

IES Valle Inclán



Encriptación asimétrica con Kleopatra y GPG

Ethan Erwin Sánchez
Víctor Rodríguez Pérez

Índice

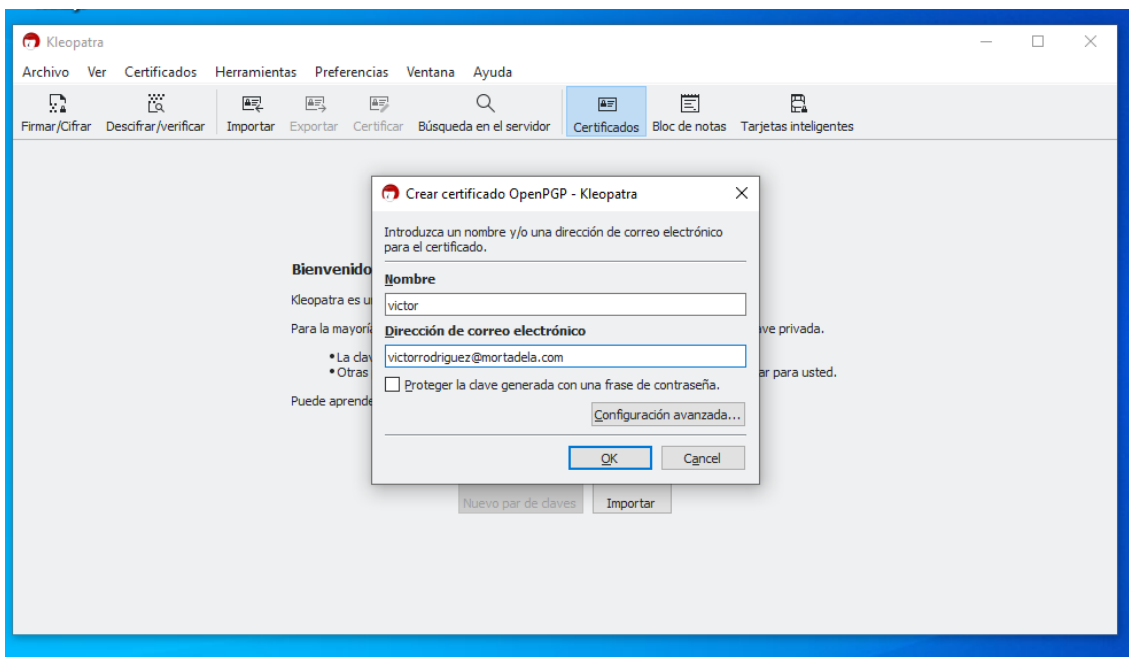
Índice	2
Creación de claves	3
Creación de clave usando Kleopatra	3
Creación de clave usando GPG	4
Exportación de claves	5
Importación de claves	6
Encriptar mensajes	8
Encriptando con Kleopatra	8
Encriptando con GPG	9
Desencriptar mensajes	10
Desencriptando con Kleopatra	10
Desencriptando con GPG	11
Conclusiones	12

Creación de claves

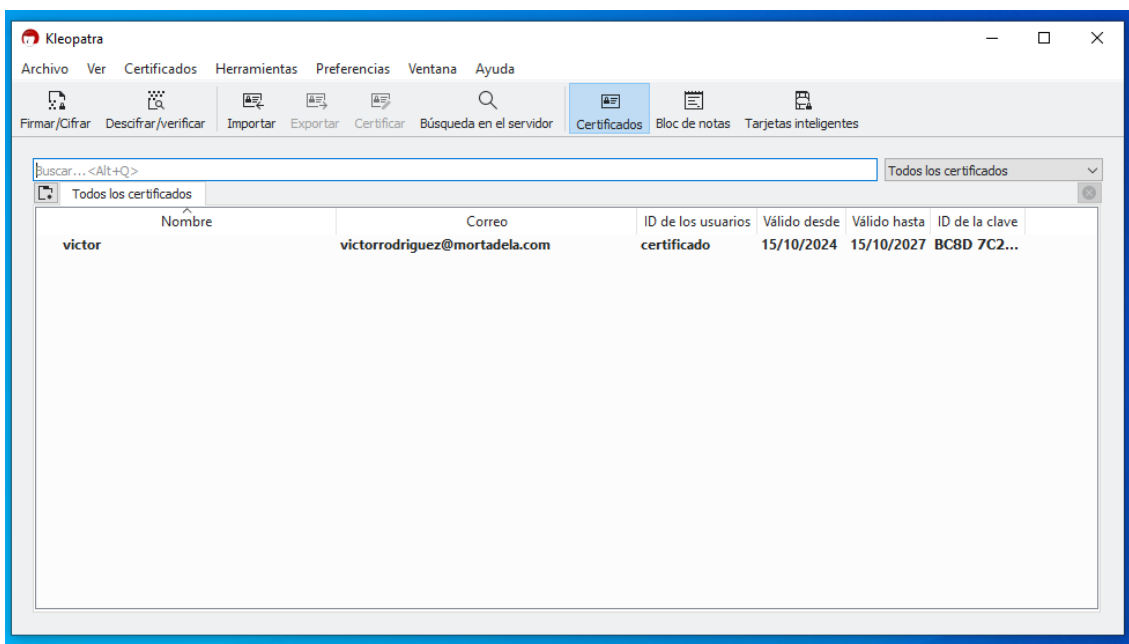
Para realizar la práctica hemos usado Kleopatra en Windows y GPG en Linux, Víctor creará en Windows y Ethan en Linux

