

# SEGURIDAD EN SISTEMAS OPERATIVOS

4º Grado en Informática – Complementos de Ing. del Software  
Curso 2019-20

---

**Práctica [1].** Administración de la seguridad en Linux.

**Sesión [6].** Cifrado de sistemas de archivos. Esteganografía y estegoanálisis.

**Autor<sup>1</sup>:** Nazaret Román Guerrero

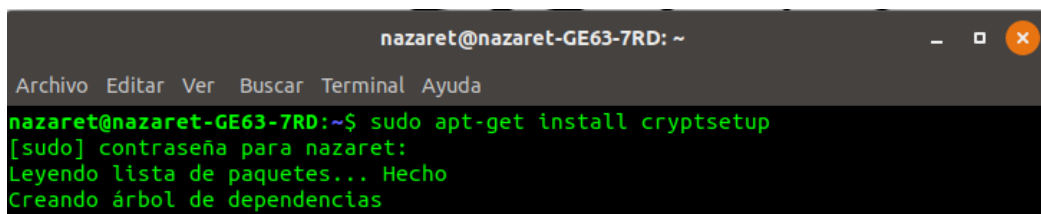
---

## Ejercicio 1.

---

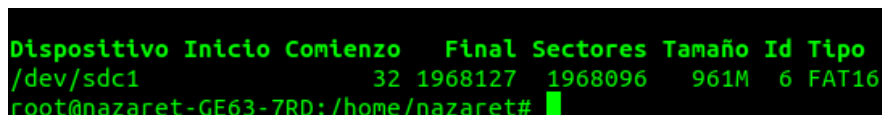
Utilizar `cryptsetup` para crear una partición encriptada en un pendrive. Escribir un archivo en él. Desmontarlo y extraerlo. ¿Qué ocurre cuando volvemos a conectarlo?

Para empezar, es necesario instalar `cryptsetup`, tal y como se muestra en la imagen:



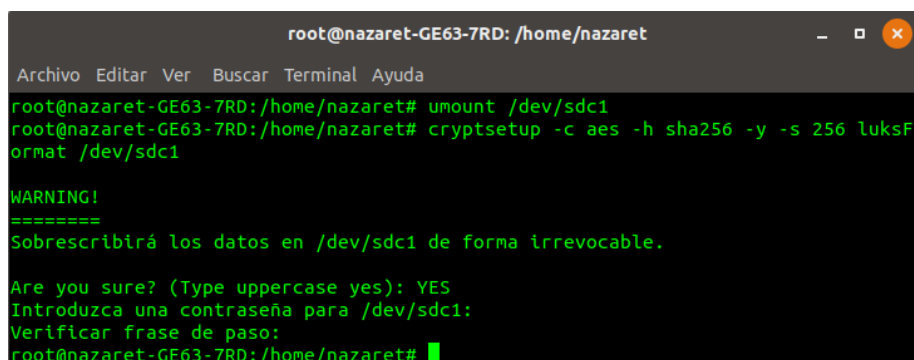
```
nazaret@nazaret-GE63-7RD: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
nazaret@nazaret-GE63-7RD:~$ sudo apt-get install cryptsetup  
[sudo] contraseña para nazaret:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias
```

Una vez instalado, vamos a comprobar la dirección que le ha asignado el sistema al pendrive. Para ello, utilizamos la orden `fdisk -l`, que nos muestra que el punto de montaje es `/dev/sdc1`.



```
Dispositivo Inicio Comienzo Final Sectores Tamaño Id Tipo  
/dev/sdc1 32 1968127 1968096 961M 6 FAT16  
root@nazaret-GE63-7RD:/home/nazaret#
```

Como el dispositivo está montado, el primer paso es desmontarlo con `umount`. Encriptamos el dispositivo con `sha256`, con un tamaño de clave de 256 bits y le damos formato con `luksFormat`:



```
root@nazaret-GE63-7RD:/home/nazaret  
Archivo Editar Ver Buscar Terminal Ayuda  
root@nazaret-GE63-7RD:/home/nazaret# umount /dev/sdc1  
root@nazaret-GE63-7RD:/home/nazaret# cryptsetup -c aes -h sha256 -y -s 256 luksFormat /dev/sdc1  
  
WARNING!  
=====  
Sobrescribirá los datos en /dev/sdc1 de forma irrevocable.  
  
Are you sure? (Type uppercase yes): YES  
Introduzca una contraseña para /dev/sdc1:  
Verificar frase de paso:  
root@nazaret-GE63-7RD:/home/nazaret#
```

---

1 Como autor declaro que los contenidos del presente documento son originales y elaborados por mi. De no cumplir con este compromiso, soy consciente de que, de acuerdo con la “Normativa de evaluación y de calificaciones de los estudiantes de la Universidad de Granada” esto “conllevará la calificación numérica de cero ... independientemente del resto de calificaciones que el estudiante hubiera obtenido ...”

