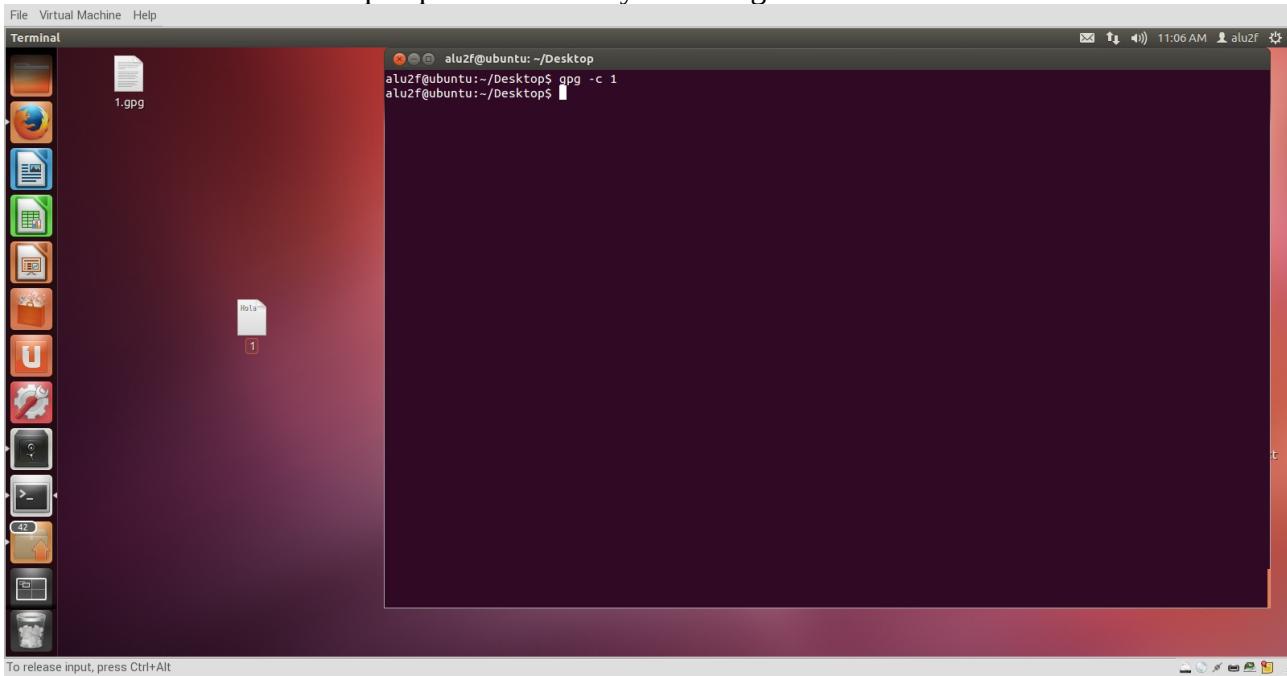


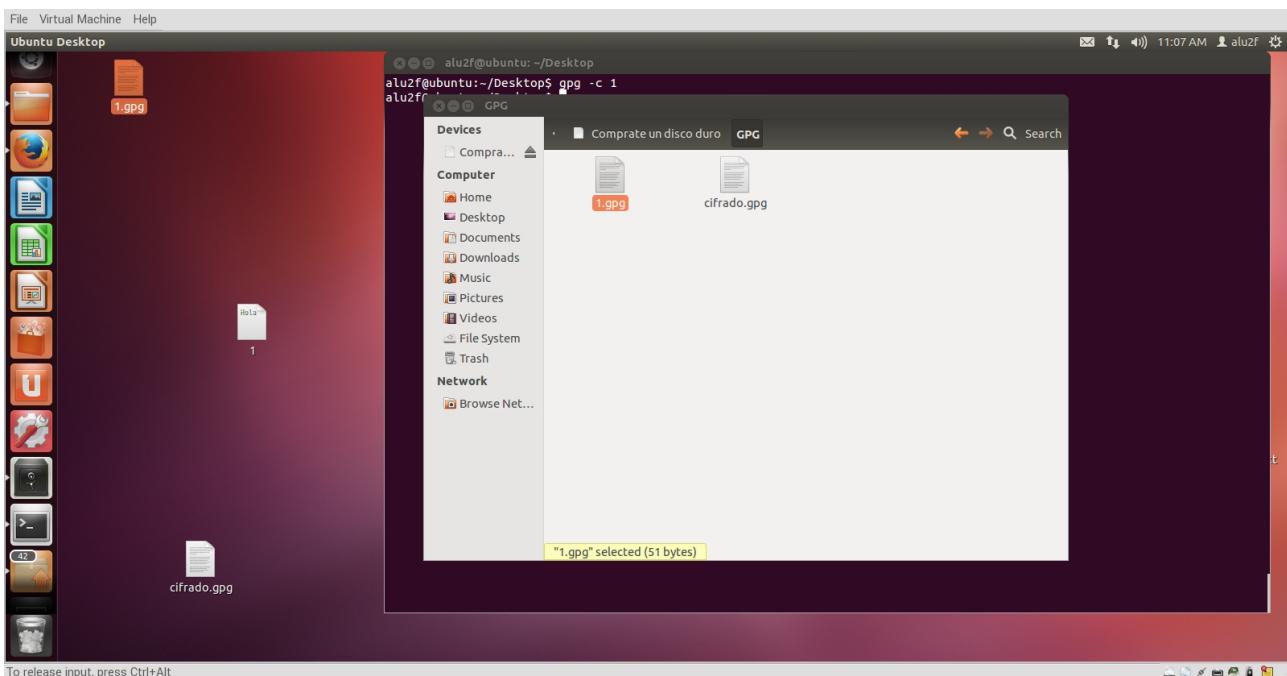
Práctica Criptografía

Ejercicio1

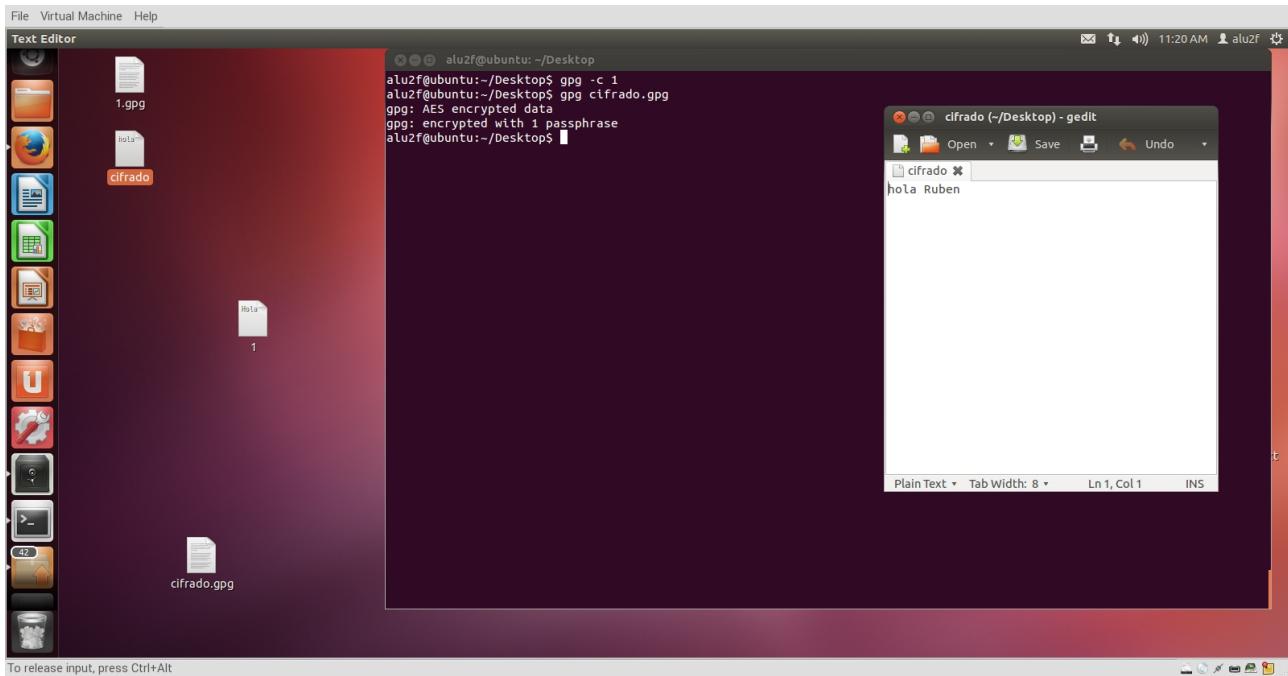
