



Práctica 7

ENCRIPCIÓN DE
CLAVE PÚBLICA CON
GPG

José Ángel Dorado González
Seguridad Informática

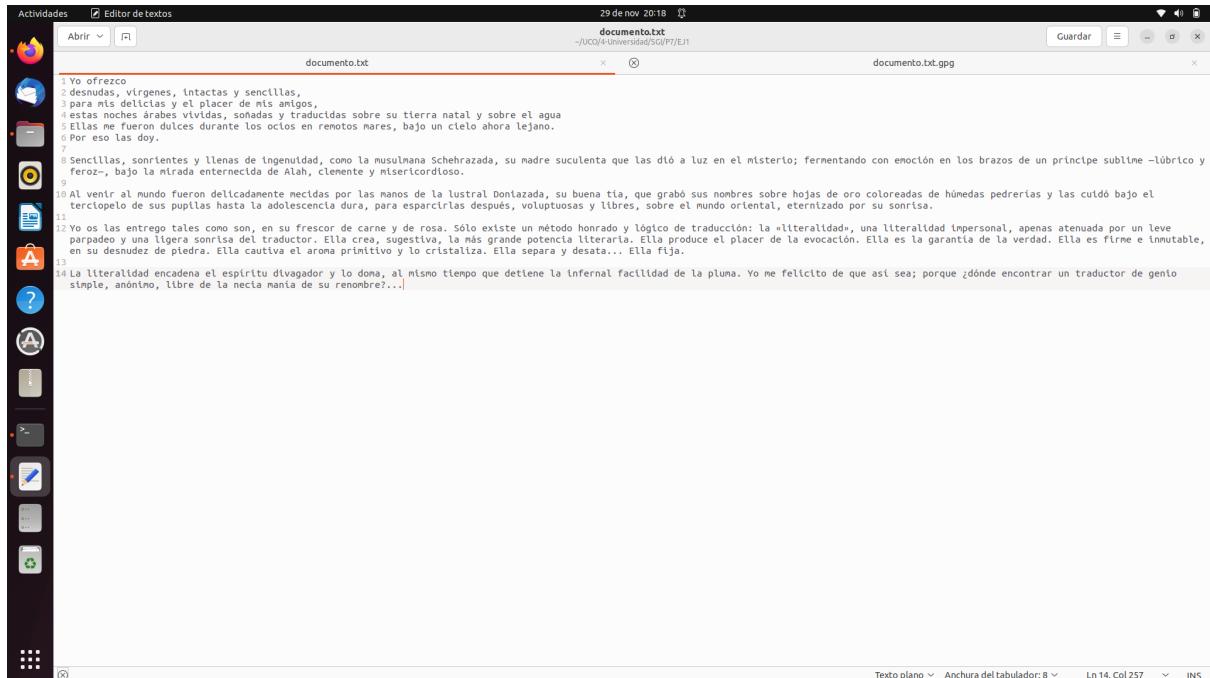


ÍNDICE

1º)Encriptación simétrica con GPG. Usar gpg para hacer encriptación de textos y documentos utilizando algoritmos de encriptación simétrica. Ver los algoritmos de encriptación disponibles y usar varios de ellos.	2
2º)Message Digest con GPG. Obtener el hash de textos y ficheros mediante gpg. Ver los algoritmos hash disponibles y usar varios de ellos.	6
3º)Encriptación de clave pública con GPG. Se trata de recrear un escenario parecido al que se ha visto en prácticas pero entre varios compañeros (el alumno puede crear varios usuarios en su ordenador y simular también así el escenario). Se generará un par de claves de prueba para firma y cifrado como se ha visto en prácticas. Estas claves serán de prueba y solo para uso durante la realización de este ejercicio. Se puede limitar su validez a un par de días por ejemplo.	9
4º)Exportar la clave creada y hacerla llegar a algunos de los compañeros de clase. Importar algunas claves públicas de otros usuarios y realizar cada uno de los ejercicios que se han visto en la sesión de prácticas: firma, cifrado, firma+cifrado, etc.	9
5º)Exportar la clave creada y hacerla llegar a algunos de los compañeros de clase. Importar algunas claves públicas de otros usuarios y realizar cada uno de los ejercicios que se han visto en la sesión de prácticas: firma, cifrado, firma+cifrado, etc.	15
6º)Como ejercicio adicional opcional, buscar la integración de GPG en vuestro cliente de correo electrónico favorito. Como por ejemplo Enigmail de Thunderbird.	16
Bibliografía	20

1º)Encriptación simétrica con GPG. Usar gpg para hacer encriptación de textos y documentos utilizando algoritmos de encriptación simétrica. Ver los algoritmos de encriptación disponibles y usar varios de ellos.

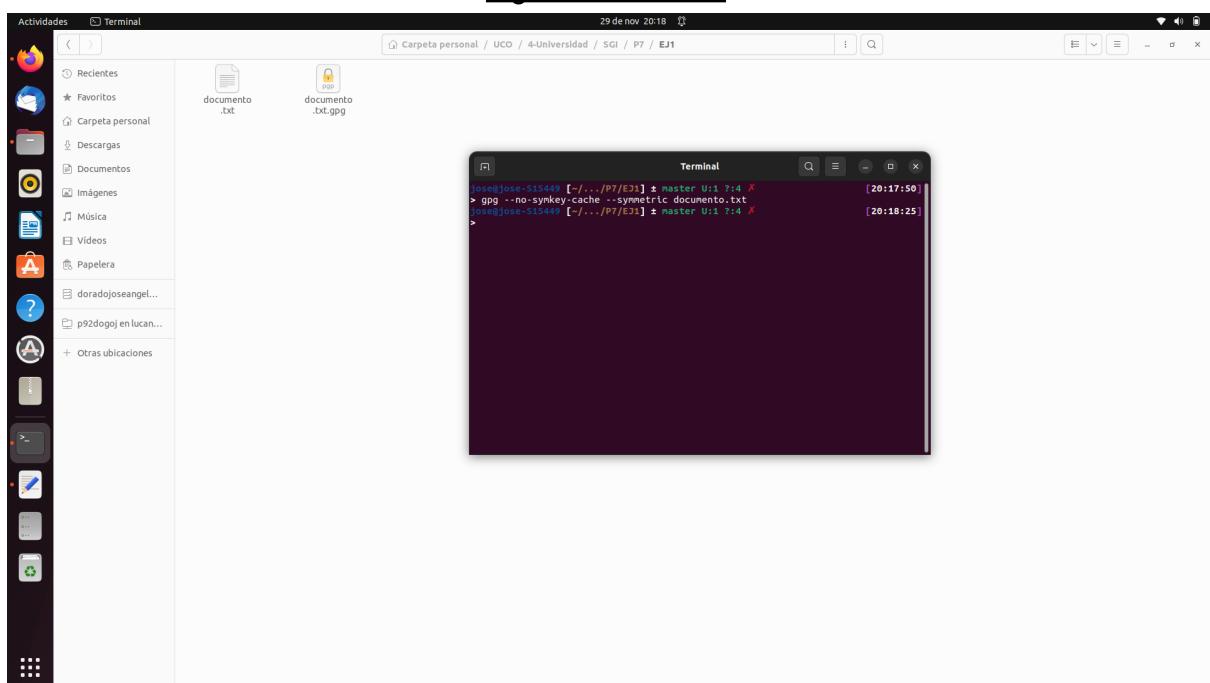
El texto que va a ser encriptado con diferentes algoritmos es el siguiente:



The screenshot shows a desktop environment with a central window titled "Editor de textos". The window contains a poem in Spanish. Above the poem, there is some metadata: "29 nov 20:18", "documento.txt", and "-/UCO/4-Universidad/SGI/P7/EJ1". Below the poem, there is a "Guardar" button. To the right of the main window, there is a smaller window titled "documento.txt.gpg". The desktop interface includes a vertical dock on the left with icons for various applications like a browser, file manager, and terminal. At the bottom, there is a toolbar with buttons for "Texto plano", "Anchura del tabulador: 8", "Ln 14, Col 257", and "INS".

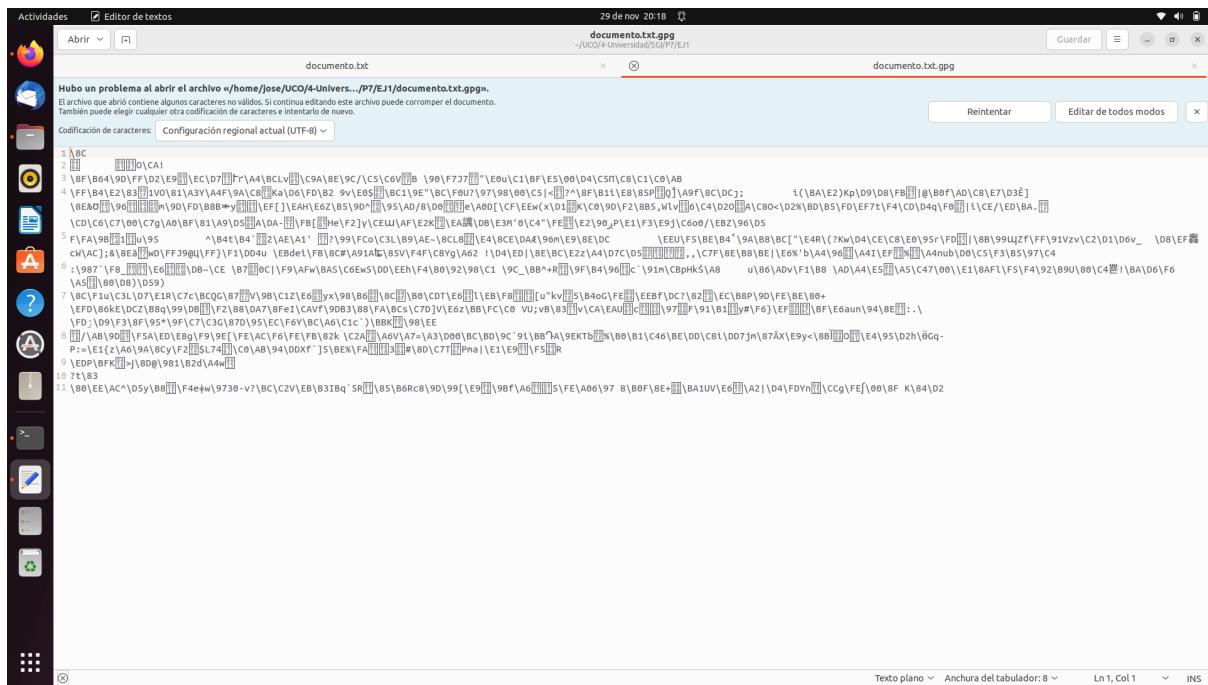
```
1 Yo ofrezco
2 desnudas, virgenes, intactas y sencillas,
3 para mis delicias y el placer de mis amigos,
4 estas noches árabes vividas, sonadas y traducidas sobre su tierra natal y sobre el agua
5 Ellas me fueron dulces durante los ojos en remotos mares, bajo un cielo ahora lejano.
6 Por eso las doy.
7 Sencillas, sonrientes y llenas de ingenuidad, como la musulmana Schehrazada, su madre suculenta que las dió a luz en el misterio; fermentando con emoción en los brazos de un príncipe sublime -lúbrico y feroz-, bajo la mirada enternecedora de Alah, Clemente y misericordioso.
8
9 Al venir al mundo fueron delicadamente meciidas por las manos de la lustral Doniazada, su buena tía, que grabó sus nombres sobre hojas de oro coloreadas de húmedas pedrerías y las cuidó bajo el terciopelo de sus pupilas hasta la adolescencia dura, para espaciarlas después, voluptuosas y libres, sobre el mundo oriental, eternizado por su sonrisa.
10
11 Yo os las entrego tales como son, en su frescor de carne y de rosa. Sólo existe un método honrado y lógico de traducción: la «literalidad», una literalidad impersonal, apenas atenuada por un leve parpadeo y una ligera sonrisa del traductor. Ella crea, sugestiva, la más grande potencia literaria. Ella produce el placer de la evocación. Ella es la garantía de la verdad. Ella es firme e inmutable, en su desnudez de piedra. Ella cautiva el aroma primitivo y lo cristaliza. Ella separa y desata... Ella fija.
12
13 La literalidad encadena el espíritu divagador y lo doma, al mismo tiempo que detiene la infernal facilidad de la pluma. Yo me felicito de que así sea; porque ¿dónde encontrar un traductor de genio simple, anónimo, libre de la necia manía de su renombre?...
```