En el proceso del paso anterior, se nos pide una contraseña, que será la que se pedirá cada vez que se inserte el pendrive y se monte el sistema de archivos de éste.

Tras esto, vamos a mapear el dispositivo en memoria para continuar. Primero abrimos el dispositivo para que se monte y después lo mapeamos. Si fuera un dispositivo fijo (no un disco extraíble como es el pendrive) se tendría que modificar la tabla de sistemas de archivos del sistema (fstab). Como no es el caso, simplemente montamos el dispositivo con el mapeo:

```
root@nazaret-GE63-7RD: /home/nazaret
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaret-GE63-7RD:/home/nazaret# cryptsetup luksOpen /dev/sdc1 test
Introduzca una contraseña para /dev/sdc1:
root@nazaret-GE63-7RD:/home/nazaret# mkfs /dev/mapper/test
mke2fs 1.44.1 (24-Mar-2018)
Se está creando un sistema de ficheros con 245500 bloques de 4k y 61440 nodos-i
UUID del sistema de ficheros: 6977d95c-ebc8-4f87-a2bc-88284f43a8c0
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Escribiendo superbloques y la información contable del sistema de archivos: hecho
root@nazaret-GE63-7RD:/home/nazaret#
```

Ahora montamos el dispositivo tras mapearlo con la orden `mount /dev/mapper/test /mnt/prueba`. El directorio `/mnt/prueba` debe crearse antes y darle permisos de lectura, escritura y ejecución tanto al propietario como al grupo y a otros, tal y como se muestra aquí:

```
root@nazaret-GE63-7RD: /home/nazaret
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaret-GE63-7RD:/home/nazaret# mkdir /mnt/prueba
root@nazaret-GE63-7RD:/home/nazaret# chmod 777 /mnt/prueba
root@nazaret-GE63-7RD:/home/nazaret# ls -l /mnt/
total 4
drwxrwxrwx 2 root root 4096 nov  8 20:36 prueba
root@nazaret-GE63-7RD:/home/nazaret#
```

Para comprobar que el dispositivo se ha montado correctamente, lo comprobamos de dos formas distintas: con `fdisk -l` y con `df -h` (en la segunda orden se saca la primera y la última línea de la salida del comando para que se vea con más claridad, puesto que si no mostraría todos los sistemas de archivos que hay montados que en mi caso son muchos).

La orden `fdisk -l` nos muestra que el dispositivo está mapeado en `/dev/mapper/test`, nos muestra los sectores del disco y su tamaño:

```
Disco /dev/mapper/test: 959 MiB, 1005568000 bytes, 1964000 sectores
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
root@nazaret-GE63-7RD:/home/nazaret#
```

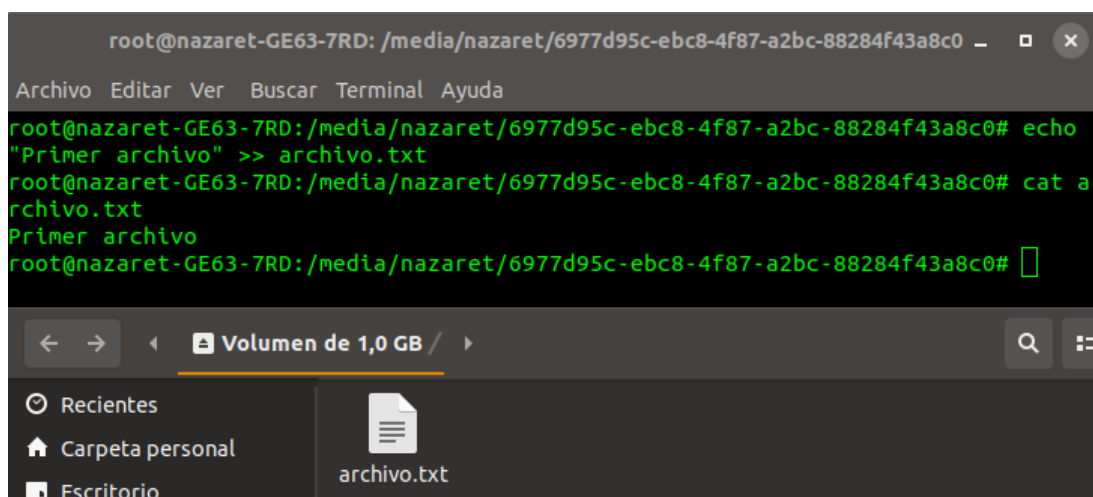
El comando `df -h` nos muestra el sistema de archivos, el tamaño, el porcentaje usado y el punto de montaje entre otras cosas:

```
root@nazaret-GE63-7RD: /home/nazaret
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaret-GE63-7RD:/home/nazaret# df -h | head -1; df -h | tail -1
S.ficheros      Tamaño Usados  Disp Uso% Montado en
/dev/mapper/test 944M   1,2M  895M   1% /mnt/prueba
root@nazaret-GE63-7RD:/home/nazaret#
```