Creamos el archivo que queremos cifrar y con el siguiente comando lo ciframos.



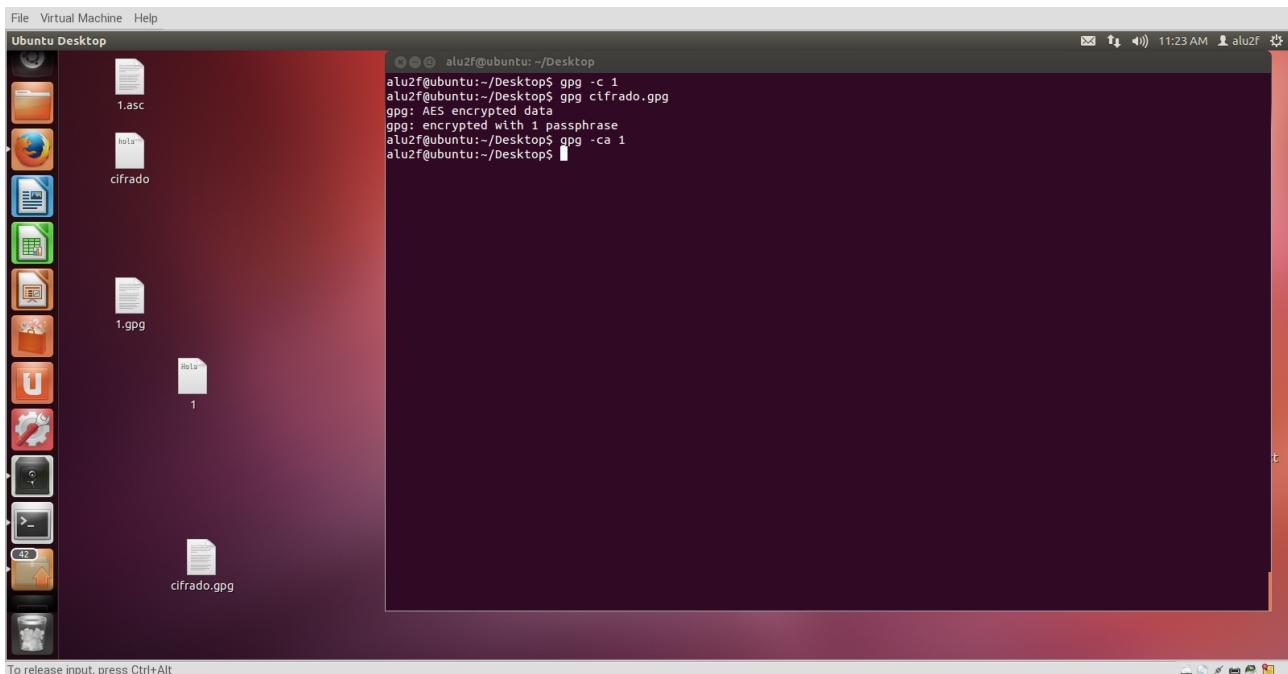
Nos intercambiamos las prácticas con nuestro compañero.