Algoritmo AES256

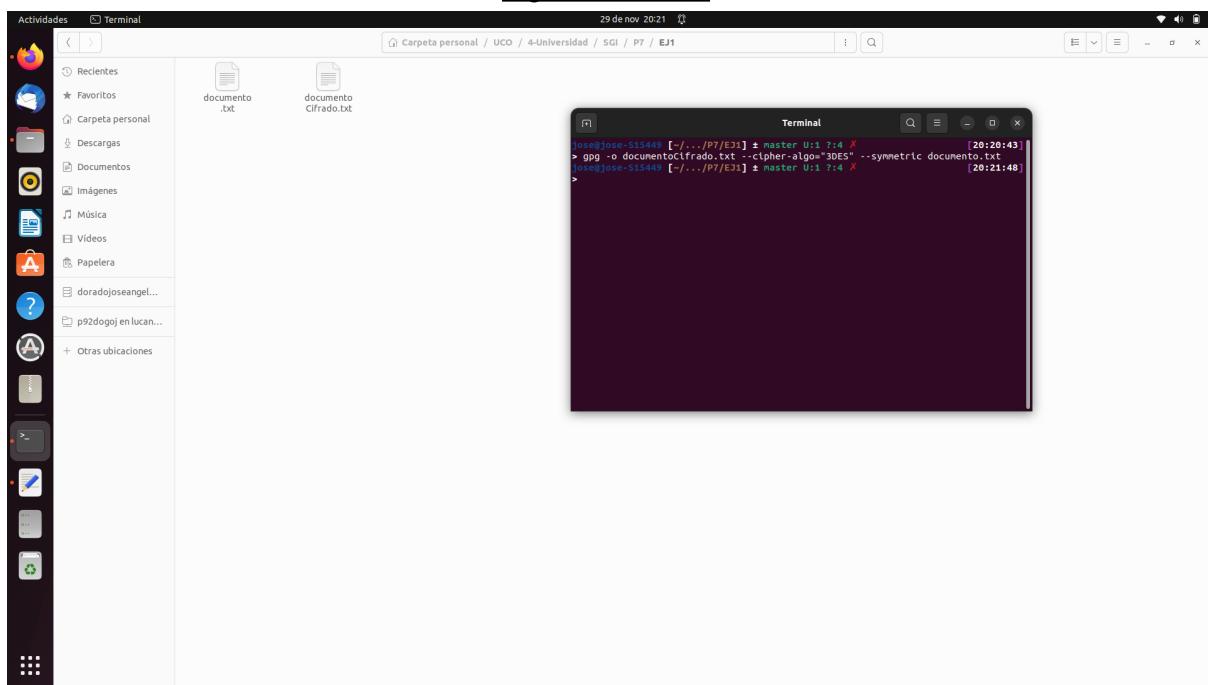


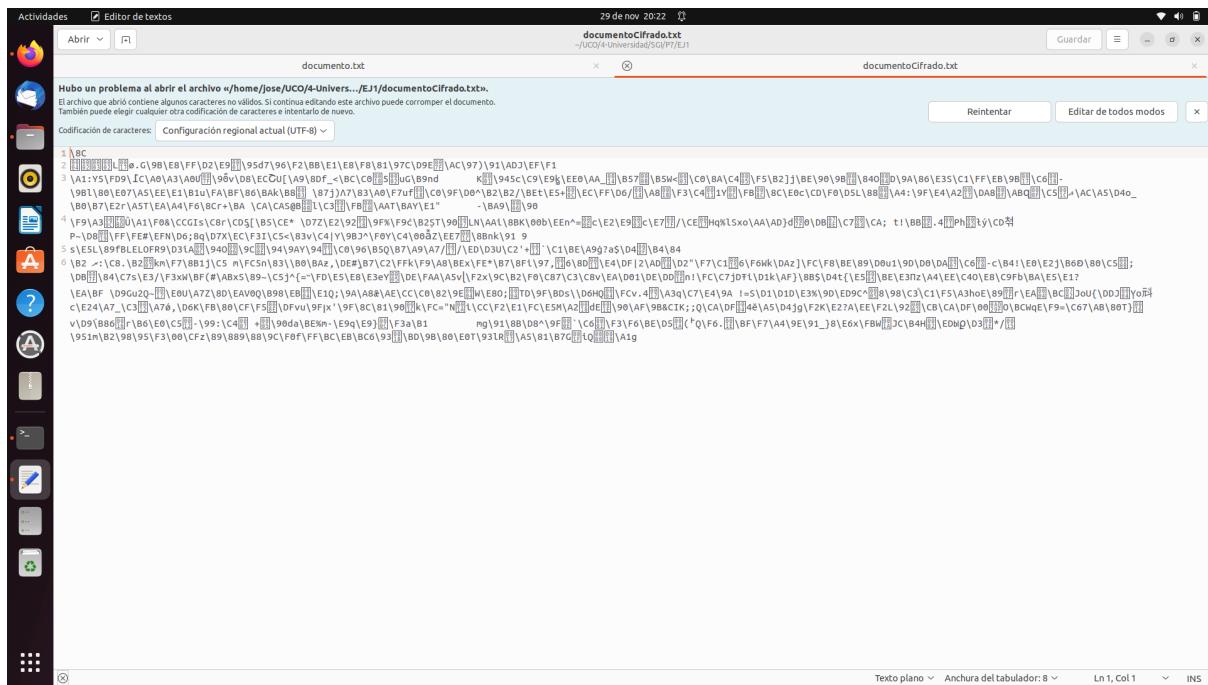
The screenshot shows a desktop environment with a file manager window on the left and a terminal window on the right. The file manager displays a folder structure with a file named "documento.txt" and a file named "documento.txt.gpg". The terminal window shows command-line output related to GPG encryption:

```
[root@jose-515449 ~]# ./P7/EJ1 [root@jose-515449 ~]# gpg -no-symkey-cache --symmetric documento.txt [root@jose-515449 ~]# ./P7/EJ1 [root@jose-515449 ~]#
```

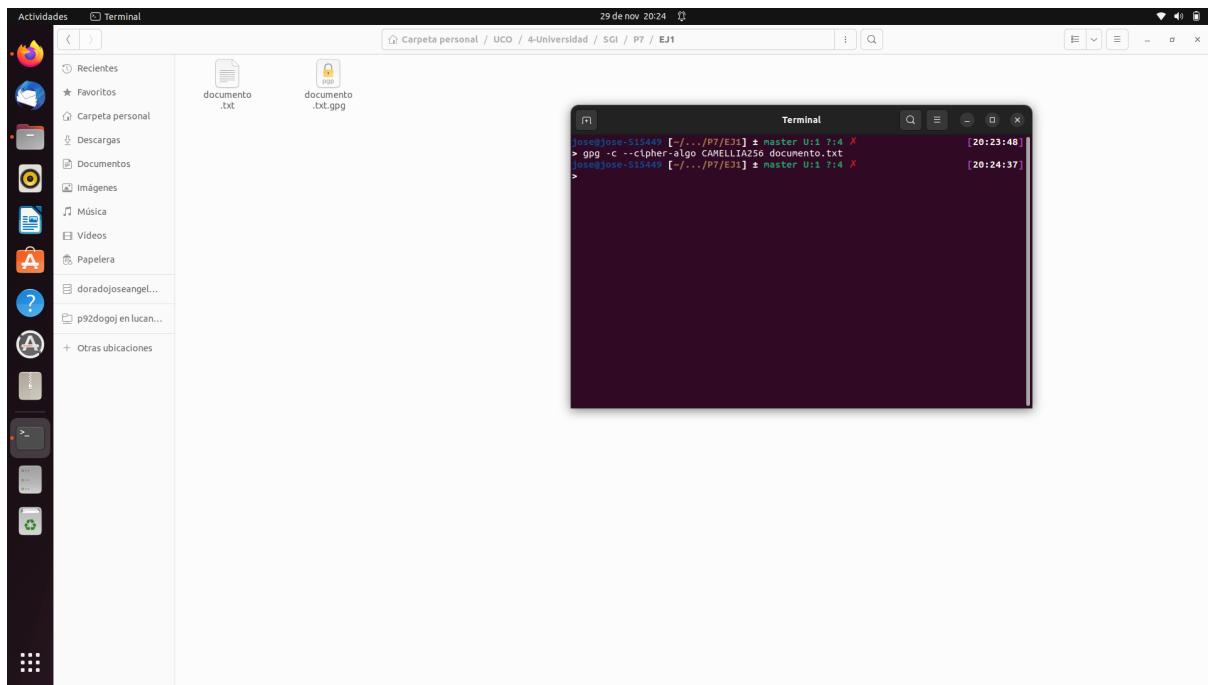


Algoritmo 3DES





Algoritmo CAMELLIA256



A screenshot of a Windows operating system desktop. In the center, there is a window titled "documento.txt.gpg" which is a GPG-encrypted file. The file's content is displayed as a large amount of illegible, garbled text consisting mostly of non-ASCII characters. The desktop background is plain white. Along the bottom edge, the taskbar is visible with several pinned icons: a browser (Microsoft Edge), File Explorer, Control Panel, Task View, Start button, and system status icons. On the left side, there is a vertical dock of pinned application icons.

El cifrado de documentos se realizará con el mismo texto que he utilizado con anterioridad.
El resultado es el siguiente:

The screenshot shows a Linux desktop environment with several windows open:

- Activities**: A dock with icons for Home, Terminal, Dash, and others.
- Terminal**: A terminal window titled "Terminal" showing a session between user "jose@jose-515449" and "jose@jose-515449". The session details the use of GPG to encrypt files "documento.txt" and "pruebaSi.odt" using a symmetric key derived from a passphrase. It also shows attempting to decrypt "pruebaSi.odt" with the wrong recipient.
- File Manager**: A file browser showing files "documento.txt", "documento.txt.gpg", "pruebaSi.odt", and "pruebaSi.odt.gpg".
- Privacy Guard**: A browser window titled "Primeros Pasos" with the URL "https://www.privacyguard.net/primeros-pasos". The page discusses using a public key as a physical safe to protect documents, mentioning multiple encryptions and the use of a recipient's public key for the final step.
- Browser Tabs**: Tabs for "Cifrar y descifrar docx", "SI - P7 - Documentos", and "practicaEncriptacionDeClave".