Creación de clave usando Kleopatra

Para crear las claves en la aplicación de Kleopatra seleccionamos la opción de “Nuevo par de claves” e introduciremos la información correspondiente



Podremos comprobar que nuestra clave ahora la tenemos enlistada y lista para usar



Creación de clave usando GPG

Para crear las claves en Linux usaremos la utilidad GPG con el siguiente comando:

```
gpg --gen-key
```

Introduciremos el nombre y email para identificar la llave.

```
$gpg --gen-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.
GnuPG needs to construct a user ID to identify your key.

Real name: Ethan
Email address: ethane66@gmail.com
You selected this USER-ID:
    "Ethan <ethane66@gmail.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /home/user/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/home/user/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/user/.gnupg/openpgp-revocs.d/7E1CBBE4215D735013F4873EFC1A9C8590762E8D.rev'
public and secret key created and signed.

pub   rsa3072 2024-10-14 [SC] [expires: 2026-10-14]
       7E1CBBE4215D735013F4873EFC1A9C8590762E8D
uid           Ethan <ethane66@gmail.com>
sub   rsa3072 2024-10-14 [E] [expires: 2026-10-14]
```

Podemos comprobar que tenemos nuestra llave creada con el comando

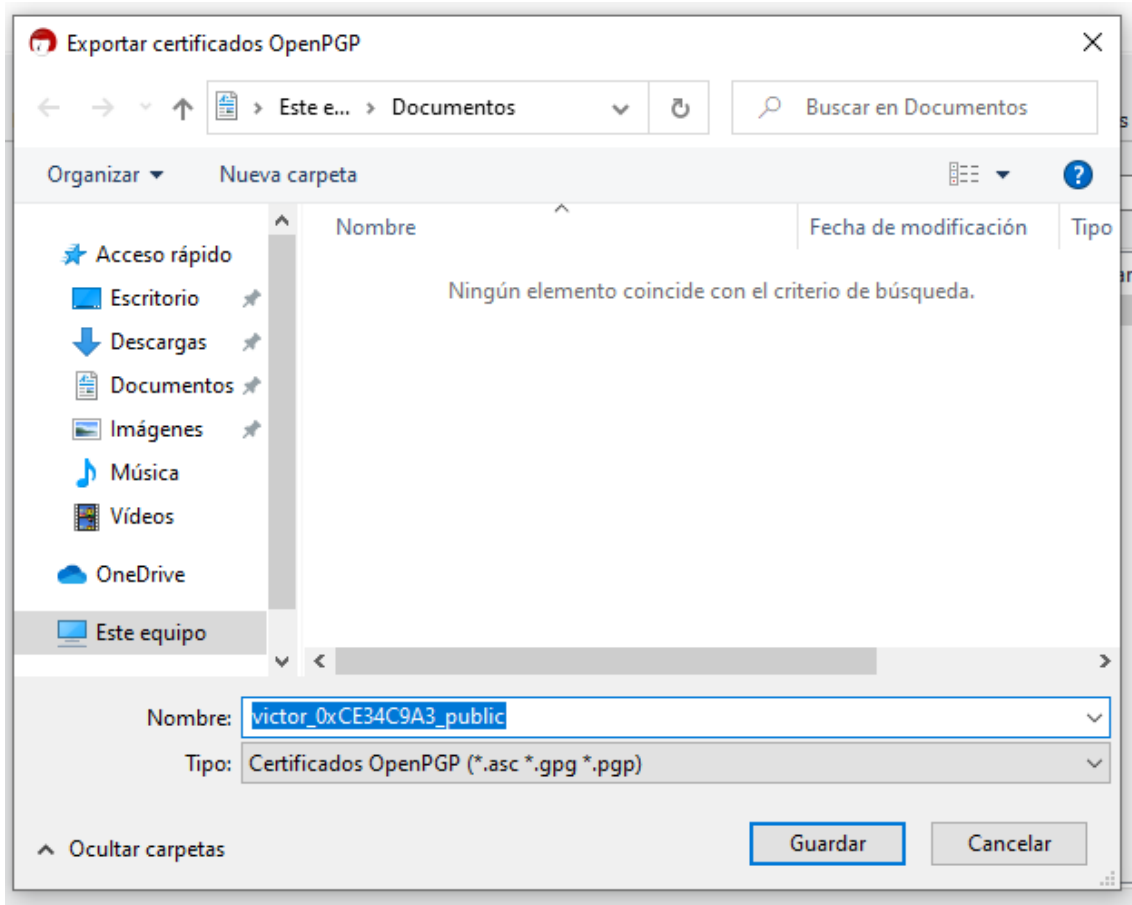
```
gpg --list-keys
```

```
$gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2026-10-14
/home/user/.gnupg/pubring.kbx
-----
pub   rsa3072 2024-10-14 [SC] [expires: 2026-10-14]
       7E1CBBE4215D735013F4873EFC1A9C8590762E8D
uid           [ultimate] Ethan <ethane66@gmail.com>
sub   rsa3072 2024-10-14 [E] [expires: 2026-10-14]

[user@parrot]~[~/Documents]
```