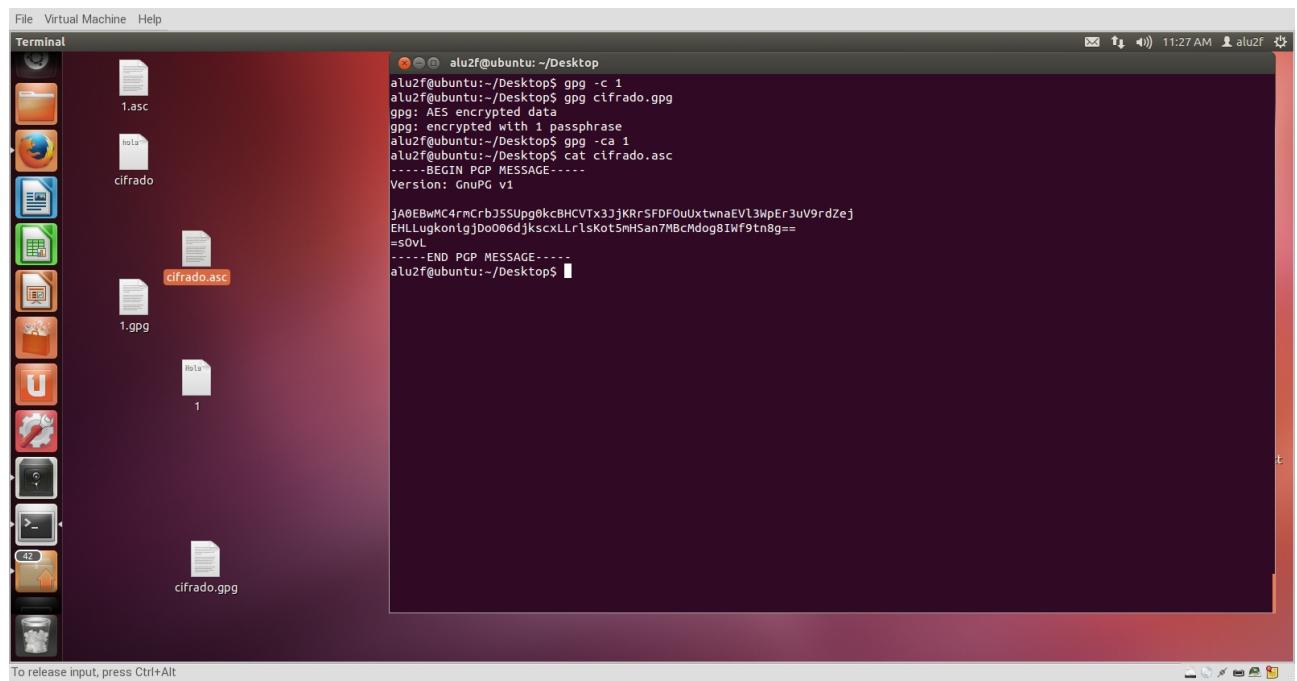
Desciframos el archivo que nos ha pasado el compañero con el siguiente comando. Tras descifrarlo nos aparecerá el contenido.



Con el siguiente comando, añadiendo la opcion “a”, ciframos el archivo de una forma mas segura.