Terminal Session Details:

```
jose@jose-515449:~$ ls
documento.txt documento.txt.gpg pruebaSi.odt
jose@jose-515449:~$ gpg --output pruebaSi.odt.gpg --encrypt --recipient 94175BC3C078B06CE68368735
1475C43723F749 cartaPepe.odt --output orden no encontrado
jose@jose-515449:~$ gpg --output pruebaSi.odt.gpg --encrypt --recipient 94175BC3C078B06CE68368735
1475C43723F749 cartaPepe.odt --output orden no encontrado
jose@jose-515449:~$ gpg --output pruebaSi.odt.gpg --encrypt --recipient 94175BC3C078B06CE68368735
1475C43723F749 pruebaSi.odt --output orden no encontrado
jose@jose-515449:~$ gpg --output pruebaSi.odt.gpg --encrypt --recipient 94175BC3C078B06CE68368735
1475C43723F749 pruebaSi.odt --output orden no encontrado
jose@jose-515449:~$
```

Privacy Guard Page Content:

Ir en una clave pública como en una caja fuerte de seguridad. Cuando un remitente cifra un documento usando esta se cifra varias veces. La parte correspondiente a la clave privada, esto es, el destinatario, es la combinación cifrado usando la clave pública asociada al cifrado.

Javier para Javier, lo haría usando la clave pública de Javier, y él lo comprime como medida adicional de seguridad, aparte de cifrarlo.

Ir como entrada el nombre del documento que se desea cifrar o, si éste se cifra por sí mismo a menos que haya incluido su propia clave privada.

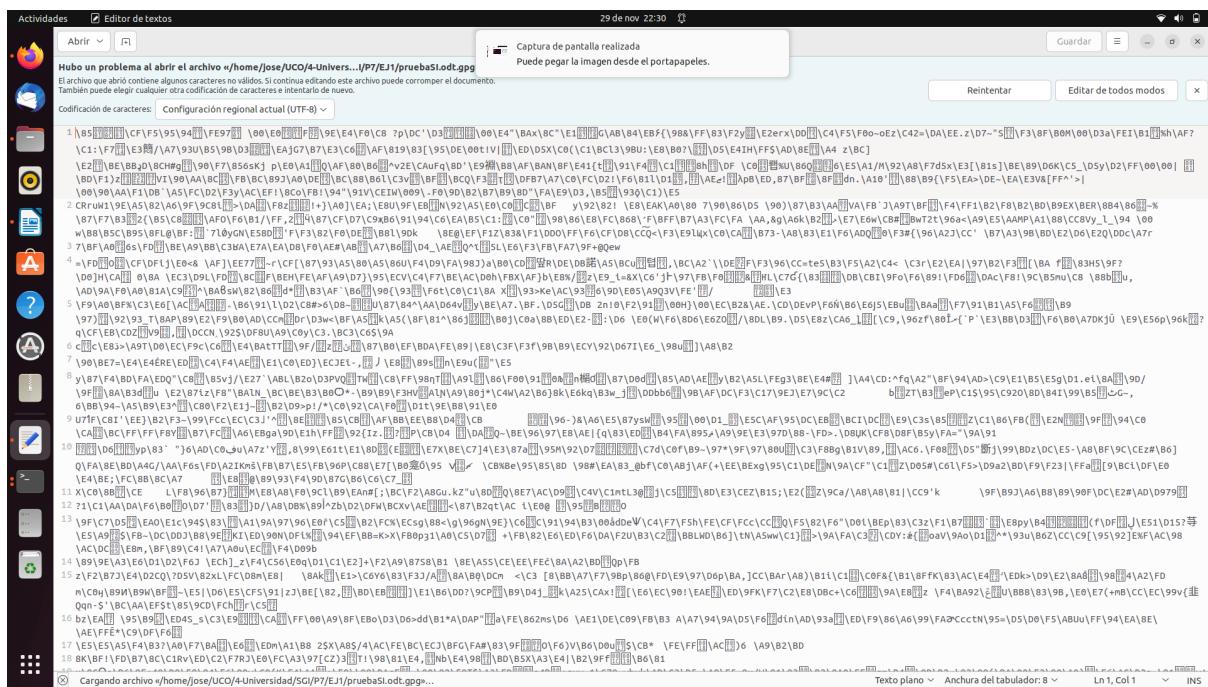
El documento cifrado sólo puede ser descifrado por alguien con la clave privada que lo cifró por sí mismo a menos que haya incluido su propia clave privada.

que en el proceso de cifrado, el documento a descifrar es la entrada, y

Footer:

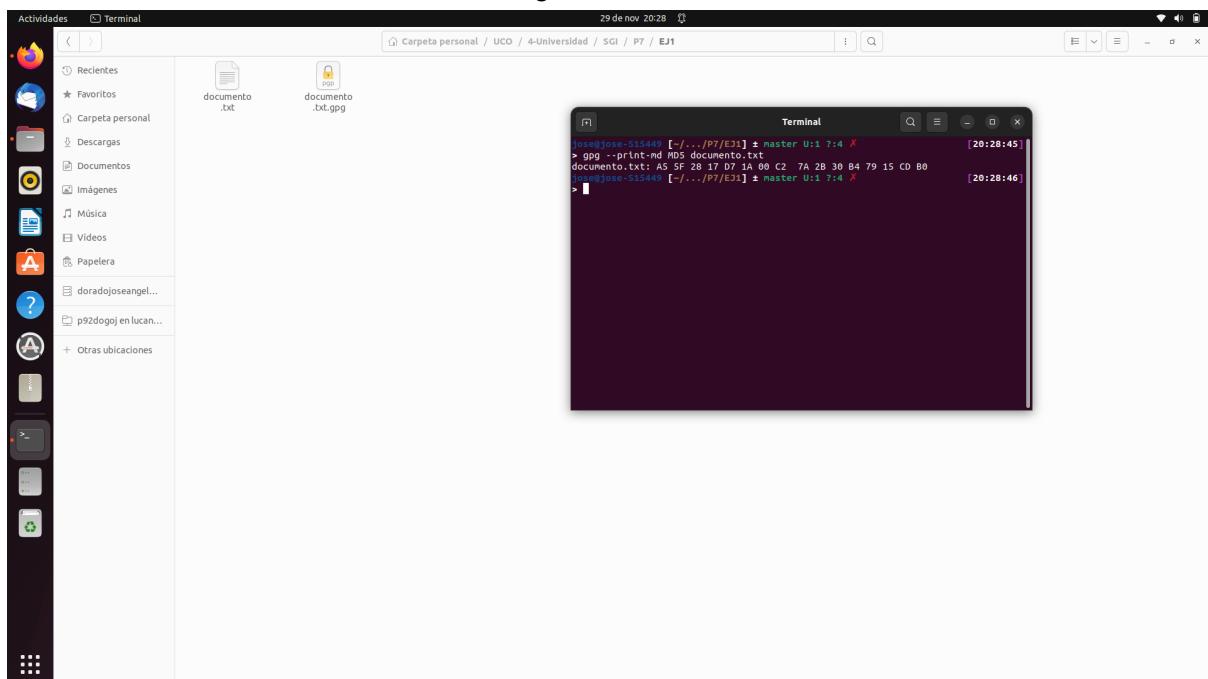
Cifrar y descifrar documentos para protegerlos. La clave que se usa para el cifrado simétrico deriva de la frase que se está usando para proteger la clave privada. El cifrado simétrico es útil para asegurar documentos ndo la opción [-symmetric](#).

Firmar y verificar firmas

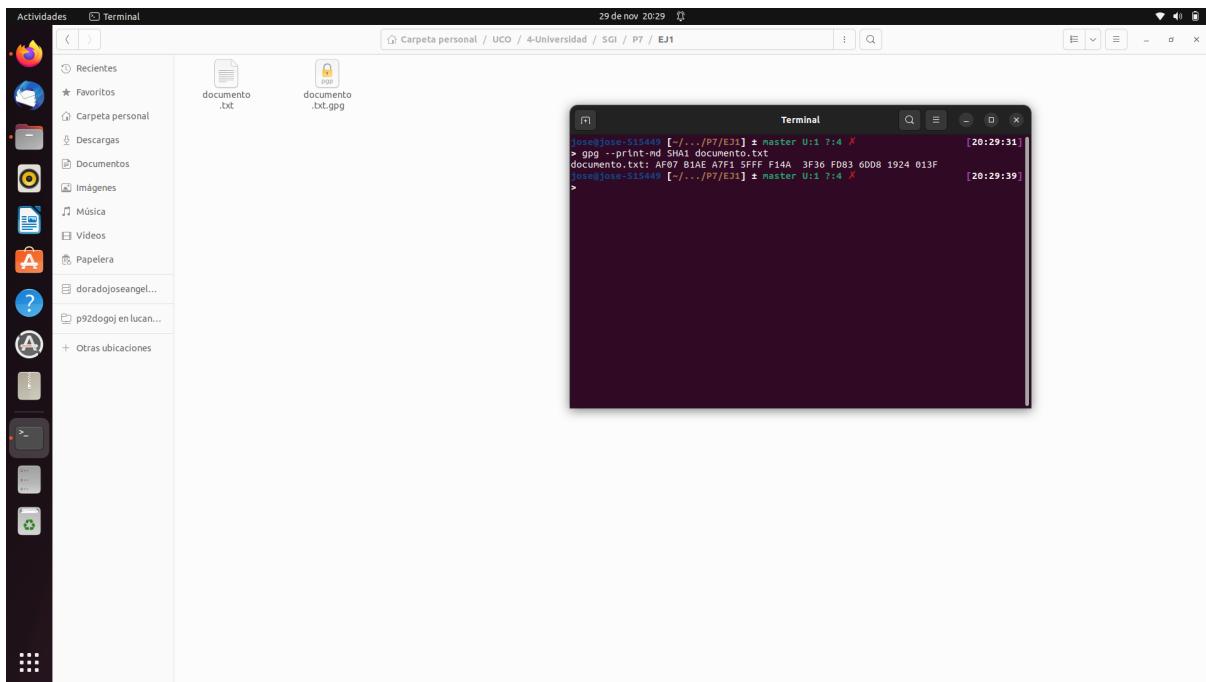


2º) Message Digest con GPG. Obtener el hash de textos y ficheros mediante gpg. Ver los algoritmos hash disponibles y usar varios de ellos.