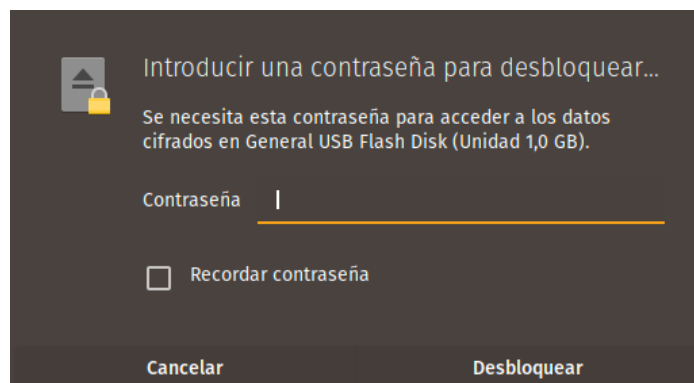
Una vez comprobado que está bien montado, cerramos el dispositivo:

```
root@nazaret-GE63-7RD: /home/nazaret
Archivo Editar Ver Buscar Terminal Ayuda
root@nazaret-GE63-7RD:/home/nazaret# cryptsetup luksClose /dev/mapper/test
El dispositivo /dev/mapper/test está todavía en uso.
root@nazaret-GE63-7RD:/home/nazaret#
```

Podemos crear archivos en él como en un dispositivo cualquiera:



Una vez creado el archivo, extraemos el dispositivo y al volverlo a introducir nos preguntará la clave:



Por lo que nuestro dispositivo está cifrado correctamente.

## Ejercicio 2.

Utilizar la herramienta Steghide para ocultar un mensaje dentro de una imagen, tal como acabamos de ver. Comparar los archivos portador antes y después de usar la técnica para ver las diferencias.

Para utilizar esta herramienta lo primero es instalarla en el sistema mediante apt-get install steghide.

Vamos a crear un archivo que irá oculto dentro de la fotografía que elijamos. El archivo que he creado es el siguiente:

```
nazaret@nazaret-GE63-7RD: ~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E... - [X]
Archivo Editar Ver Buscar Terminal Ayuda
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...
eganografia$ echo "Te voy a contar un secreto que irá dentro de la foto" >> ar
chivo.txt
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...
eganografia$ cat archivo.txt
Te voy a contar un secreto que irá dentro de la foto
```

Y la imagen en la que estará oculta pertenece a un comic japonés (un manga) que me gusta mucho:



Una vez que tenemos ambas cosas, tanto la imagen como el mensaje que vamos a ocultar, solo queda llevar a cabo el proceso de ocultación de la información en el portador. Para ello, utilizamos el programa que acabamos de instalar.

Para ello, utilizamos el comando steghide embed -cf foto.jpeg -ef archivo.txt, que indica que se va a integrar el archivo dado con la opción -ef en el archivo portador dado con la opción -cf. Nos pide una contraseña al ocultar el mensaje dentro de la imagen portadora:

```
nazaret@nazaret-GE63-7RD: ~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E... - [X]
Archivo Editar Ver Buscar Terminal Ayuda
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...
eganografia$ steghide embed -cf foto.jpeg -ef archivo.txt
Anotar salvoconducto:
Re-ingresar salvoconducto:
adjuntando "archivo.txt" en "foto.jpeg"... hecho
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...
eganografia$ █
```

La fotografía con el mensaje oculto se ve igual que la original, como podemos apreciar aquí:



Ahora vamos a comprobar la diferencia entre una y otra imagen para saber en qué se diferencian. Para ello, podemos mirar el tamaño de cada una. Como podemos observar, la imagen original (original.jpeg) es un poco más pequeña que la imagen que contiene la información oculta (foto.jpeg).

