Tarea 1: Crea una clave SSH en tu sistema operativo y aporta tu clave pública, es decir, .pub

Para generar una clave SSH introducimos el siguiente código seguido de nuestro correo electrónico, nos preguntará donde queremos guardar la clave, también podremos cambiarle el nombre al archivo, en mi caso lo he dejado por defecto.

```
brus@brus-VirtualBox:~$ ssh-keygen -t ed25519 -C "brunobm98@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/brus/.ssh/id_ed25519):
Created directory '/home/brus/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/brus/.ssh/id_ed25519
Your public key has been saved in /home/brus/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:9arP8seq1ofJr1ARpjwp7XUjaH3tEER5LeD91EcYKQs brunobm98@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|          oo=o.=.|
|       o *Eoo+=.o|
|      . O *.*+o.+|
|       + + *.+ o.|
|        S . . . .|
|         . .     |
|        .o.+     |
|        o+= +    |
|       .o=**.    |
+----[SHA256]-----+
brus@brus-VirtualBox:~$
```

En el directorio que nos ha creado vemos que nos ha generado la clave privada y la pública (.pub).

```
brus@brus-VirtualBox:~$ cd /home/brus/.ssh/
brus@brus-VirtualBox:~/.ssh$ ls
id_ed25519  id_ed25519.pub
brus@brus-VirtualBox:~/.ssh$
```

```
brus@brus-VirtualBox:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEm+nJDzkD3y27z3F9aMR0g38V0AxIe2jj4GN0afb6e7
 brunobm98@gmail.com
brus@brus-VirtualBox:~/.ssh$ cat id_ed25519
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACBJvpyQ85A98tu89xfWjEdIN/FdAMSHto4+BjdGn2+nuwAAAJgZ7aRmGe2k
```

Por último agregamos la clave privada al ssh-agent

```
-----END OPENSSH PRIVATE KEY-----
brus@brus-VirtualBox:~/.ssh$ ssh-add ~/.ssh/id_ed25519
Identity added: /home/brus/.ssh/id_ed25519 (brunobm98@gmail.com)
brus@brus-VirtualBox:~/.ssh$
```

Tarea 2: Crea un programa con el nombre “tarea2.py” en Python que muestre por pantalla el porcentaje de espacio ocupado en cada una de las particiones de tu sistema, de forma que se muestre tal que así: /dev/sda1 78,9% /dev/sdb1 18,5%

Para crear el siguiente programa necesitaremos utilizar la librería psutil, por lo que la tendremos que instalarla previamente, en mi caso también tuve que instalar el paquete *python3-pip* que es la principal herramienta para instalar y administrar paquetes de python.

```
brus@brus-VirtualBox:/etc$ pip install psutil
No se ha encontrado la orden «pip», pero se puede instalar con:
sudo apt install python3-pip
brus@brus-VirtualBox:/etc$ sudo apt install python3-pip
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
```

En el ejemplo que subiste al aula me daba un error de sintaxis en la lista *Salida* asi que lo he hecho utilizando variables y print.
En resumen el código consta de un bucle *for* que selecciona cada partición y de cada una se imprime el directorio y el porcentaje de uso.

Tarea2.py > ...

```python
import psutil

# Obtener información sobre las particiones del disco y le damos el valor a la
#variable particiones
particiones = psutil.disk_partitions()

# Para mostrar la informacion de cada particion utilizanos un ciclo for que introduce
# en la variable 'particion' cada cada una de estas
print("Particiones del disco:")
for particion in particiones:
    # Obtenemos la informacion del uso del disco con la funcion disk_usage y le damos
    # el valor a la variable espacio
    espacio = psutil.disk_usage(particion.mountpoint)
    #Despues imprimimos el directorio de cada particion con 'particion.device'
    # y el porcentaje de uso con 'espacio.percent
    print(f"{particion.device} {espacio.percent:.1f}%")


```

```
brus@brus-VirtualBox:~/Documentos/test$ python3 Tarea2.py
Particiones del disco:
/dev/sda3 69.2%
/dev/loop0 100.0%
/dev/loop1 100.0%
/dev/loop2 100.0%
/dev/loop3 100.0%
/dev/loop4 100.0%
/dev/loop5 100.0%
/dev/loop6 100.0%
/dev/loop7 100.0%
/dev/loop9 100.0%
/dev/loop8 100.0%
/dev/loop10 100.0%
/dev/sda3 69.2%
```

Tarea 3: Implementa un programa en el fichero llamado “tarea3.py” que ejecute un bucle 5 veces donde creará una carpeta con el nombre folder1, folder2 ...folder5, reando dentro de ellos 10 ficheros con el siguiente nombre y siguiente contenido: nombre fichero: fichero1.txt contenido: Este es el contenido del fichero 1