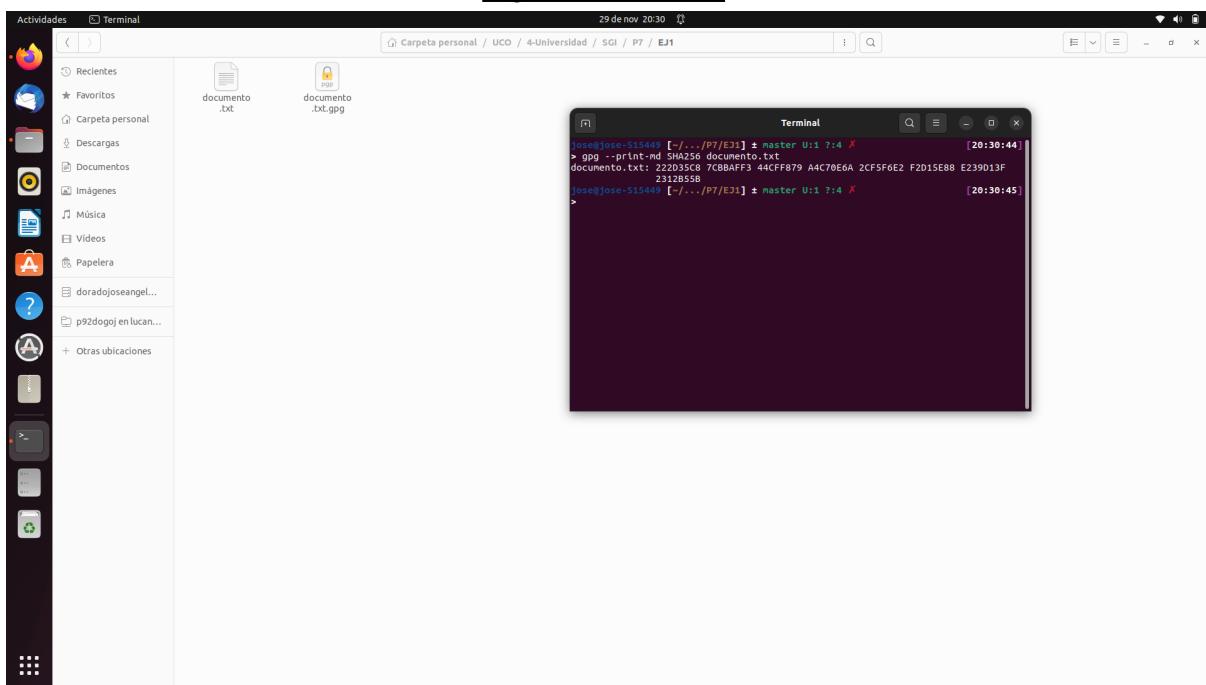
Algoritmo MD5



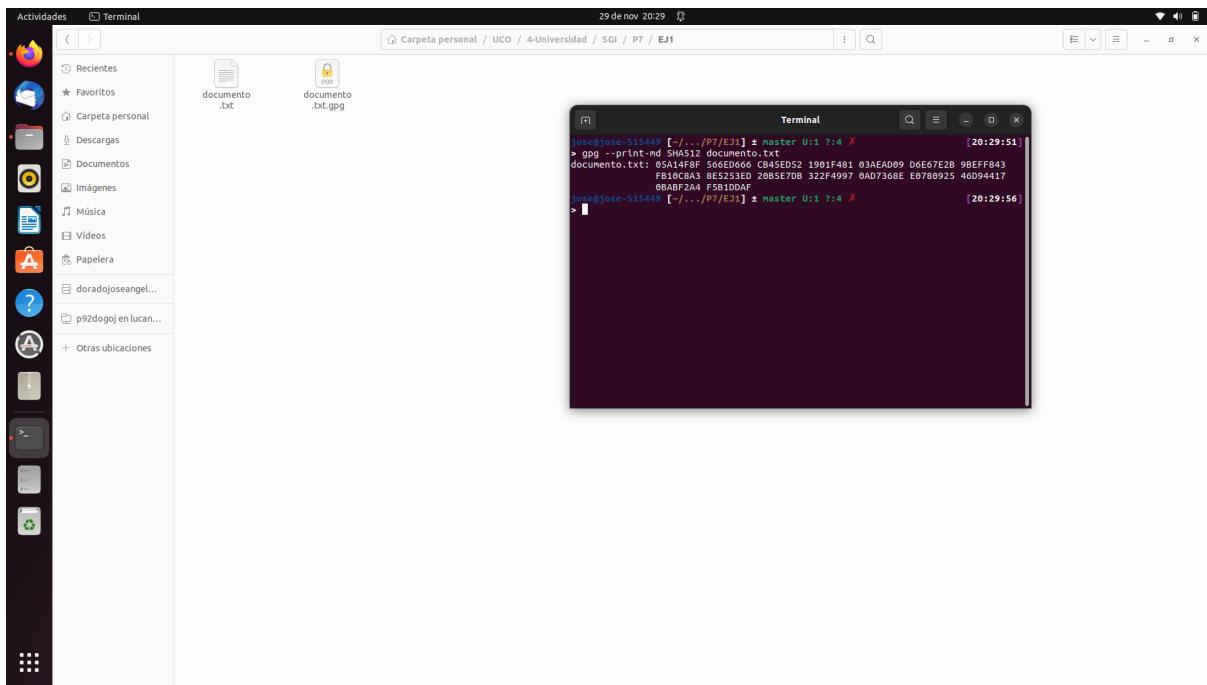
Algoritmo SHA1



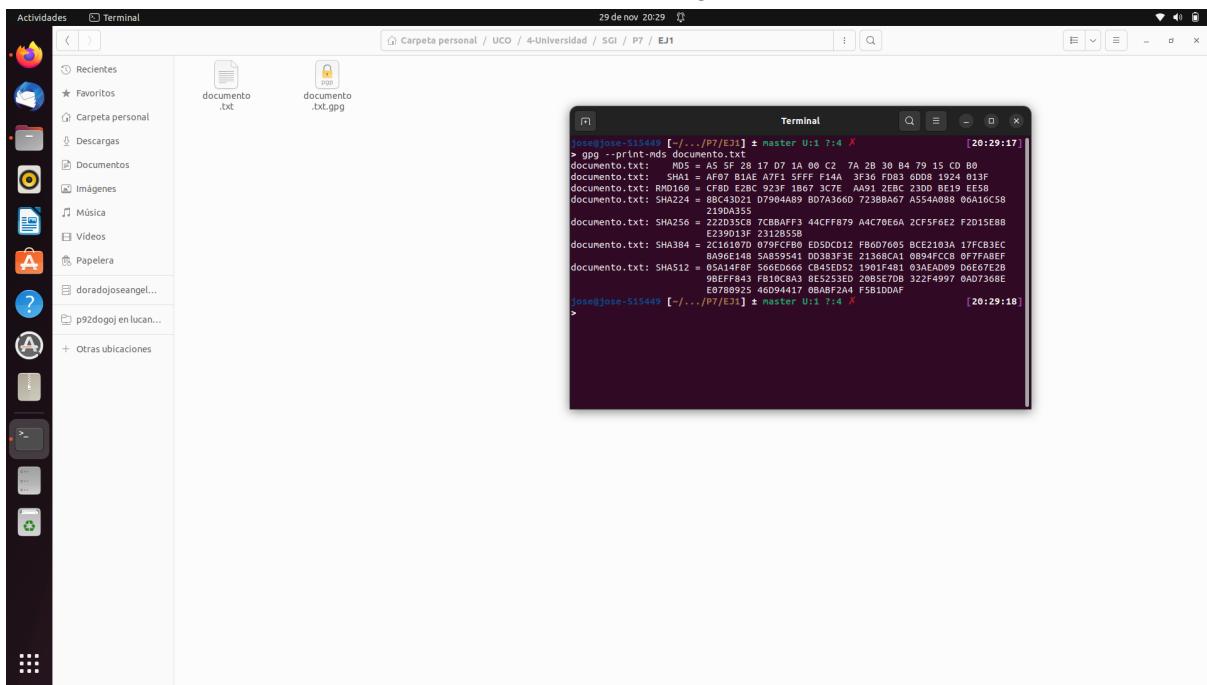
Algoritmo SHA256



Algoritmo SHA512

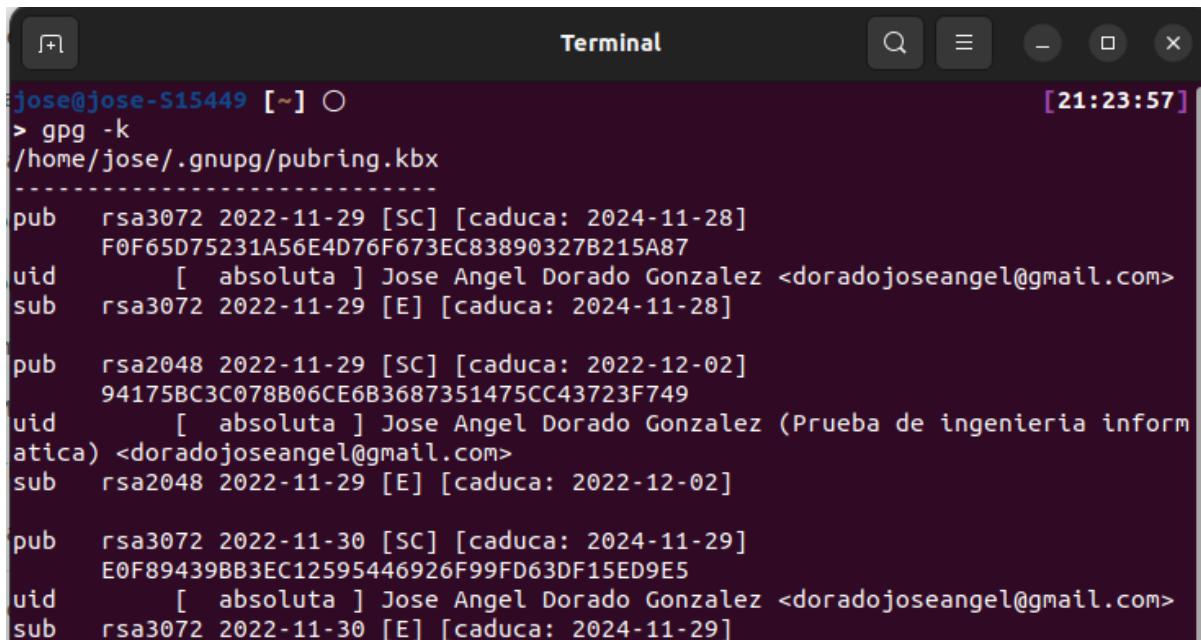


Listado de todos los algoritmos



3º)Encriptación de clave pública con GPG. Se trata de recrear un escenario parecido al que se ha visto en prácticas pero entre varios compañeros (el alumno puede crear varios usuarios en su ordenador y simular también así el escenario). Se generará un par de claves de prueba para firma y cifrado como se ha visto en prácticas. Estas claves serán de prueba y solo para uso durante la realización de este ejercicio. Se puede limitar su validez a un par de días por ejemplo.

Para este ejercicio debemos crear un par de claves, se hace mediante el siguiente comando **\$gpg --gen-key**. A continuación aparecen las que tengo creadas en mi ordenador:



```
jose@jose-S15449:~$ > gpg -k /home/jose/.gnupg/pubring.kbx
-----
pub rsa3072 2022-11-29 [SC] [caduca: 2024-11-28]
F0F65D75231A56E4D76F673EC83890327B215A87
uid      [ absoluta ] Jose Angel Dorado Gonzalez <doradojoseangel@gmail.com>
sub rsa3072 2022-11-29 [E] [caduca: 2024-11-28]

pub rsa2048 2022-11-29 [SC] [caduca: 2022-12-02]
94175BC3C078B06CE6B3687351475CC43723F749
uid      [ absoluta ] Jose Angel Dorado Gonzalez (Prueba de ingenieria informatica) <doradojoseangel@gmail.com>
sub rsa2048 2022-11-29 [E] [caduca: 2022-12-02]

pub rsa3072 2022-11-30 [SC] [caduca: 2024-11-29]
E0F89439BB3EC12595446926F99FD63DF15ED9E5
uid      [ absoluta ] Jose Angel Dorado Gonzalez <doradojoseangel@gmail.com>
sub rsa3072 2022-11-30 [E] [caduca: 2024-11-29]
```

4º)Exportar la clave creada y hacerla llegar a algunos de los compañeros de clase. Importar algunas claves públicas de otros usuarios y realizar cada uno de los ejercicios que se han visto en la sesión de prácticas: firma, cifrado, firma+cifrado, etc.

Para exportar la clave debemos utilizar el comando **\$gpg --export-secret-key --armor KeyID**, para saber el KeyID debemos listar las claves mediante el comando **gpg --list-secret-keys --keyid-format LONG**, una vez que se muestre debemos coger el identificador que aparece en sec de la clave deseada.