Además, utilizando el comando diff comprobamos que son diferentes puesto que la salida de ejecutar el comando es, literalmente, “los archivos binarios son distintos”.

Pero ese es todo el cambio que notamos, puesto que ambas imágenes se ven exactamente iguales para nuestro ojo si las abrimos con un visor de imágenes.

```
nazaret@nazaret-GE63-7RD: ~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...  
Archivo Editar Ver Buscar Terminal Ayuda  
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...  
eganografia$ ls -l  
total 676  
-rw-r--r-- 1 nazaret nazaret 54 nov 8 21:45 archivo.txt  
-rw-rw-r-- 1 nazaret nazaret 353832 nov 8 21:53 foto.jpeg  
-rw-rw-r-- 1 nazaret nazaret 328598 nov 8 21:57 original.jpeg  
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...  
eganografia$ diff original.jpeg foto.jpeg  
Los archivos binarios original.jpeg y foto.jpeg son distintos  
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...  
eganografia$
```

Podemos extraer el mensaje oculto mediante la orden extract. Primero eliminamos el mensaje original, ya que se extrae con el mismo nombre. Tras esto, sacamos el mensaje y lo mostramos:

```
nazaret@nazaret-GE63-7RD: ~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...  
Archivo Editar Ver Buscar Terminal Ayuda  
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...  
eganografia$ rm archivo.txt  
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...  
eganografia$ steghide extract -sf foto.jpeg  
Anotar salvoconducto:  
anote los datos extraidos e/"archivo.txt".  
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...  
eganografia$ cat archivo.txt  
Te voy a contar un secreto que irá dentro de la foto  
nazaret@nazaret-GE63-7RD:~/Escritorio/ETSIIT_comp/4º/Cuatri 1/SSO/Practicas/P1/E...  
eganografia$
```



Como podemos comprobar, el mensaje es el mismo que habíamos ocultado.

### Ejercicio 3.

Utilizar VSL para analizar la imagen esteganográfica generada en el ejercicio anterior para detectar información oculta.

Para utilizar esta herramienta es necesario utilizar un entorno de ventanas, así que voy a utilizar Windows puesto que en mi host de Ubuntu no funciona.

Primero comprobamos que tenemos java instalado para poder ejecutar el programa. Para ello, accedemos a la terminal de Windows y comprobamos la versión:

```
C:\Windows\system32\cmd.exe

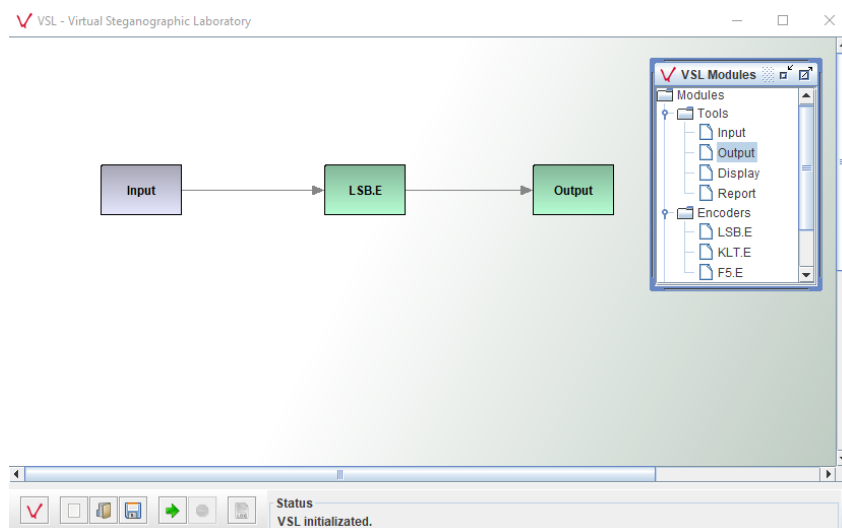
Microsoft Windows [Versión 10.0.17134.885]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\nazar>java -version
java version "1.8.0_231"
Java(TM) SE Runtime Environment (build 1.8.0_231-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.231-b11, mixed mode)

C:\Users\nazar>
```

Una vez que sabemos que tenemos java, descargamos la aplicación y la ejecutamos. Lo primero es crear un modelo de lo que vamos a hacer, uno como el que se muestra en la imagen de abajo. Para ello, seleccionamos el input y el output de la pestaña tools y el codificador del bit menos significativo de la pestaña encoders.