Exportación de claves

Una vez creadas las claves, en Windows usando Kleopatra seleccionaremos “exportar” o CTRL + E y guardaremos el certificado



En Linux usaremos GPG con el comando:

```
gpg --armor --export ethane66@gmail.com > clavepublicaethan
```

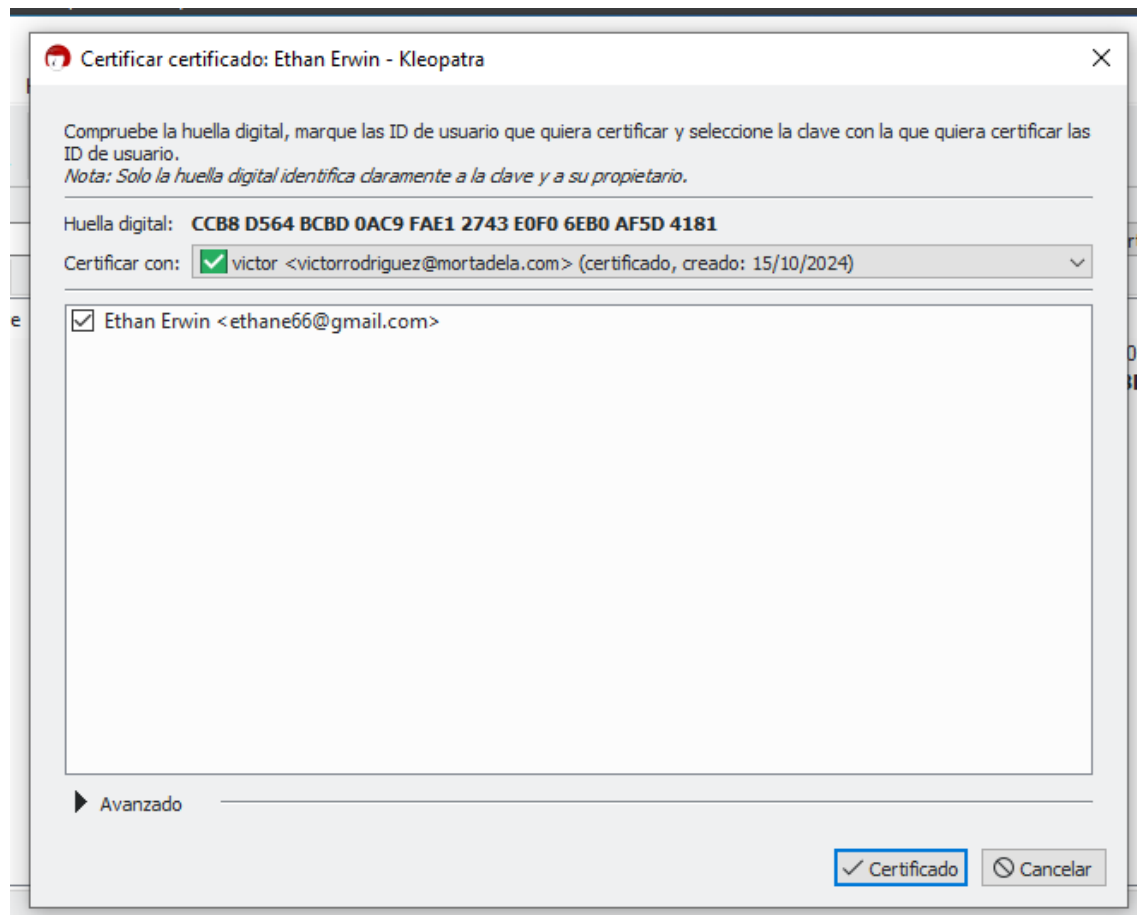
```
[user@parrot]-[~]  
$gpg --armor --export ethane66@gmail.com > clavepublicaethan
```