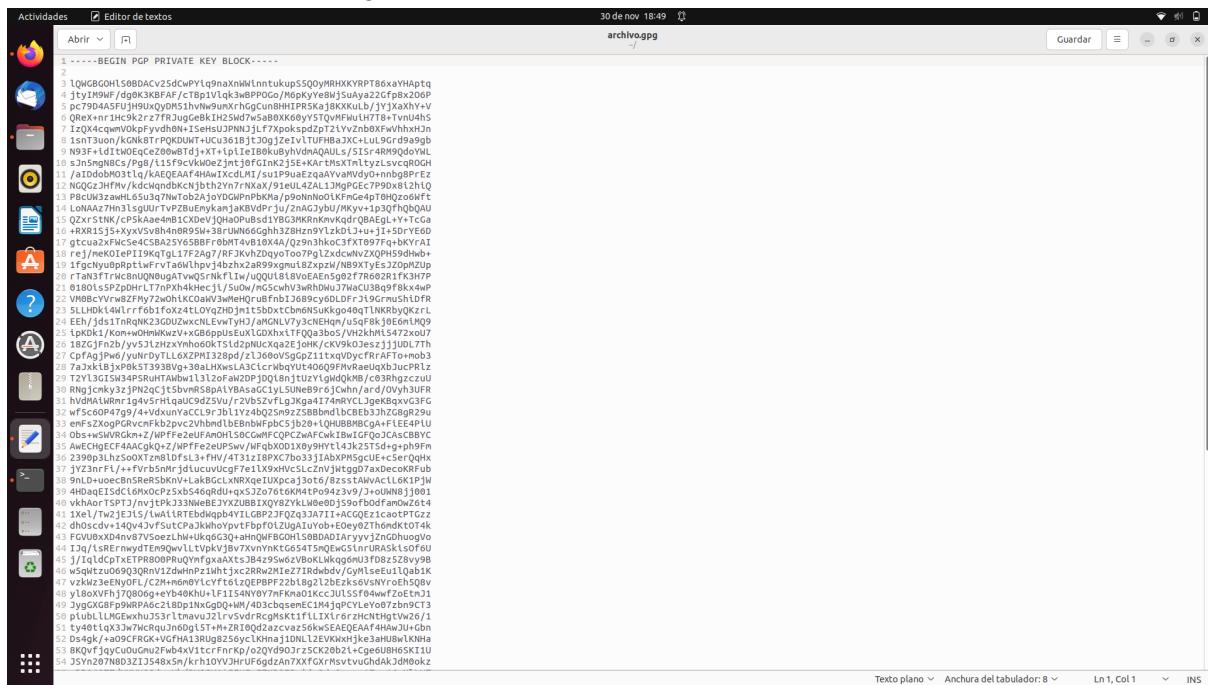
Una vez hecho esto debemos exportar la clave mediante un archivo, en mi caso lo haré mediante un archivo cifrado.

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Terminal". The terminal window has a dark background and displays the following command history:

```
jose@jose-515449: ~
> gpg --export-secret-key --armor F99FD63DF15ED9E5 >> archivo.gpg
jose@jose-515449: ~
```

The terminal window is part of a desktop interface with a dock on the left containing icons for various applications like a browser, file manager, and system tools. The top bar shows the date and time as "30 de nov 18:49". A status bar at the bottom right shows the current time as "[18:48:54]" and "[18:49:03]".

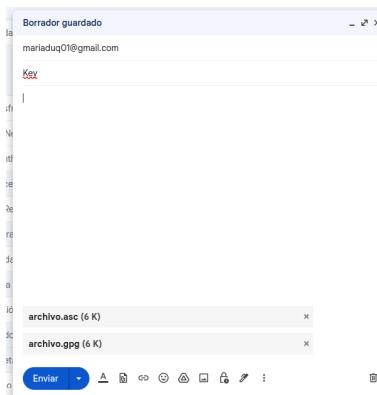
El archivo cifrado sería el siguiente:



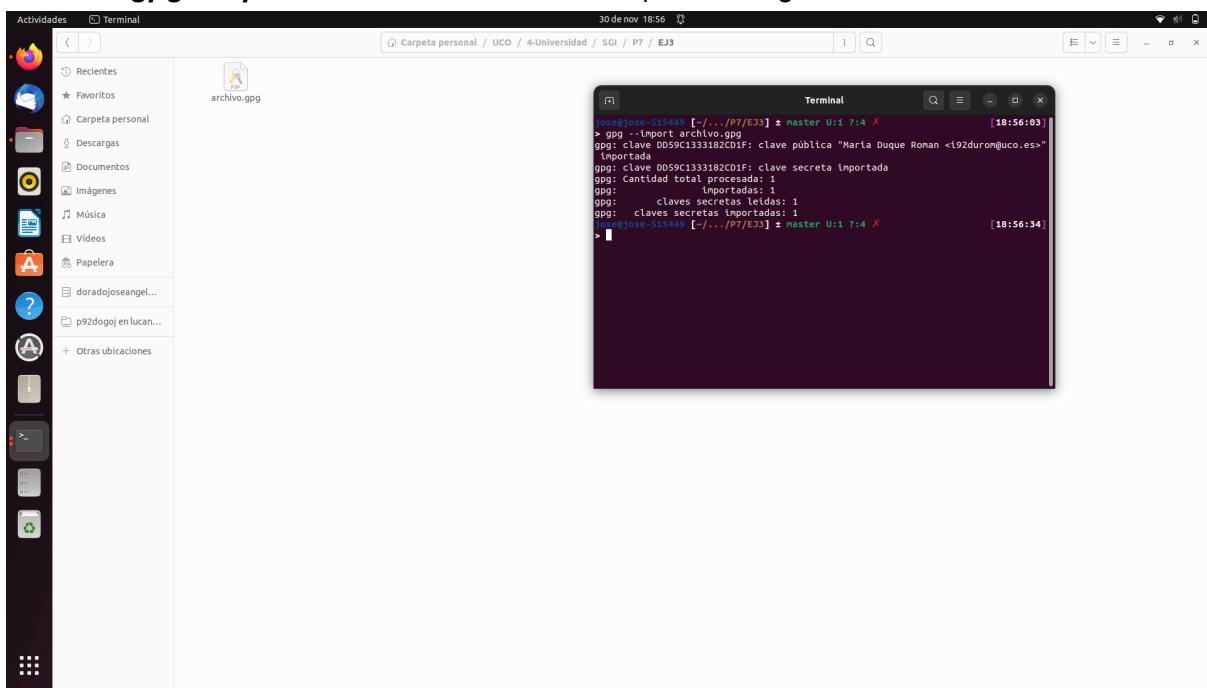
```
-----BEGIN PGP PRIVATE KEY BLOCK-----  
-----END PGP PRIVATE KEY BLOCK-----  
-----BEGIN PGP PUBLIC KEY BLOCK-----  
-----END PGP PUBLIC KEY BLOCK-----
```

The terminal window contains a large amount of encrypted data, starting with the PGP header and ending with the PGP footer.

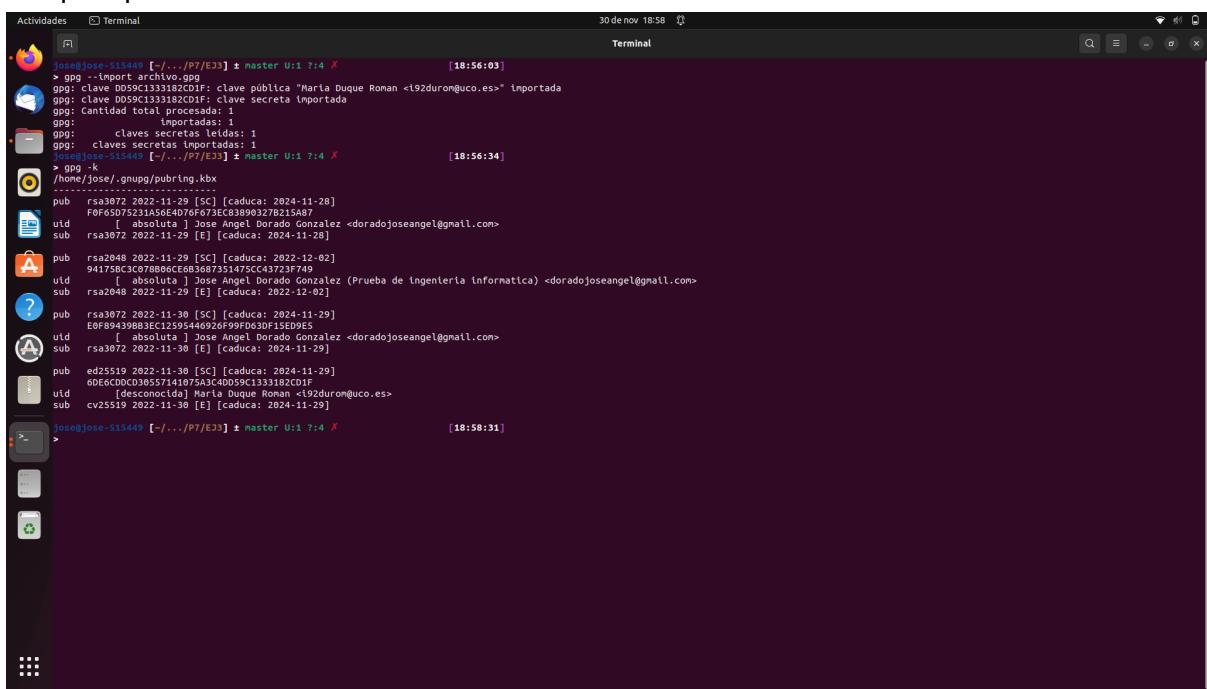
Ahora enviaré dicho archivo por correo a mi compañero:



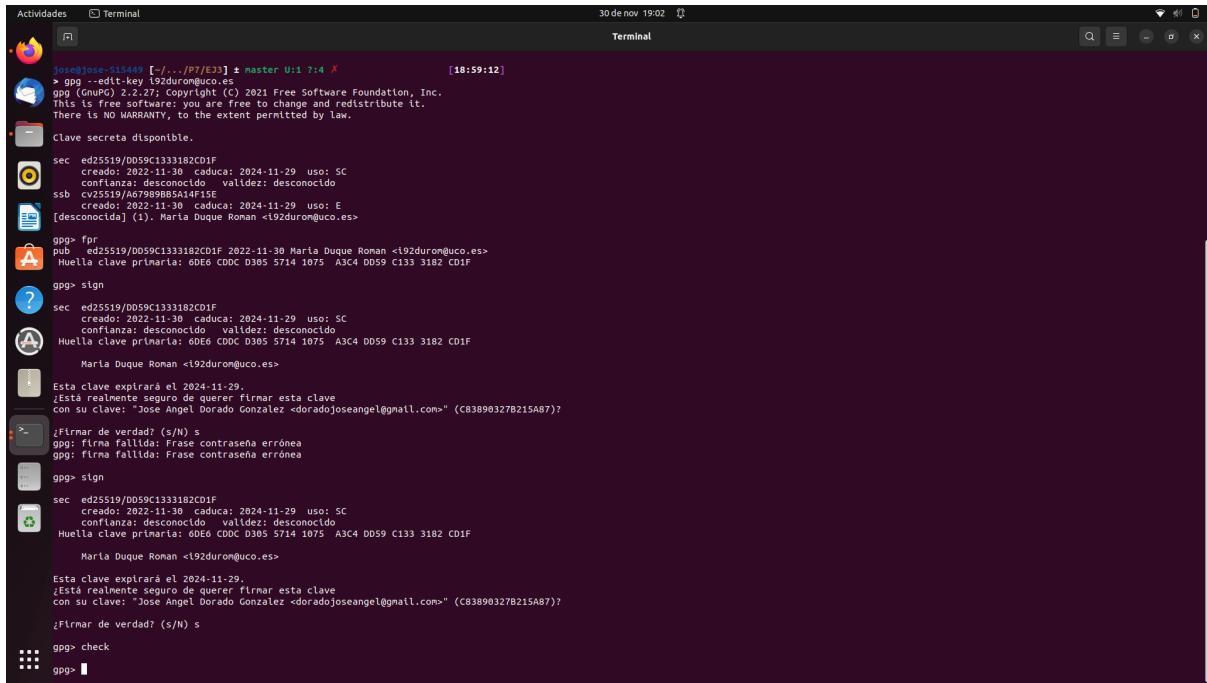
Para importar la clave del compañero debemos descargar el archivo y utilizar el siguiente comando ***gpg –import archivo***, el archivo es el que contenga la clave.



