev/sda5          39944192 41940991  1996800    975M 82 Linux swap / Solaris

Disco /dev/sdb: 500 GiB, 536870912000 bytes, 1048576000 sectores
Modelo de disco: VBOX HARDDISK
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x3c21e163

Disposit.  Inicio Comienzo      Final Sectores Tamaño Id Tipo
/dev/sdb1  *          2048 1048573951 1048571904    500G  7 HPFS/NTFS/exFAT
jenny@debian:~$

```



3. Creo una carpeta donde se montará el disco y monto el disco en el nuevo punto de montaje

**sudo mkdir /mnt/evidencia**

```
jenny@debian:~$ sudo mkdir /mnt/evidencia
jenny@debian:~$ sudo mount /dev/sdb1 /mnt/evidencia
mount: /mnt/evidencia: tipo de sistema de ficheros 'ntfs' desconocido.
dmesg(1) may have more information after failed mount system call.
jenny@debian:~$ sudo apt install ntfs-3g
```

**sudo mount -t ntfs-3g /dev/sdb1 /mnt/evidencia**

```
jenny@debian:~$ sudo mount -t ntfs-3g /dev/sdb1 /mnt/evidencia
jenny@debian:~$
```

4. Ejecuto la herramienta AVML **./avml /mnt/evidencia/imagram.lime**

```
jenny@debian:~/x86_64-unknown-linux-gnu/release$ sudo ./avml /mnt/evidencia/imagram.lime
```

5. compruebo que sí se creó la imagen

```
jenny@debian:~/x86_64-unknown-linux-gnu/release$ sudo ./avml /mnt/evidencia/imagram.lime
jenny@debian:~/x86_64-unknown-linux-gnu/release$ cd /mnt/evidencia/
jenny@debian:/mnt/evidencia$ ls
imagram.lime
jenny@debian:/mnt/evidencia$
```

6. Calcular el hash **sha512sum imaram.lime**

```
jenny@debian:/mnt/evidencia$ sudo sha512sum imagram.lime
803ad90c05c30619f277a41698b05f28ed29d646c46f3415d9e7b48abb842b4d47198ac97f629ff5fadb7db4e31e6da7c46b
9fc7b7b4415fe6bb6b8404ace5f9  imagram.lime
jenny@debian:/mnt/evidencia$
```

7. Calcular el hash con otro sistema de arranque sin encender la máquina.

```
(kali@kali)-[~]
$ sudo mkdir /mnt/evidenciakali

(kali@kali)-[~]
$ sudo mount -t ntfs-3g /dev/sda1 /mnt/evidenciakali

(kali@kali)-[~]
$ cd /mnt/evidenciakali

(kali@kali)-[/mnt/evidenciakali]
$ ls
$RECYCLE.BIN      copraosf2.mem      DumpStack.log.tmp  memdumpftk.mem
copramosf.cfg     DESKTOP-3TLHLBI-20241029-093816.dmp  imagram.lime       pagefile.sys
copramosf.mem     DESKTOP-3TLHLBI-20241029-093943.dmp  memcapture.ad1     'System Volume Information'
copraosf2.cfg     DumpIt.exe         memcapture.ad1.txt

(kali@kali)-[/mnt/evidenciakali]
$ sha512sum imagram.lime
b5b9372db4583f2360253bd02b67096d153d31a984b119db06272ee96de126d4f0a77098de78c9c2f41aeb02fd9a0032e5044fd398d677
938a4f2d4b75d8a24  imagram.lime

(kali@kali)-[/mnt/evidenciakali]
$
```

[índice](#)