Nos compartiremos las claves para poder enviarnos archivos encriptados

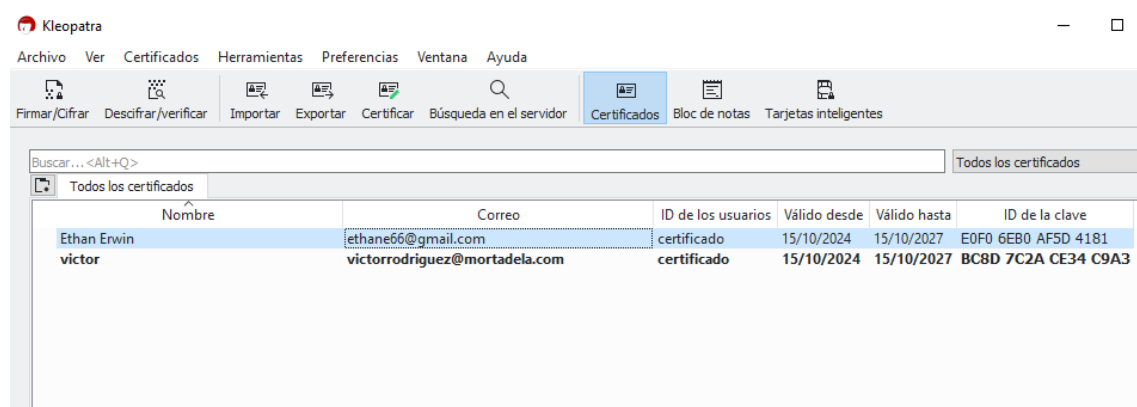
Importación de claves

En Windows, en la interfaz de Kleopatra seleccionaremos la opción de “Importar” y seleccionaremos el certificado de Ethan

Para poder utilizarlo antes tendremos que certificar el certificado con nuestra llave, si no lo marcará como desconocido y no tendremos opción a usarlo.



Podemos comprobar que ahora tenemos el certificado de Víctor y de Ethan para poder cifrar y descifrar



Para importar claves en Linux usaremos el comando

```
gpg --import victor_xxx_public.asc
```

Antes tendremos que realizar el proceso similar como en Windows que hay que certificar la clave para poder utilizarla

```
gpg --sign-key victorrodriguez@mortadela.com
```

```
ethan@ubuntu-redes:~/Documentos$ gpg --sign-key victorrodriguez@mortadela.com

pub  ed25519/BC8D7C2ACE34C9A3
     creado: 2024-10-15  caduca: 2027-10-15  uso: SC
     confianza: desconocido  validez: desconocido
sub  cv25519/0B2E4AF8A8F48724
     creado: 2024-10-15  caduca: 2027-10-15  uso: E
[desconocida] (1). victor <victorrodriguez@mortadela.com>

pub  ed25519/BC8D7C2ACE34C9A3
     creado: 2024-10-15  caduca: 2027-10-15  uso: SC
     confianza: desconocido  validez: desconocido
Huella clave primaria: 7904 0CF7 384E FAC6 29DB  A541 BC8D 7C2A CE34 C9A3

     victor <victorrodriguez@mortadela.com>

Esta clave expirará el 2027-10-15.
¿Está realmente seguro de querer firmar esta clave
con su clave: "Ethan Erwin <ethane66@gmail.com>" (E0F06EB0AF5D4181)?

¿Firmar de verdad? (s/N) s
```