Con esto ya tendríamos la clave importada, si las listamos con el comando ***gpg -k*** podemos ver que aparece:



Ahora procederemos a firmar esta clave editándola con el comando ***gpg –edit-key correoElectrónico***.



The screenshot shows a terminal window titled "Terminal" with the command "gpg --edit-key i92durom@uco.es" running. The session starts with the GPG copyright notice. It then lists two keys: a secret key (sec) and a public key (pub). Both keys have a creation date of 2022-11-30 and an expiration date of 2024-11-29. The secret key's usage is SC (Sign and Certify), and its fingerprint is ed25519/D059C1333182CD1F. The public key's usage is E (Encrypt), and its fingerprint is ed25519/D059C1333182CD1F. The user information for both keys is "Maria Duque Roman <i92durom@uco.es>". The "gpg> sign" command is issued, followed by a question about signing the key. The user responds with "s" (yes). The process then asks for a passphrase, which is entered twice. Finally, the user is prompted to "Firmar de verdad? (S/N)" and enters "s" again. The session ends with "gpg> check".

```
Actividades Terminal 30 de nov 19:02 Terminal
inse@Inse-S15449 [-/.../PT/E3] ± master U:1 7:4 X [18:59:12]
> gpg --edit-key i92durom@uco.es
gpg (GnuPG) 2.2.27; copyright (C) 2021 Free Software Foundation, Inc.
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Clave secreta disponible.

sec ed25519/D059C1333182CD1F
    creada: 2022-11-30 caduca: 2024-11-29 uso: SC
    confianza: desconocido validez: desconocido
    sbb cv25519/A67989BB5A14F1SE
    creado: 2022-11-30 caduca: 2024-11-29 uso: E
    [desconocida] (1). Maria Duque Roman <i92durom@uco.es>

gpg> fpr
pub ed25519/D059C1333182CD1F 2022-11-30 Maria Duque Roman <i92durom@uco.es>
Huella clave primaria: 60E6 CDDC D305 5714 1075 A3C4 D059 C133 3182 CD1F

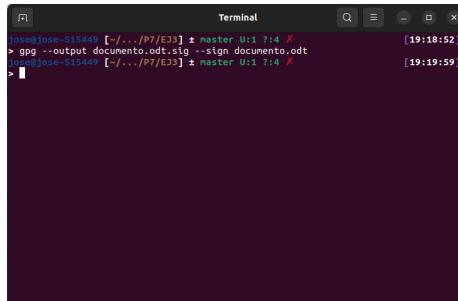
gpg> sign
? sec ed25519/D059C1333182CD1F
    creada: 2022-11-30 caduca: 2024-11-29 uso: SC
    confianza: desconocido validez: desconocido
    Huella clave primaria: 60E6 CDDC D305 5714 1075 A3C4 D059 C133 3182 CD1F
    Maria Duque Roman <i92durom@uco.es>

Esta clave expirará el 2024-11-29.
¿Está realmente seguro de querer firmar esta clave
con su clave: "Jose Angel Dorado Gonzalez <doradojoseangel@gmail.com>" (C838903278215A87)?
:firmar de verdad? (S/N) s
gpg: firma fallida: Frase contraseña errónea
gpg: firma fallida: Frase contraseña errónea

gpg> sign
sec ed25519/D059C1333182CD1F
    creada: 2022-11-30 caduca: 2024-11-29 uso: SC
    confianza: desconocido validez: desconocido
    Huella clave primaria: 60E6 CDDC D305 5714 1075 A3C4 D059 C133 3182 CD1F
    Maria Duque Roman <i92durom@uco.es>

Esta clave expirará el 2024-11-29.
¿Está realmente seguro de querer firmar esta clave
con su clave: "Jose Angel Dorado Gonzalez <doradojoseangel@gmail.com>" (C838903278215A87)?
:Firmar de verdad? (S/N) s
gpg> check
gpg> [1]
```

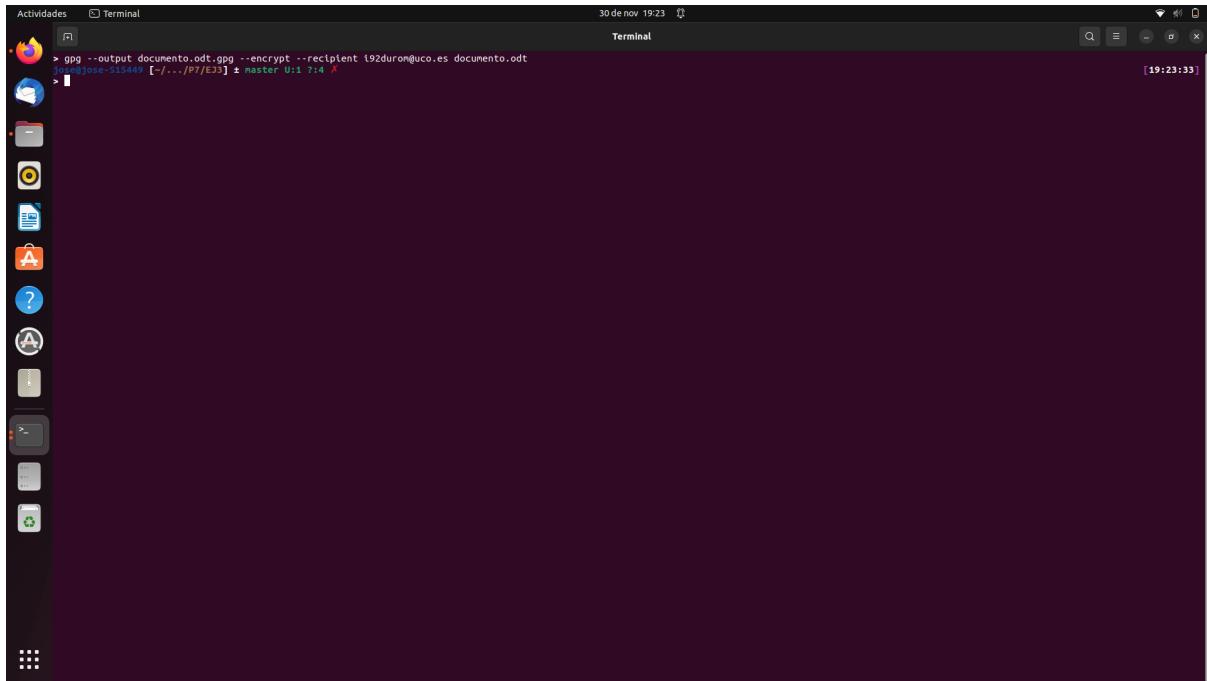
Ahora podemos firmar un documento mediante el comando ***gpg –output documento.odt.sig –sign documento.odt***:



The screenshot shows a terminal window titled "Terminal" with the command "gpg --output documento.odt.sig --sign documento.odt" running. The session starts with the GPG copyright notice. The user then types "gpg --output documento.odt.sig --sign documento.odt" and presses Enter. The terminal then displays the command again, indicating it is being processed.

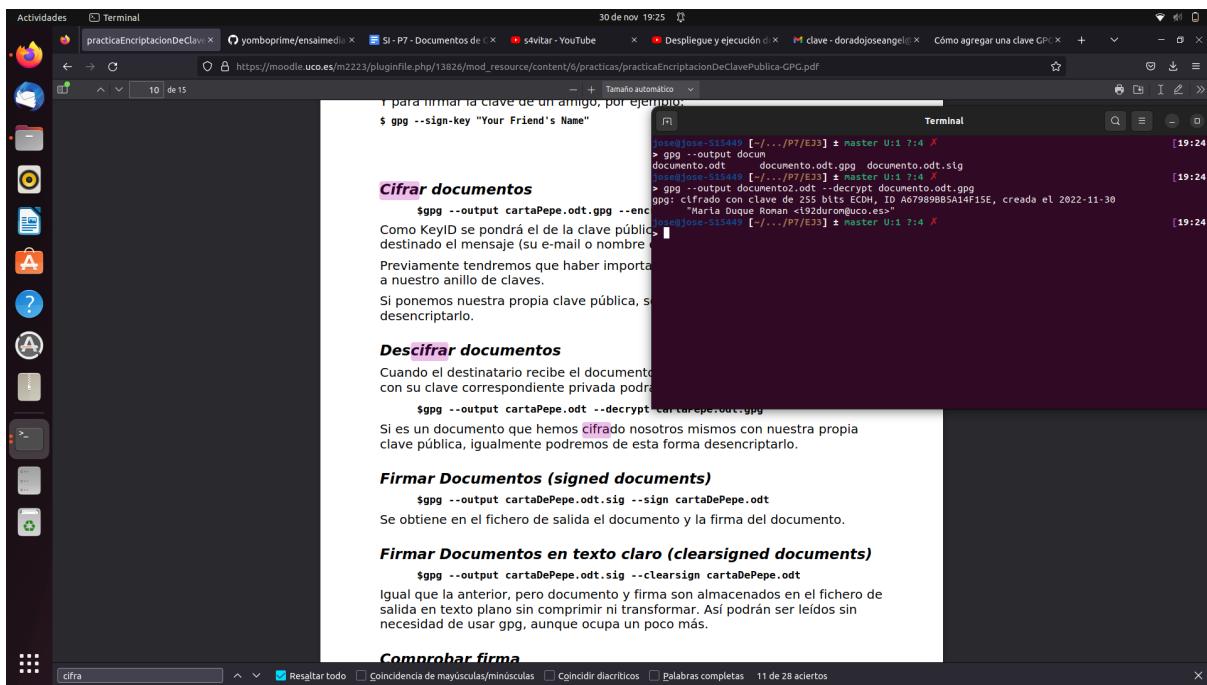
```
Terminal
inse@Inse-S15449 [-/.../PT/E3] ± master U:1 7:4 X [19:18:52]
> gpg --output documento.odt.sig --sign documento.odt
inse@Inse-S15449 [-/.../PT/E3] ± master U:1 7:4 X [19:19:59]
```

Para encriptar un documento mediante la clave de nuestro compañero utilizaremos el comando ***gpg –output documento.odt.gpg –encrypt –recipient correcoElectrónico documento.odt***:



```
30 de nov 19:23 Terminal
> gpg --output documento.odt.gpg --encrypt --recipient l92durom@uco.es documento.odt
[...]
```

Si queremos descifrar este documento utilizamos el comando ***gpg –output documento2.odt –decrypt documento.odt.gpg*** y ponemos la contraseña de la clave de nuestro compañero:



```
30 de nov 19:25 Terminal
$ gpg --sign-key "Your Friend's Name"
$ gpg --output cartaPepe.odt.gpg --encrypt --recipient l92durom@uco.es documento.odt
$ gpg --output documento2.odt --decrypt documento.odt.gpg
```

Cifrar documentos
Como KeyID se pondrá el de la clave pública destinado el mensaje (su e-mail o nombre). Previamente tendremos que haber importado nuestro anillo de claves. Si ponemos nuestra propia clave pública, se desencriptará.