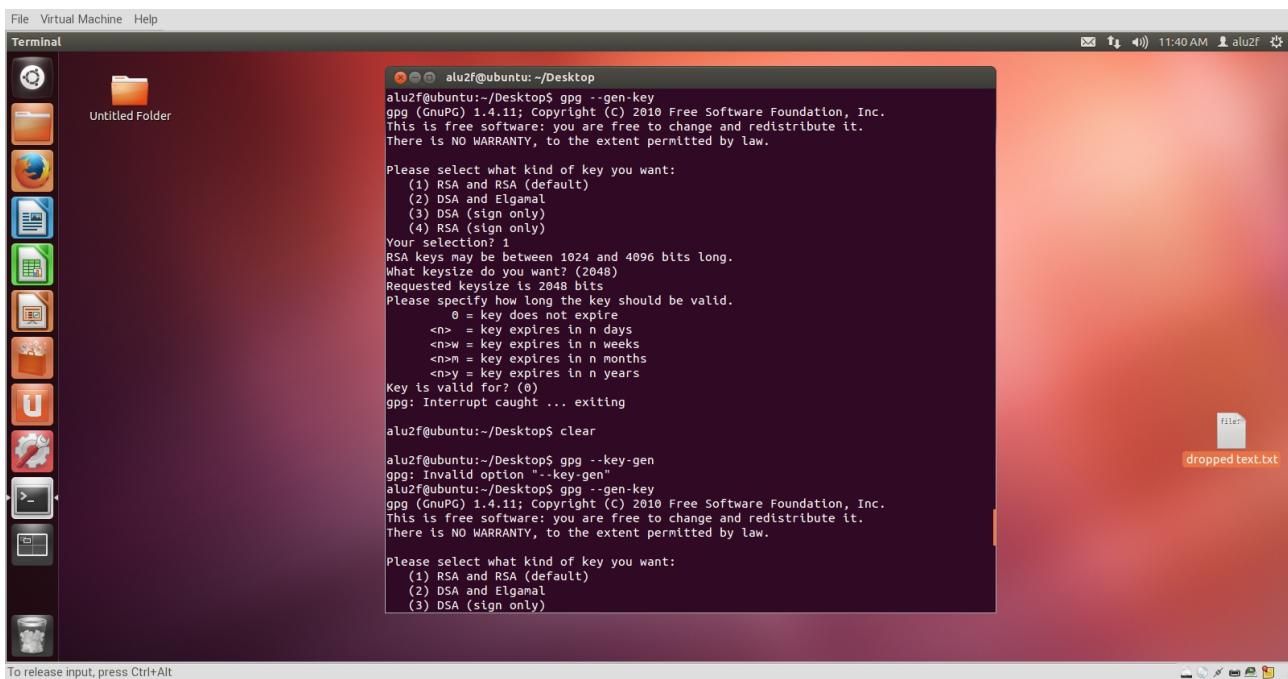


Por ultimo comprobamos que el contenido final de el archivo está cifrado y no podemos acceder a el.



Ejercicio 2

Generamos una clave pública.



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal" and the prompt is "aluzf@ubuntu:~/Desktop". The user is generating a GPG key:

```
aluzf@ubuntu:~/Desktop$ gpg --gen-key
gpg (GnuPG) 1.4.11; Copyright (C) 2010 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.

Please select what kind of key you want:
 (1) RSA and RSA (default)
 (2) DSA and Elgamal
 (3) DSA (sign only)
 (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
 0 = key does not expire
 <n> = key expires in n days
 <n>w = key expires in n weeks
 <n>m = key expires in n months
 <n>y = key expires in n years
Key is valid for? (0)
gpg: Interrupt caught ... exiting

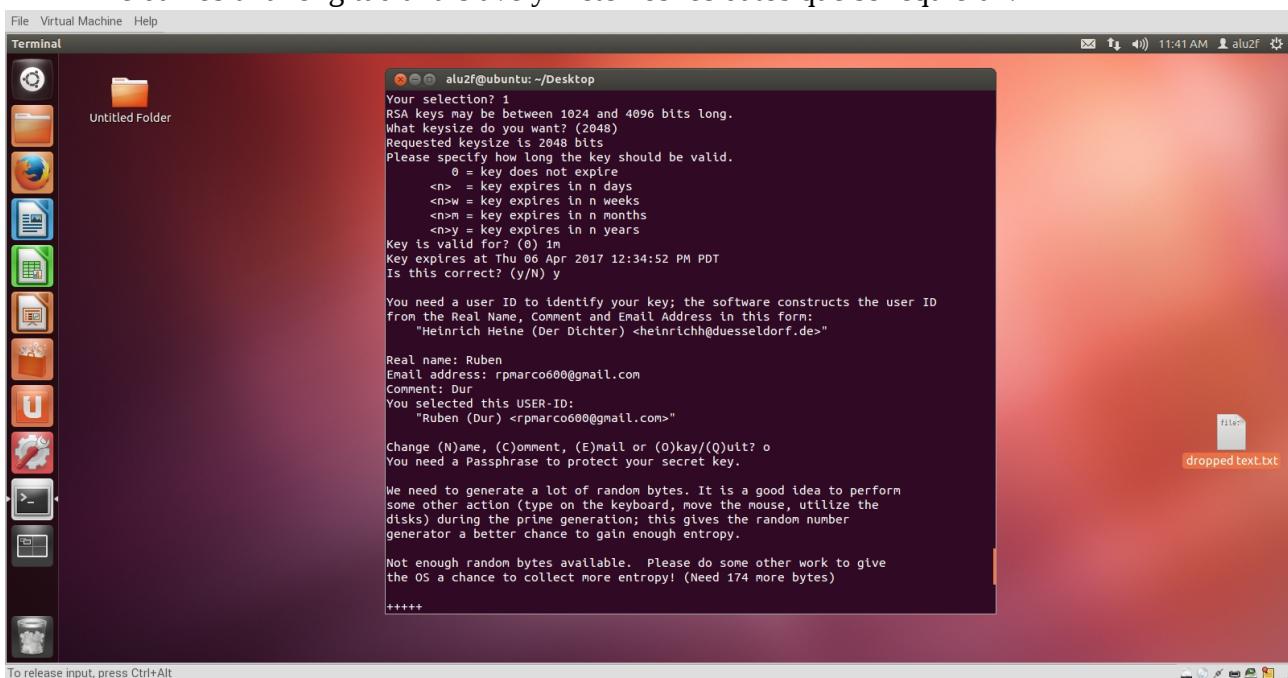
aluzf@ubuntu:~/Desktop$ clear

aluzf@ubuntu:~/Desktop$ gpg --key-gen
gpg: Invalid option "--key-gen"
aluzf@ubuntu:~/Desktop$ gpg --gen-key
gpg (GnuPG) 1.4.11; Copyright (C) 2010 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.

Please select what kind of key you want:
 (1) RSA and RSA (default)
 (2) DSA and Elgamal
 (3) DSA (sign only)
```

A file named "dropped text.txt" is visible on the desktop.