Podemos comprobar las claves que tenemos con

```
gpg --list-keys
```

```
ethan@ubuntu-redes:~/Documentos$ gpg --list-keys
gpg: comprobando base de datos de confianza
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: nivel: 0  validez: 1  firmada: 1  confianza: 0-, 0q, 0n, 0m, 0f, 1u
gpg: nivel: 1  validez: 1  firmada: 0  confianza: 1-, 0q, 0n, 0m, 0f, 0u
gpg: siguiente comprobación de base de datos de confianza el: 2027-10-15
/home/ethan/.gnupg/pubring.kbx
-----
pub  ed25519 2024-10-15 [SC] [caduca: 2027-10-15]
     CCB8D564BCBD0AC9FAE12743E0F06EB0AF5D4181
uid  [ absoluta ] Ethan Erwin <ethane66@gmail.com>
sub  cv25519 2024-10-15 [E] [caduca: 2027-10-15]

pub  ed25519 2024-10-15 [SC] [caduca: 2027-10-15]
     79040CF7384EFAC629DBA541BC8D7C2ACE34C9A3
uid  [ total ] victor <victorrodriguez@mortadela.com>
sub  cv25519 2024-10-15 [E] [caduca: 2027-10-15]

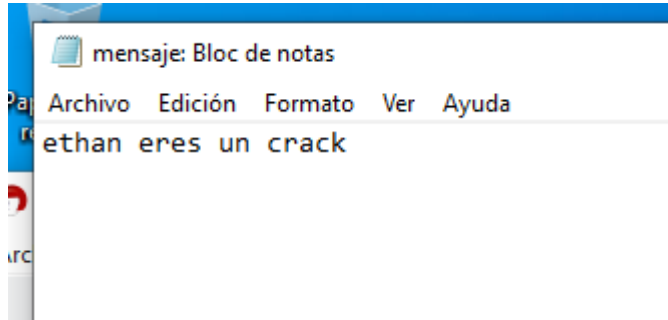
ethan@ubuntu-redes:~/Documentos$
```

Encriptar mensajes

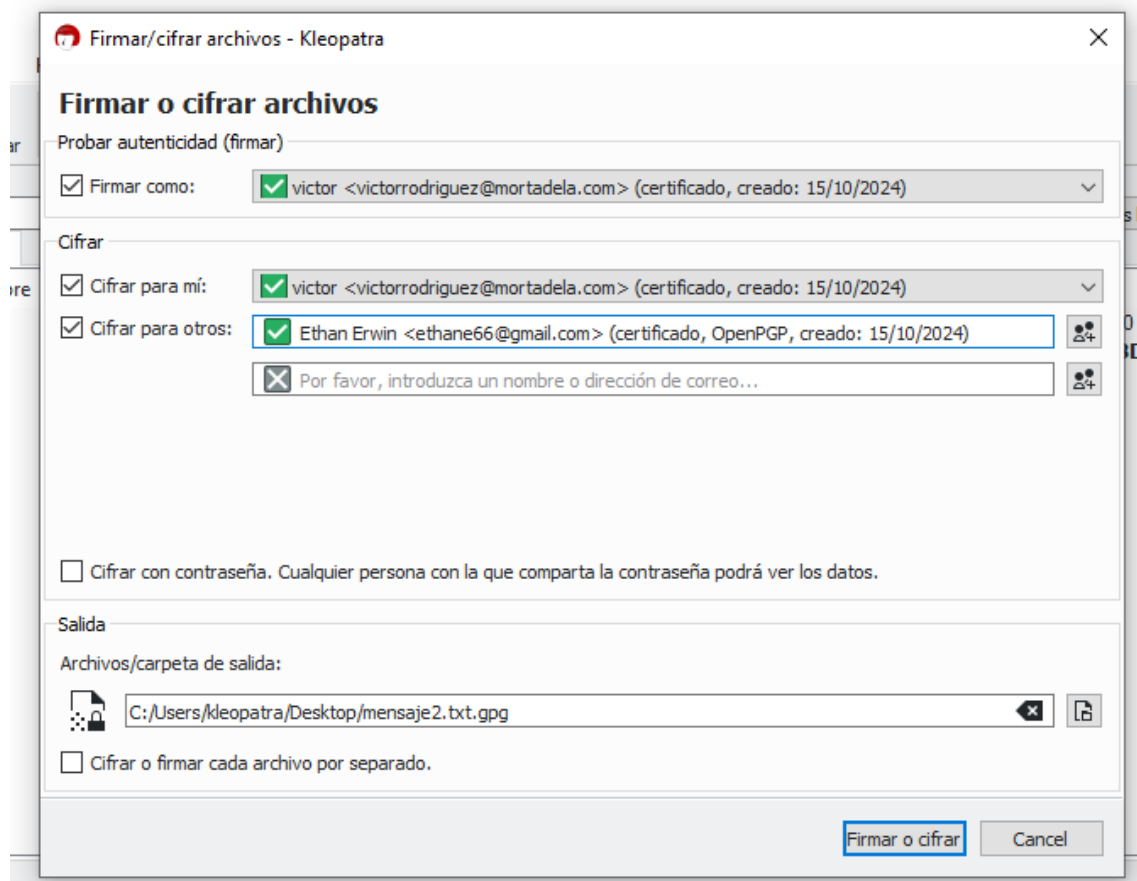
Crearemos unos archivos de texto con unos mensajes y los cifraremos con las claves para que nadie más que nosotros los podamos abrirlos