Descifrar documentos
Cuando el destinatario recibe el documento con su clave correspondiente privada podrá:

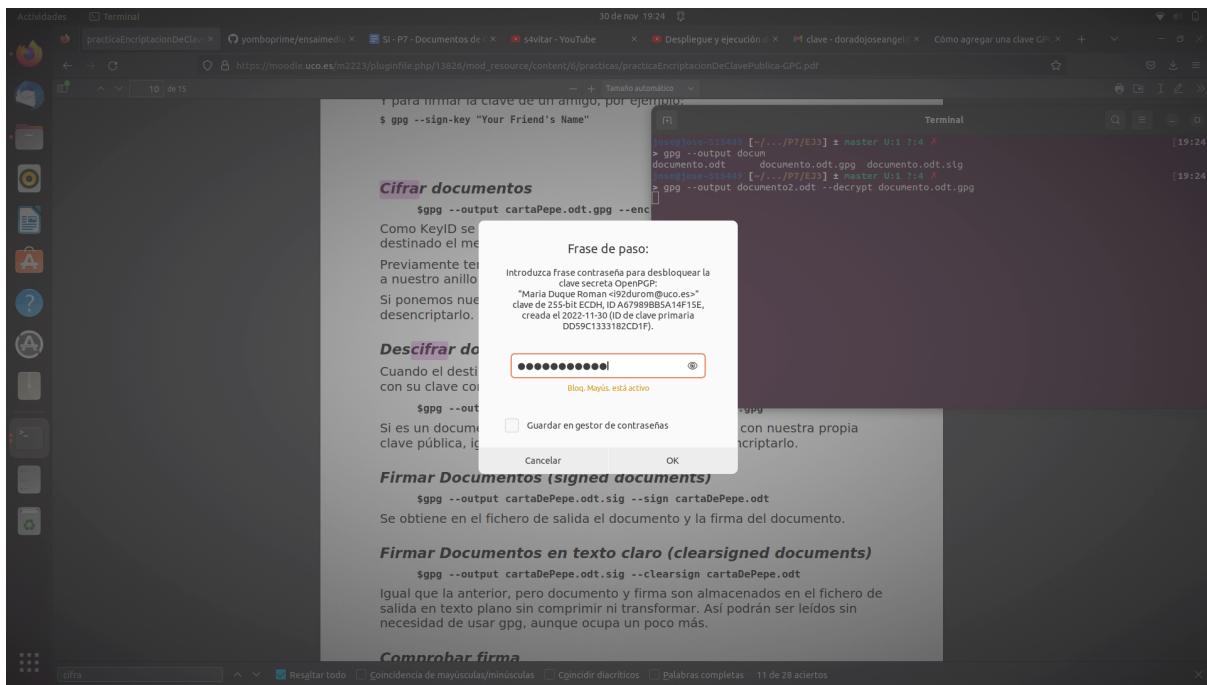
```
$ gpg --output cartaPepe.odt --decrypt cartaPepe.odt.gpg
```

Si es un documento que hemos cifrado nosotros mismos con nuestra propia clave pública, igualmente podremos de esta forma desencriptarlo.

Firmar Documentos (signed documents)
\$ gpg --output cartaDePepe.odt.sig --sign cartaDePepe.odt
Se obtiene en el fichero de salida el documento y la firma del documento.

Firmar Documentos en texto claro (clearsigned documents)
\$ gpg --output cartaDePepe.odt.sig --clearsign cartaDePepe.odt
Igual que la anterior, pero documento y firma son almacenados en el fichero de salida en texto plano sin comprimir ni transformar. Así podrán ser leídos sin necesidad de usar gpg, aunque ocupa un poco más.

Comprobar firma



5º) Exportar la clave creada y hacerla llegar a algunos de los compañeros de clase. Importar algunas claves públicas de otros usuarios y realizar cada uno de los ejercicios que se han visto en la sesión de prácticas: firma, cifrado, firma+cifrado, etc.

Para comprobar el fingerprint de nuestro compañero utilizaremos el comando **gpg –fingerprint correoElectrónico:**

```
jose@jose-S15449:~/.PT/E33] ~ master U:1 7:4 X
> gpg --fingerprint i92duron@uco.es
pub ed25519 2022-11-30 [SC] [caduc: 2024-11-29]
    0DE6 CDCC D305 5714 1075 A3C4 DD59 C133 3182 CD1F
uid          [ total ] María Duque Roman <i92duron@uco.es>
sub cv25519 2022-11-30 [E] [caduc: 2024-11-29]

[jose@jose-S15449:~/.PT/E33] ~ master U:1 7:4 X
>
```

Para firmar la clave utilizaremos el comando **gpg –sign-key correoElectrónico**, en mi caso como la firmé con anterioridad no se puede hacer otra vez.

```
jose@jose-S15449:~/.PT/E33] ~ master U:1 7:4 X
> gpg -s-sign-key i92duron@uco.es

sec ed25519 /0D59C1333182CD1F
    creado: 2022-11-30  caduc: 2024-11-29  uso: SC
    confianza: desconocido  validez: total
ssb cv25519 /A67989B85A14F15E
    creado: 2022-11-30  caduc: 2024-11-29  uso: E
    [ total ] (1). María Duque Roman <i92duron@uco.es>

"María Duque Roman <i92duron@uco.es>" ya estaba firmada por la clave C838903278215A87
Nada que firmar con la clave C838903278215A87

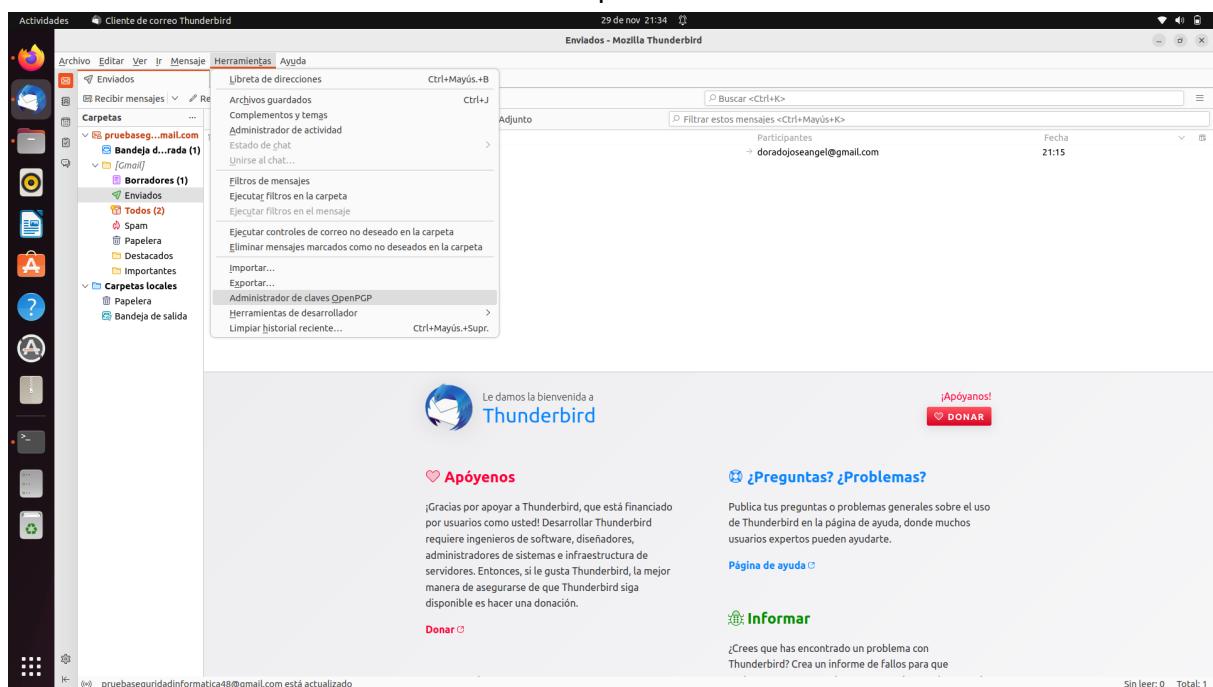
Clave sin cambios, no se necesita actualización.
>
```

6ºComo ejercicio adicional opcional, buscar la integración de GPG en vuestro cliente de correo electrónico favorito. Como por ejemplo Enigmail de Thunderbird.

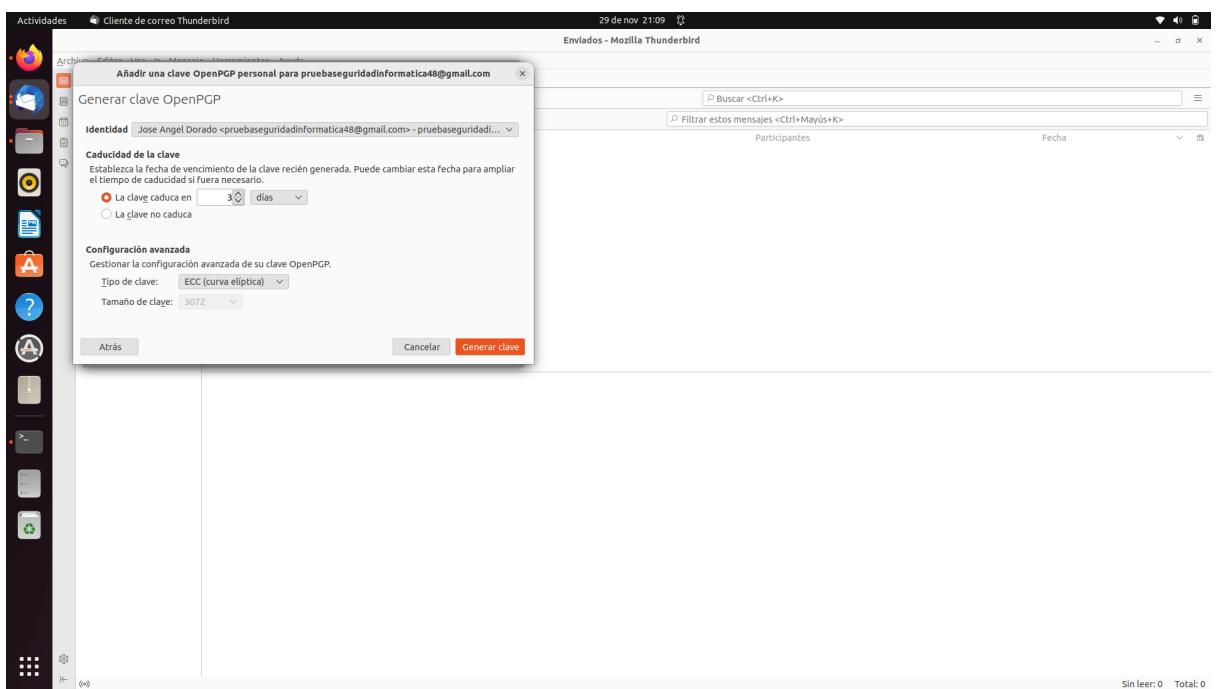
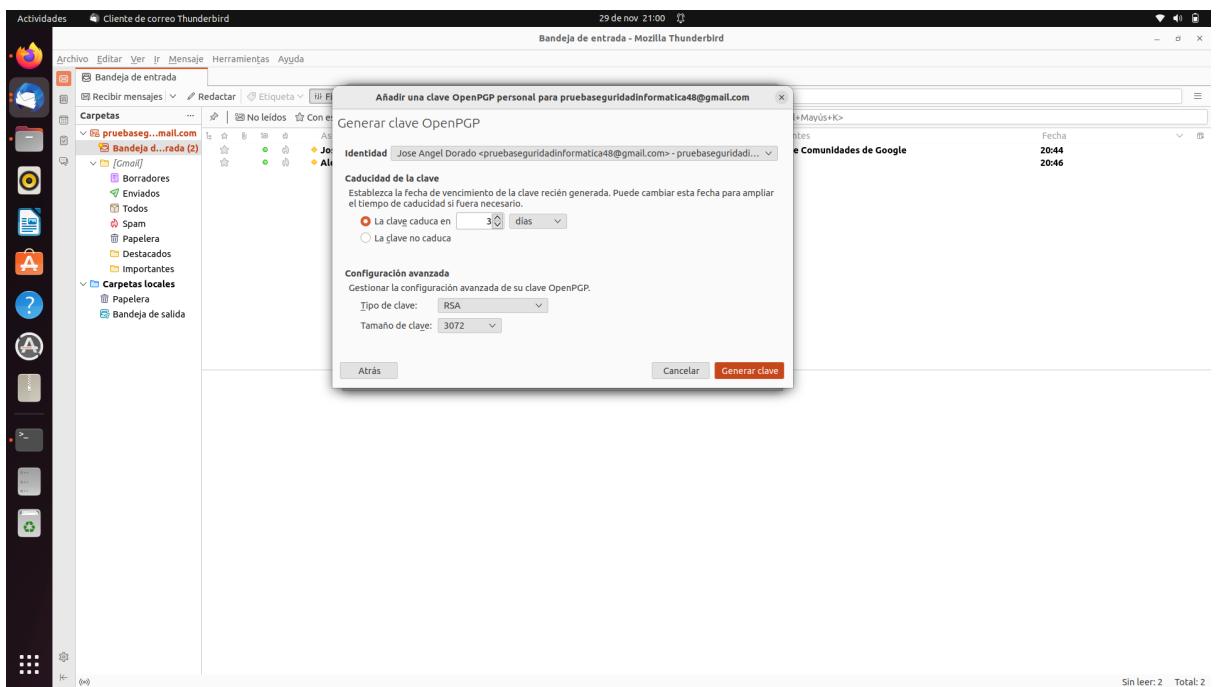
En mi caso utilizaré Thunderbird, el cliente de correo electrónico que viene con Ubuntu. Thunderbird permite integrar la encriptación de GPG para el envío de mensajes, en el caso de las versiones antiguas se puede utilizar Enigmail, que es un complemento de este cliente de correo electrónico. Sin embargo, para las versiones más recientes se ha reemplazado Enigmail por OpenPGP, que ya viene instalado previamente.