Le damos una longitud a la clave y metemos los datos que se requieran.



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal" and the prompt is "aluzf@ubuntu:~/Desktop". The user is generating a GPG key and providing user information:

```
aluzf@ubuntu:~/Desktop$ Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
 0 = key does not expire
 <n> = key expires in n days
 <n>w = key expires in n weeks
 <n>m = key expires in n months
 <n>y = key expires in n years
Key is valid for? (0) 1m
Key expires at Thu 06 Apr 2017 12:34:52 PM PDT
Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
  "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Ruben
Email address: rpmarco600@gmail.com
Comment: Dur
You selected this USER-ID:
  "Ruben (Dur) <rpmarco600@gmail.com>"

Change (N)ame, (C)omment, (E)m ail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.

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.

Not enough random bytes available. Please do some other work to give
the OS a chance to collect more entropy! (Need 174 more bytes)

+++++
```

A file named "dropped text.txt" is visible on the desktop.

```

File Virtual Machine Help
Terminal alu2f@ubuntu: ~/Desktop
[...]
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.

Not enough random bytes available. Please do some other work to give
the OS a chance to collect more entropy! (Need 174 more bytes)

+++++
....+++++
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.

Not enough random bytes available. Please do some other work to give
the OS a chance to collect more entropy! (Need 68 more bytes)

+++++
Not enough random bytes available. Please do some other work to give
the OS a chance to collect more entropy! (Need 11 more bytes)
.+++++
gpg: /home/alu2f/.gnupg/trustdb.gpg: trustdb created
gpg: key 8BDD2A4A marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed. PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2017-04-06
pub 2048R/8BDD2A4A 2017-03-07 [expires: 2017-04-06]
    Key fingerprint = 6BFE 9A26 320E 3394 A796 D18D 1A97 576B 8BDD 2A4A
uid             Ruben (Dur) <rpmarco600@gmail.com>
sub 2048R/934AB941 2017-03-07 [expires: 2017-04-06]

alu2f@ubuntu:~/Desktop$ gpg -a --export Ruben
alu2f@ubuntu:~/Desktop$ 