Encriptando con Kleopatra

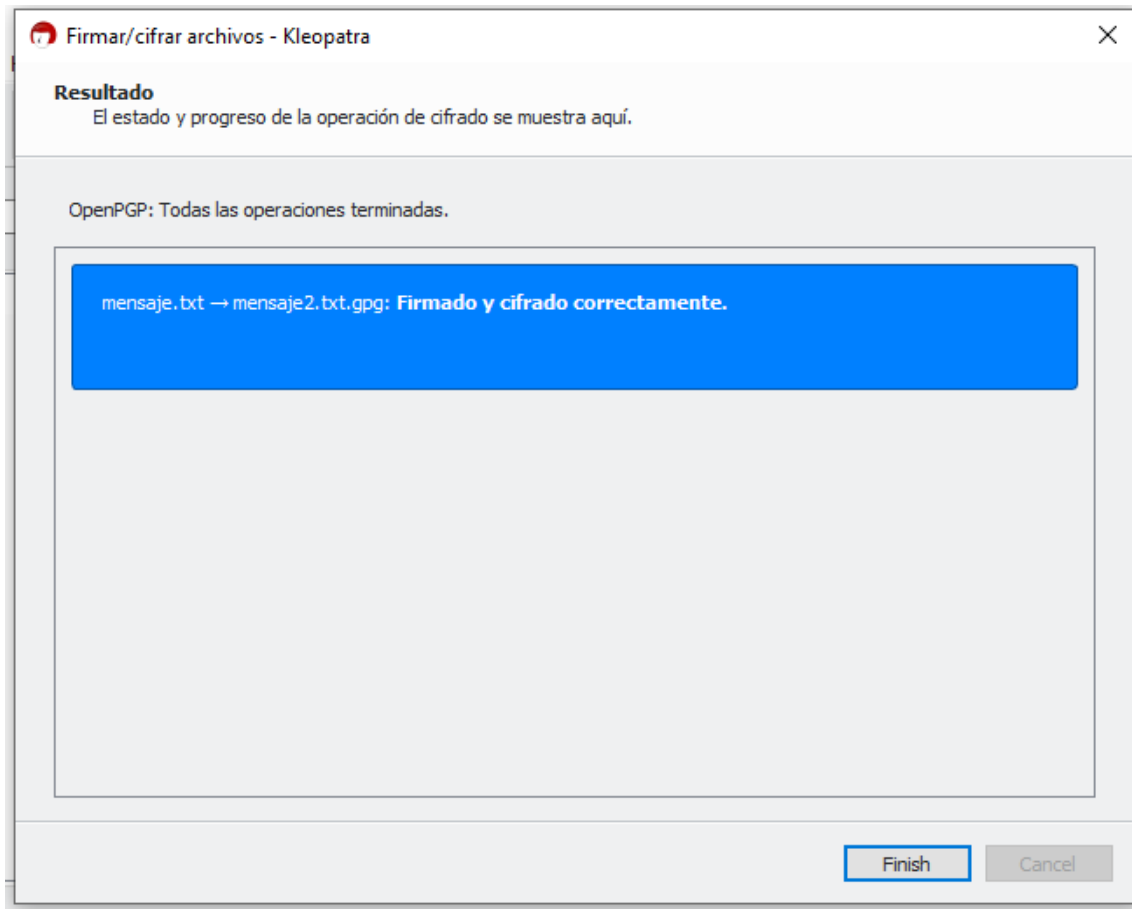
Crearemos un archivo con un texto cualquiera y lo tendremos a mano para encriptar



En Kleopatra seleccionaremos la opción de “Cifrar/Encriptar” y seleccionaremos nuestro archivo recién creado.