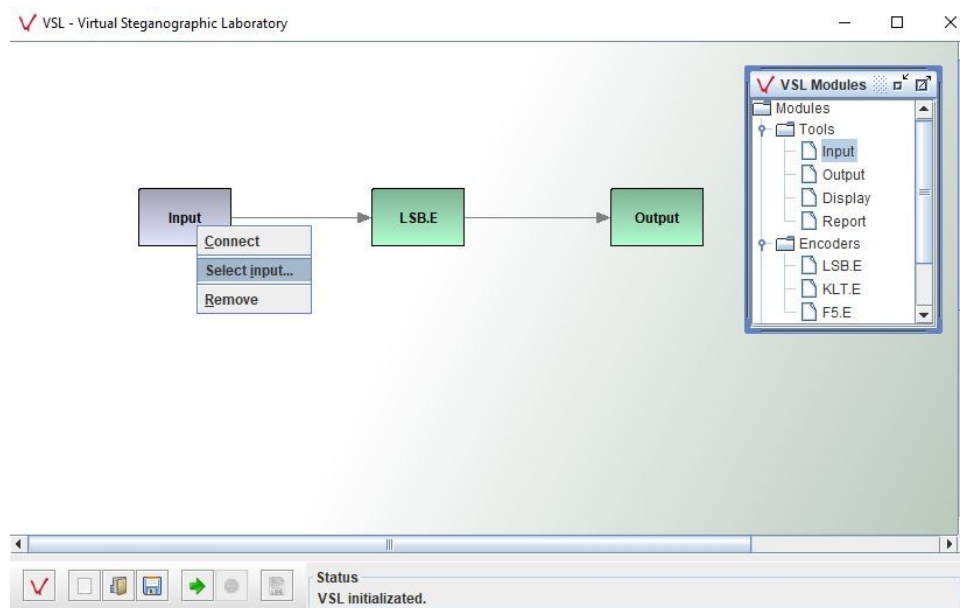
Una vez que los tenemos, pulsamos click derecho sobre cada uno y los conectamos entre sí tal y como se ve en la imagen.



Una vez hecho esto, seleccionamos el archivo de entrada. La foto en este ejercicio será distinta a la del anterior. La fotografía es la siguiente:



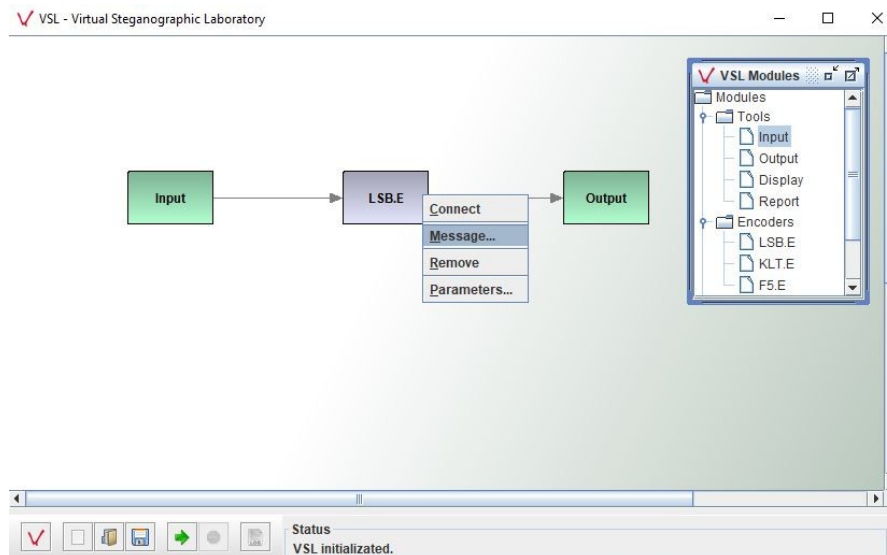
Añadimos dicha imagen al input del modelo tal y como se puede ver:



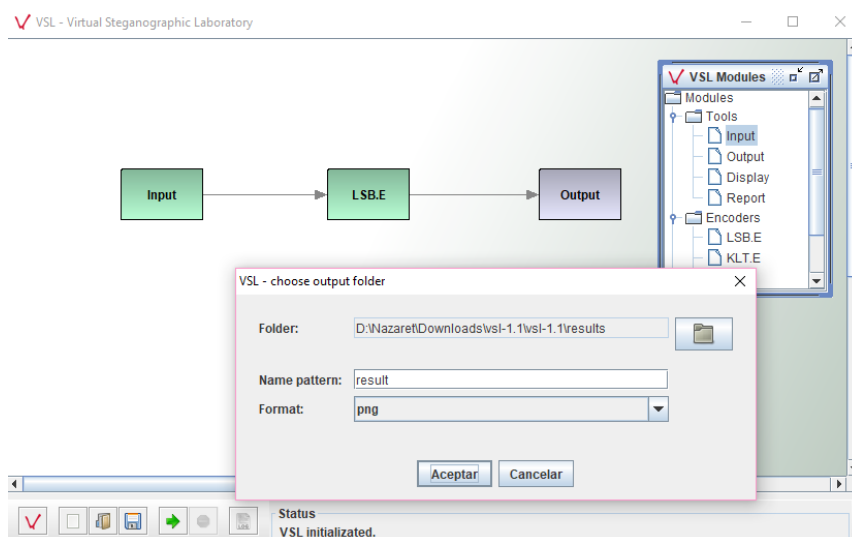
Ahora introducimos el mensaje en el item LSB. El mensaje es este:

mensaje: Bloc de notas  
 Archivo Edición Formato Ver Ayuda  
 Este achivo va a ser ocultado por VSL

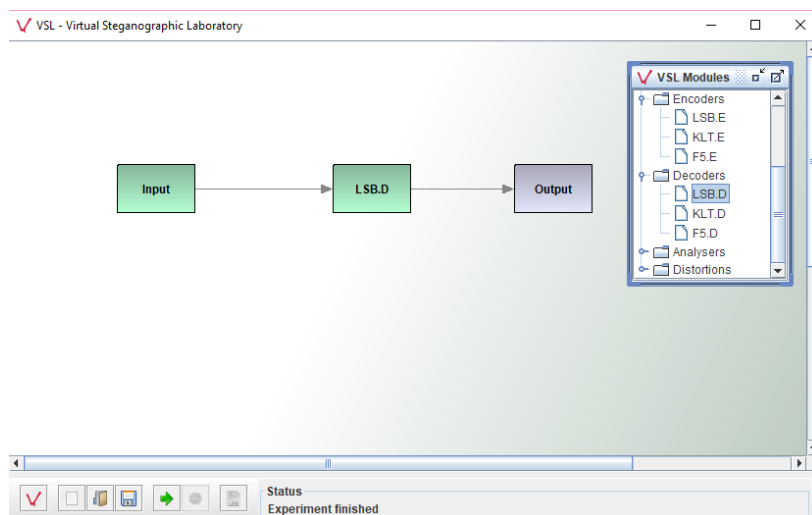
Añadimos dicho mensaje como se observa aquí:



Por último, comprobamos la salida. Se creará un archivo por defecto en la ruta que se muestra en las propiedades del output:



El archivo ya ha sido introducido en la imagen. Ahora vamos a comprobar que dicho archivo de verdad está ahí haciendo el proceso inverso. Primero, eliminamos el codificador para sustituirlo por un decodificador (pestaña decoders) como se puede ver en la siguiente imagen:



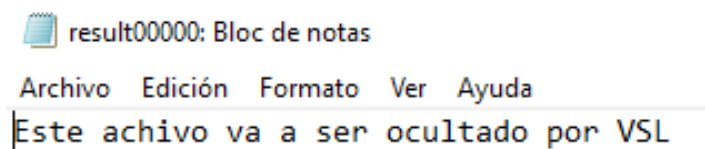


Una vez que lo tenemos, cambiamos el archivo del item input. El archivo que seleccionamos es el que se ha generado en el proceso anterior.

Tras hacer esto, ejecutamos el decodificador. Se generará un archivo como el que podemos ver:

 result00000	09/11/2019 11:02	Archivo	1 KB
---	------------------	---------	------

Una vez que tenemos dicho archivo, lo abrimos para ver qué hay dentro. Lo abrimos con el bloc de notas. El archivo contiene lo siguiente:



Por lo tanto, el archivo estaba correctamente oculto dentro de la imagen.