```

To release input, press Ctrl+Alt

El siguiente paso es exportar la clave que hemos generado y nos saldra en la pantalla del terminal.

```

File Virtual Machine Help
Terminal alu2f@ubuntu: ~/Desktop
[...]
uid                  Ruben (Dur) <rpmarco600@gmail.com>
sub 2048R/934AB941 2017-03-07 [expires: 2017-04-06]

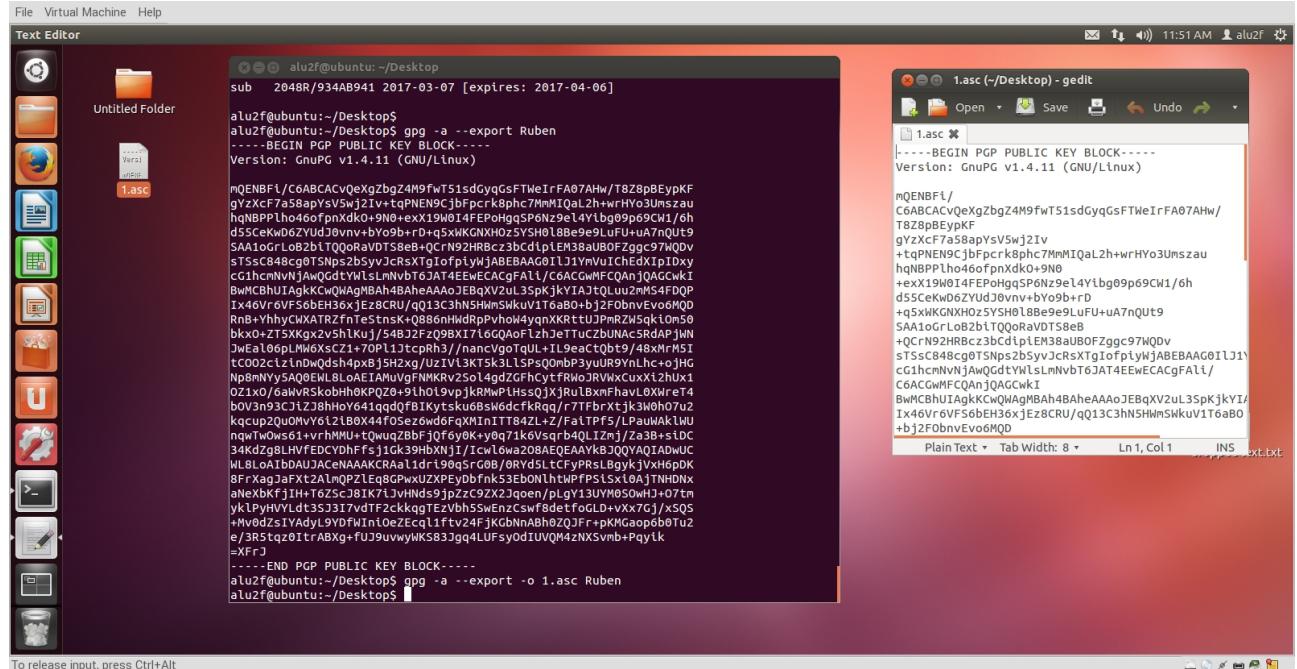
alu2f@ubuntu:~/Desktop$ gpg -a --export Ruben
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.11 (GNU/Linux)

mQENBFi/C6ABCACvqeXgzbgbz4M9FwT51sdGyqGsfTWelrFA07AHw/T8Z8pBEypKF
gYzXcf7a5apvys5wJzIv+tqPNE9CjbFpcrk8phc7MMiM0aL2h-wrHvo3UnszaU
hgNPBPPlho46ofpnDxk0+SN0+exx19W014FEPoHggSP6Nz9e14Ylbg9p69Cw1/6h
d5scKwD6ZYUdJ0vnv+bY9s+b-rb+q5xKGXHOzSYSh0lBe9e9LuFU+uA7nQUt9
SA1oGrLoBzb1TQoRaVDTs8e+oCrn2HRbcz3CdciplE38auLB0fZggc97QDv
sTscB48cg9TSNs2bsvvJcrsxtqIoifplywJABEBAAAG0I1j1YvNUtChEdXpiDxy
cg1hcnNWnJaW0GdtvWlsLnvbT6jAT4EwEcACgFA1l/C6ACGwMFCAQnjQAACGwKI
BwMCBHUtaqKcw0WAgAAoJEBqXV2uL3SpkJKYTA1tQluu2MS4FDQP
IX46Vr6VF56beH36xjezbCRU/q013C3hNSHm5kuvu176aB0+j2j2ObnvEv6MQ
Rnb-YhhcWxATRzfTeStnsK-0886nHdRpPhwo4yqnXKRttUJpmRzsWqklon58
blkox+ZTSKgx2vsh1Kuj/54B2fZez09BX1716QoFlzheTTuZBuNaCsRdAPjWn
JwEa166plMW6xsCz1+TOPl1JtcPrh3//naaVgoTqlu+I19eactqb1q/48xMrMSI
tc002cizinhdshdshapxB15h2xg/lzTz3Ktsk31LSPS0mb3vuLR9Ynlhc+oJHG
Np8mNy5aQ9EmL8LoAeIAmuVgFNMKRv2So14gdZGFhCytFRwJRVwxCuxX12hUx1
021x0/6awVRSkobHh0kPOZ9+9t0h19vpjkRMwPihssQjXjU18xfhavL0XWrer4
b0V3n93C1zJ9hHo641qq0FBtkytsku6BsW6dcfkRqg/-77fbryXj3kWhoh7u2
kocup2Qu0MvY612Lb0x44f0Sezowd6rqXMInIT842L-Z/FalTPf5/LPau4k1wU
nowTwos61+vrHMU+10wuaZbfj0fSy0k+y0q71keVsqr4QlL2m/1z3B+s1D
34KdZg9LHFEDCYDHf1516k39hBNX1j/lcmL6wa208AEQAAVkb3QQYAO1AdwUc
Ml8loA1boAUJAcaNAAKCRa1ld19q65rG0B/0Yd5LcFyPRslBgykJxH6pDk
8fXadgjaFxt2AlmPQ2Le08GPwxJ2XPeybfnk53eb0NhtwPfPSLxt1dAjTNHDN
aleXbfkjH+T6ZSc81K7iJvNdsjpbzC92Xzqoen/plgy13UYMOSwHJ+07tm
yk1PyhVYldt35317vdtF2ckkgqIEzbh5SwEnzcswfbdetf0GLd+Vxx7Gj/xS05
+mv0dzsIVAdy19YDFWn10eZeqcliftv24fJKGbnnAbh0ZQJF+rPKGaoPb0bTu2
e/3R5tqzo1trABxg+fuJ9uwyWKS33jq4LUfsyod1UVQM4zNXSvmb+Pqylk
=xFJ
-----END PGP PUBLIC KEY BLOCK-----
alu2f@ubuntu:~/Desktop$ 