El programa nos dará un archivo .gpg que es el que tendremos que enviar para que nadie pueda acceder a él



Encriptando con GPG

Para poder encriptar con GPG el archivo de texto a enviar tendremos que usar la siguiente línea de comando

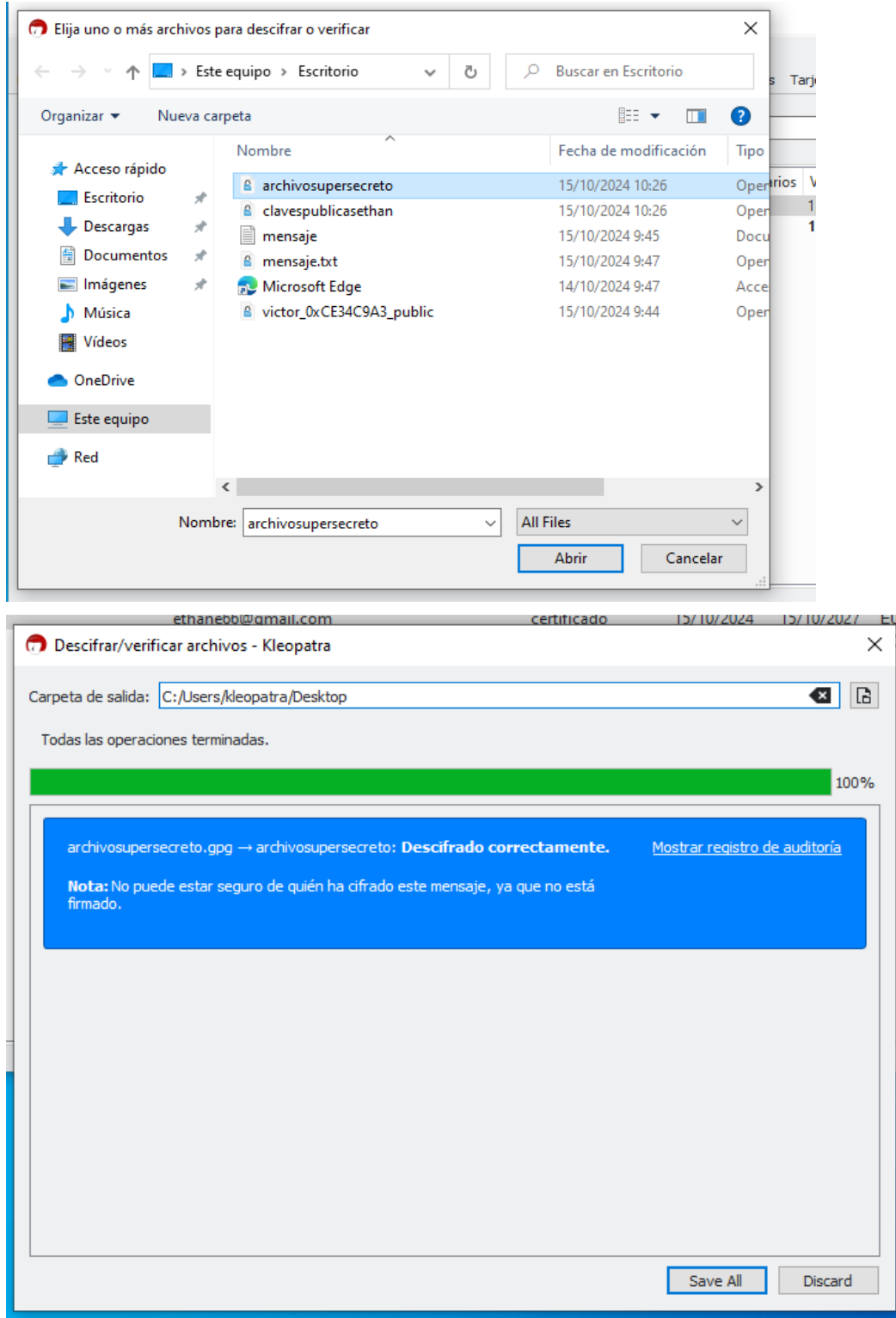
```
gpg -c archivotopsecret
```

```
[user@parrot]~[~/Documents]
$gpg -c archivotopsecret
[user@parrot]~[~/Documents]
$
```