Para una mejor explicación pondré varias capturas de pantalla utilizando OpenPGP ya que tengo la última actualización. A continuación se exponen los pasos a seguir para configurar el par de claves:

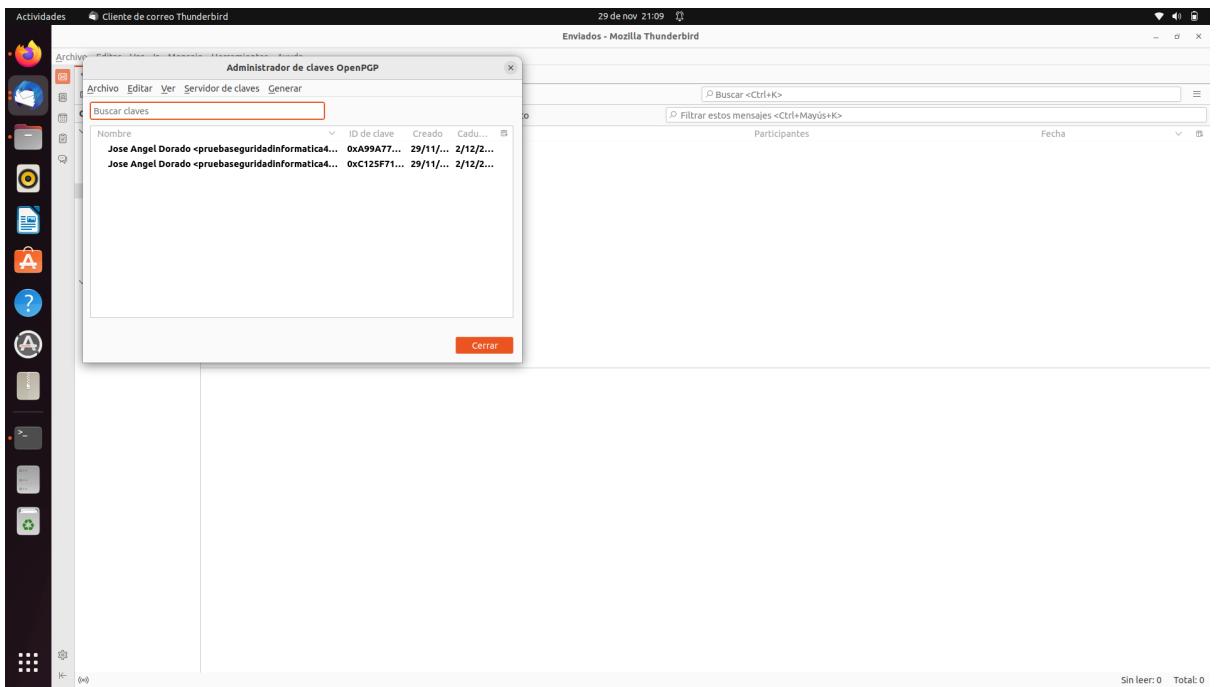
1. En primer lugar, una vez abierto Thunderbird, iremos a **Herramientas->Administrador** de claves OpenPGP.



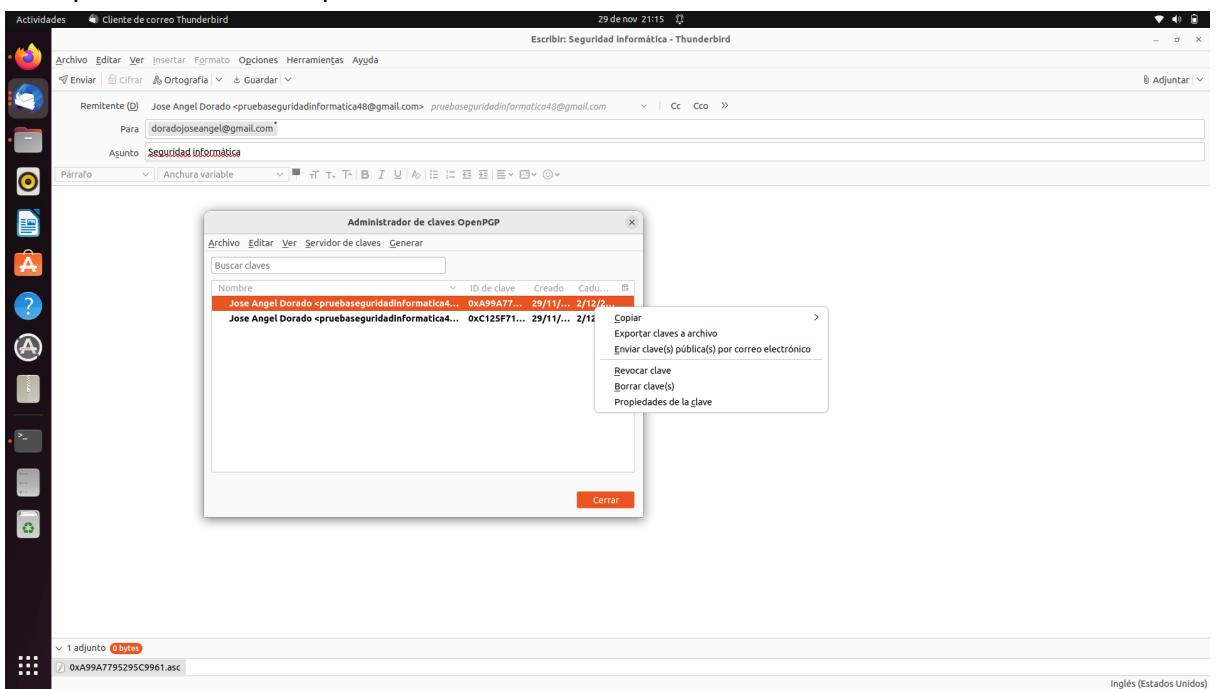
2. Ahora nos aparecerá una ventana, para generar las claves iremos a **Generar->Nuevo par de claves**. Una vez hayamos llegado aquí podemos elegir el tipo de clave, RSA o ECC, además de poder seleccionar el tiempo de caducidad de nuestra clave.



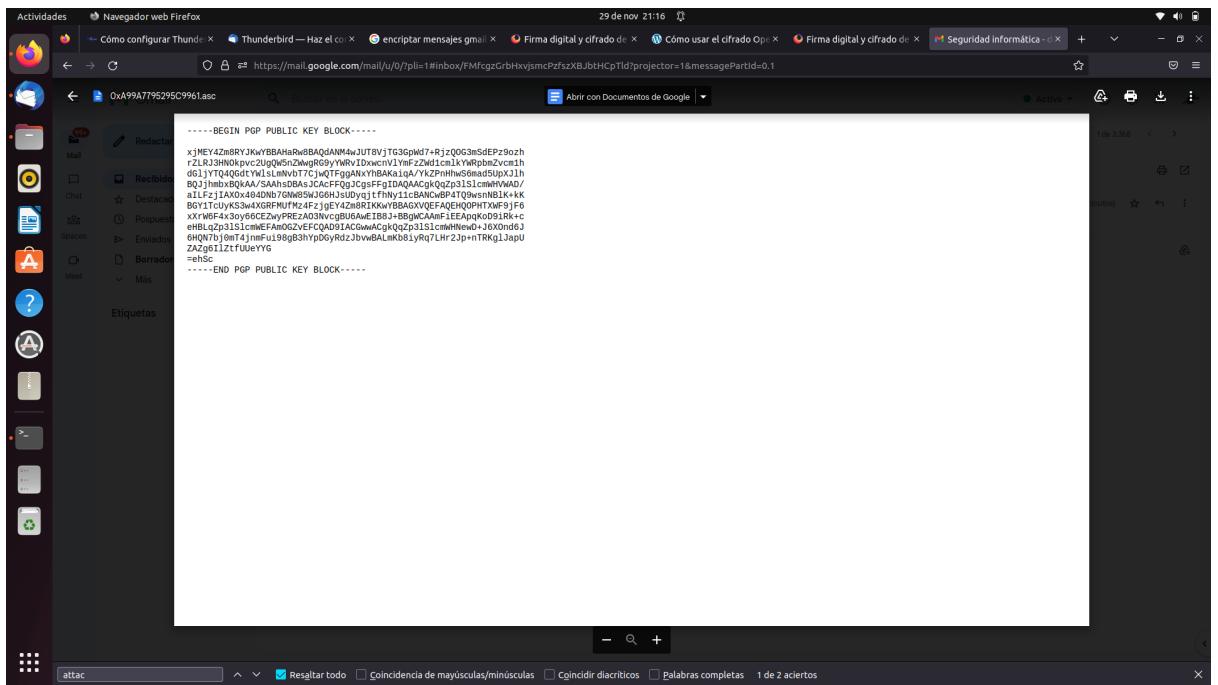
3. Generadas las claves ya nos saldrán en la ventana de OpenPGP.



- Para poder enviar mensajes entre dos usuarios ambos tienen que tener dichas claves, para ello podemos exportar nuestras claves, si presionamos click derecho nos aparecerán dichas opciones.



- Ahora enviaré un mensaje a un correo que tengo creado para ver cómo llegaría la clave que compartimos. El resultado es el siguiente:



Estos serían todos los pasos para configurar OpenPGP, en mi caso solo he realizado la parte del emisor aunque para la correcta encriptación de los mensajes, ambos usuarios se tienen que compartir sus respectivas claves, como mencioné anteriormente.

Bibliografía

<https://support.mozilla.org/es/kb/firma-digital-y-cifrado-de-mensajes>

<https://www.redeszone.net/tutoriales/seuridad/configurar-openpgp-thunderbird-emails-cifrados/>