```

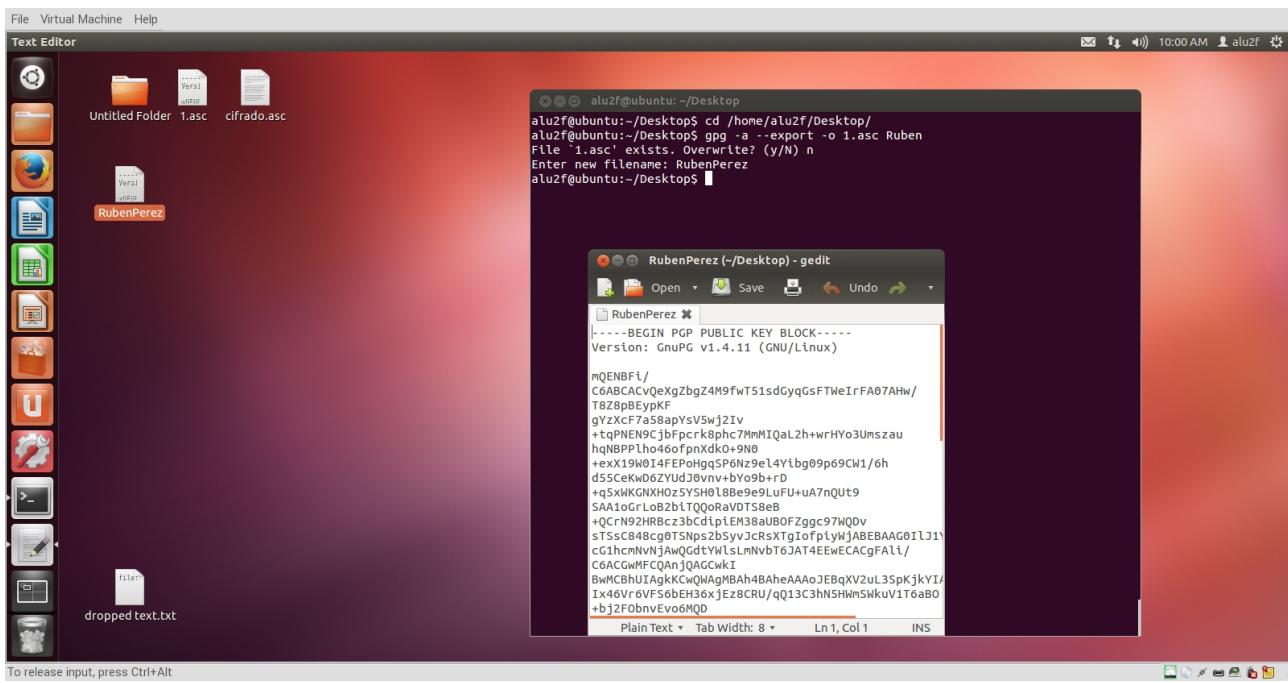
To release input, press Ctrl+Alt

Comprobamos que al abrir el archivo nos sale la información cifrada y que no podemos leerla.



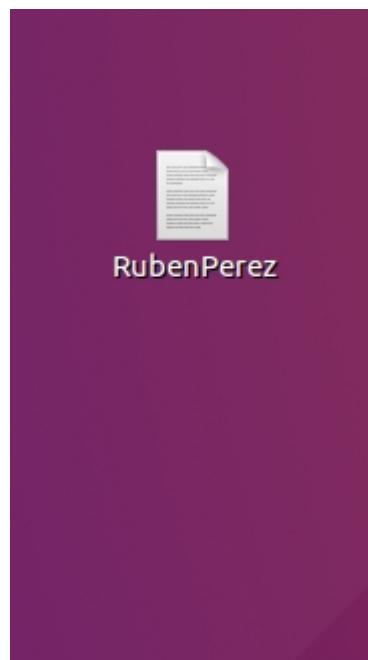
Ejercicio 3

Exportamos el archivo 1.asc y le cambiamos el nombre.



```
alu2f@ubuntu:~/Desktop$ gpg -a --export -o 1.asc Ruben
```

Este es el archivo con el nuevo nombre.



Importamos el archivo en el ordenador de nuestro compañero

```
alu2f@ubuntu:~/Desktop$ sudo gpg --import 1.asc
[sudo] password for alu2f:
gpg: /home/alu2f/.gnupg/trustdb.gpg: trustdb created
gpg: key 8BDD2A4A: public key "Ruben (Dur) <rpmarco600@gmail.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1  (RSA: 1)
```

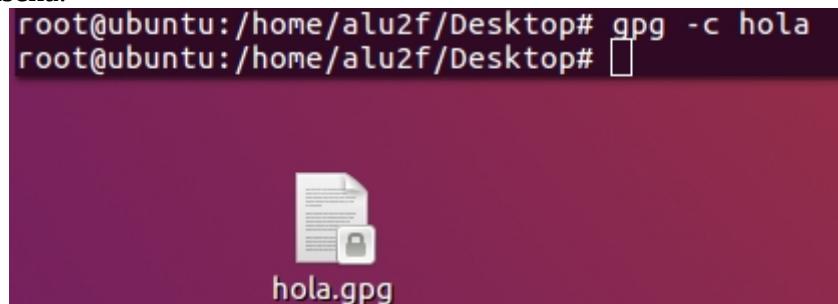
Con el siguiente comando comprobamos que el archivo se ha importado con éxito.

```
root@ubuntu:/home/alu2f/Desktop# gpg -kv
/root/.gnupg/pubring.gpg
-----
pub    2048R/8BDD2A4A 2017-03-07 [expires: 2017-04-06]
uid          Ruben (Dur) <rpmarco600@gmail.com>
sub    2048R/934AB941 2017-03-07 [expires: 2017-04-06]
```