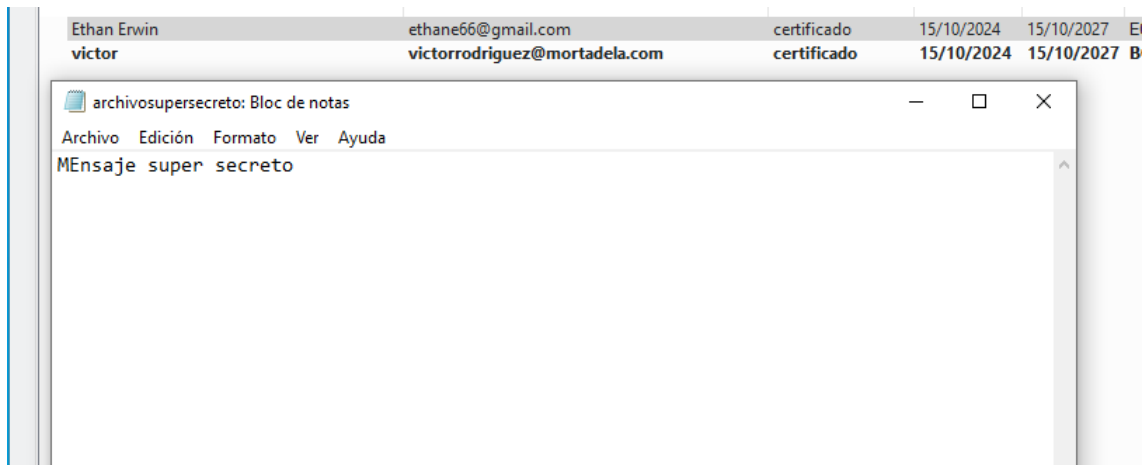
Desencriptar mensajes

Desencriptando con Kleopatra

En la interfaz de Kleopatra seleccionaremos la opción de “Descifrar/verificar” y escogeremos el archivo que Ethan nos ha compartido



Podemos comprobar que ahora el archivo es de texto legible



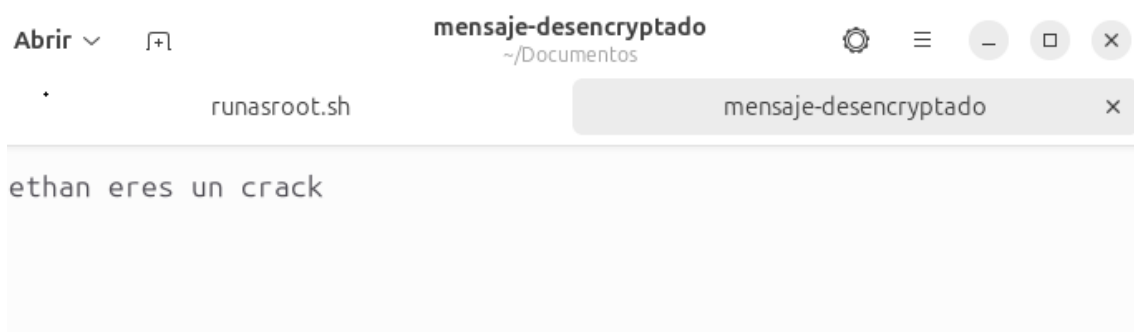
Desencriptando con GPG

Para desencriptar con GPG usaremos el siguiente comando con el fichero que Víctor nos ha compartido

```
gpg --decrypt mensaje2.txt.gpg > mensaje-desencryptado
```

```
ethan@ubuntu-redes:~/Documentos$ gpg --decrypt mensaje2.txt.gpg > mensaje-desencryptado
gpg: encrypted with cv25519 key, ID 0B2E4AF8A8F48724, created 2024-10-15
"victor <victorrodriguez@mortadela.com>"
gpg: encrypted with cv25519 key, ID AED8B5726ECFC618, created 2024-10-15
"Ethan Erwin <ethane66@gmail.com>"
gpg: Firmado el mar 15 oct 2024 10:48:04 CEST
gpg: usando EDDSA clave 79040CF7384EFAC629DBA541BC8D7C2ACE34C9A3
gpg: Firma correcta de "victor <victorrodriguez@mortadela.com>" [total]
```

Podemos comprobar que ahora el archivo es de texto legible



Conclusiones

En conclusión la encriptación de archivos es una manera de proteger la información sensible para aquellos que lo necesiten, garantizando que solo quienes poseen las claves correspondientes puedan acceder a los datos.

Con esto se consigue reducir el riesgo al acceso no autorizado de la información que se puede enviar a través de distintos medios, por ejemplo ataques Man-in-the-middle.