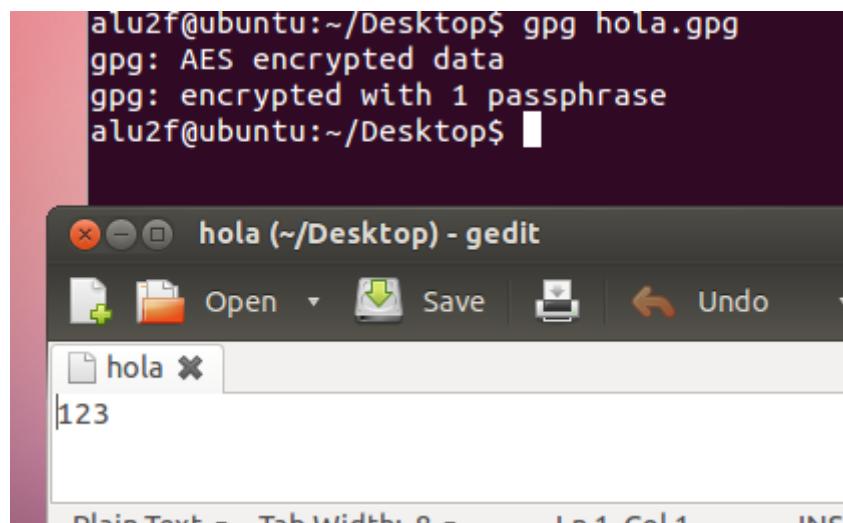
Ejercicio 4

Creamos un archivo y con el siguiente comando lo ciframos, para que solo podamos acceder a el con la contraseña.

```
root@ubuntu:/home/alu2f/Desktop# gpg -c hola  
root@ubuntu:/home/alu2f/Desktop#
```

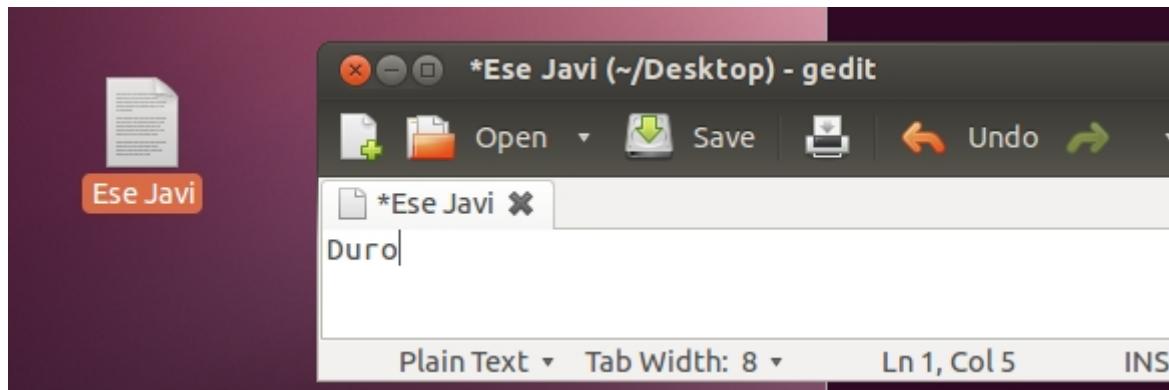


Lo mandamos al otro usuario donde el ya tenia la clave publica por los ejercicios anteriores, lo abrimos y comprobamos que muestra su contenido. Si lo hicésemos en un ordenador que no tenga la clave no nos mostrara el contenido.



Ejercicio 5

Creamos un documento con algo escrito dentro, para posteriormente modificarlo.



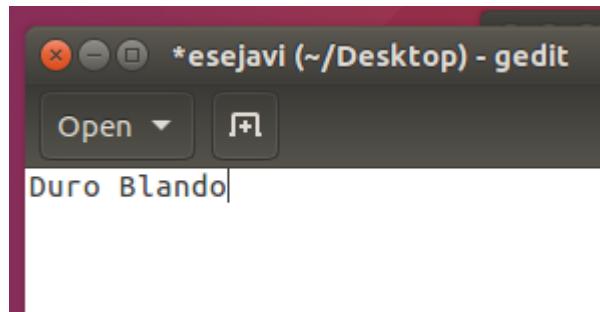
Creamos una clave para el documento.

```
alu2f@ubuntu:~/Desktop$ gpg -sb -a esejavি
You need a passphrase to unlock the secret key for
user: "Ruben (Dur) <rpmarco600@gmail.com>"
2048-bit RSA key, ID 8BDD2A4A, created 2017-03-07
alu2f@ubuntu:~/Desktop$
```

Ejecutamos el siguiente comando para desencriptar el documento anterior.

```
root@ubuntu:/home/alu2f/Desktop# gpg --decrypt -o esejavি esejavি.asc
gpg: assuming signed data in `esejavি'
gpg: Signature made Mon 13 Mar 2017 08:45:52 AM PDT using RSA key ID 8BDD2A4A
gpg: Good signature from "Ruben (Dur) <rpmarco600@gmail.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                 There is no indication that the signature belongs to the owner.
Primary key fingerprint: 6BFE 9A26 320E 3394 A796  D10D 1A97 576B 8BDD 2A4A
root@ubuntu:/home/alu2f/Desktop#
```

Comprobamos que podemos modificar el archivo cuando introducimos la contraseña que hemos añadido anteriormente.



Con el siguiente comando verificamos si hemos modificado el archivo y comprobamos que si no ponemos la contraseña no nos dejara modificar.

```
root@ubuntu:/home/alu2f/Desktop# gpg --decrypt -o esejavi esejavi.asc
gpg: assuming signed data in `esejavi'
gpg: Signature made Mon 13 Mar 2017 08:45:52 AM PDT using RSA key ID 8BDD2A4A
gpg: BAD signature from "Ruben (Dur) <rpmarco600@gmail.com>"
```