Tarea3.py > ...

```python
import os



# Creamos el bucle para las carpetas
for i in range(5):
    # Crear la variable nombre carpeta y le damos el valor folder mas el numero de cada iteración
    nom_carpeta = f"folder{i}"

    # Para crear la carpeta usamos os.makedirs que crea el directorio
    # Agregando exists_ok=True le indicamos que si algunos d elos directorios ya existe
    # no se genere un error si no que continue
    os.makedirs(nom_carpeta, exist_ok=True)

    # Bucle para crear los ficheros dentro de la carpeta
    for n in range(10):
        # Creamos la variable que contiene el nombre de los ficheros
        nom_archivo = f"fichero{n}.txt"

        # Creamos la variable con el contenido del archivo
        texto = f"Este es el contenido del fichero {n}"

        # os.path.join recoge el nombre de la carpeta y del archivo par acrear el directorio
        # w es de write, es decir se abre cada archivo con permiso de escritura
        # por ultimo file.write() escribe el texto dentro del archivo
        with open(os.path.join(nom_carpeta, nom_archivo), "w") as file:
            file.write(texto)

print("Archivos creados")
```

```
brus@brus-VirtualBox:~/Documentos/test$ ls
ExamenASO  Tarea2.py  Tarea3.py
brus@brus-VirtualBox:~/Documentos/test$ python3 Tarea3.py
Proceso completado.
brus@brus-VirtualBox:~/Documentos/test$ ls
ExamenASO  folder0  folder1  folder2  folder3  folder4  Tarea2.py  Tarea3.py
brus@brus-VirtualBox:~/Documentos/test$ cd folder0
brus@brus-VirtualBox:~/Documentos/test/folder0$ ls
fichero0.txt  fichero2.txt  fichero4.txt  fichero6.txt  fichero8.txt
fichero1.txt  fichero3.txt  fichero5.txt  fichero7.txt  fichero9.txt
brus@brus-VirtualBox:~/Documentos/test/folder0$
```

Tarea 4: Desarrolla el programa con el nombre “tarea4.py” donde se analice el espacio disponible en la partición correspondiente a la raíz(“/”),sacando un mensaje de logging mediante la librería logging en el fichero /home//logs/espacio.log Si el espacio ocupado es mayor que 80% se usara un mensaje de error. Si el espacio ocupado es mayor que 60% y menor que 80% se usará un mensaje de warning Si el espacio ocupado es mayor que 0% y menor que 60% se usará un mensaje de info

Para crear el programa tenemos que hacer uso de las librerías *psutil* para obtener el espacio de la partición, *os* para la creación del directorio y *logging* para los mensajes.
En principio creé el siguiente código pero luego tuve que rehacer la parte de creación del directorio dado que me daba problemas al crear el servicio más tarde y también cuando el directorio ya estaba creado .

```python
#importamos las librerias de psutil para ver el espacio de la particion
#la libreria os para que cree el directorio de log
#y la libreria logging para los mensajes
import psutil
import logging
import os

def ver_espacio():
    #Creamos la variable usuario para obtener el nombre del usuario y asi
    #poder crear el directorio
    usuario = input("Introduce tu nombre de usuario: ")

    #comprobamos que el usuario existe
    if os.path.exists(f'/home/{usuario}'):
        #con os.chdir nos movemos a la ruta donde queremos crear el directorio
        os.chdir(f'/home/{usuario}')
        #con os.system creamos el directorio logs
        os.system('mkdir logs')
        # y a continución creamos el archivo espacio.log
        with open ('logs/espacio.log', 'w') as file:
            pass
    else :
        print("Error el introducir nombre de usuario o no existe")


    #ahora creamos la varaible espacio y llamamos a la funcion disk_usage indicando la raiz
    espacio = psutil.disk_usage('/')
    #transformamos el resultado de espacio a un porcentaje con la funcion percent
    porcentaje = espacio.percent

    #configuramos logging para que se ejecute en la ruta creada
    logging.basicConfig(filename=f'/home/{usuario}/logs/espacio.log')
    #por ultimo creamos un if que muestre los mensajes dependiendo del espacio ocupado
    if porcentaje >= 80:
        logging.error(f'Espacio ocupado: {porcentaje}%  Error: El espacio ocupado es mayor que 80%')
    elif 60 <= porcentaje < 80:
        logging.warning(f'Espacio ocupado: {porcentaje}%  Warning: El espacio ocupado supera el 60%')
    else:
        logging.info(f'Espacio ocupado: {porcentaje}%  Info: Espacio libre suficiente')

ver_espacio()
```

A continuación está el código final, en esencia es lo mismo pero he separado el código en tres funciones y he simplificado la funcion de creacion del directorio.

```python
#importamos las librerias de psutil para ver el espacio de la particion
#la libreria os para que cree el directorio de log
#y la libreria logging para los mensajes
import os
import psutil
import logging

#Funcion para crear los directorios, en este caso he utilizado os.makedirs en lugar de
#chdir y system, ademas lo he empaquetado en una funcion para que sea mas facil despues
#realizar la lógica de si existe o no el directorio
def crear_directorio_logs(usuario):
    ruta_logs = f'/home/{usuario}/logs'
    if not os.path.exists(ruta_logs):
        os.makedirs(ruta_logs)
        print(f'Se ha creado el directorio de logs en {ruta_logs}')
    else:
        print(f'El directorio de logs en {ruta_logs} ya existe. No ha sido necesario crearlo.')

#Funcion para configurar el directorio del logging que la recoge de la funcion ver_espacio
def configurar_registro(ruta_archivo):
    logging.basicConfig(filename=ruta_archivo)
```

```python
#Por ultimo la funcion ver_espacio comprueba si existe el directorio, en caso de que no,
def ver_espacio():
    usuario = input("Introduce tu nombre de usuario: ")
    if os.path.exists(f'/home/{usuario}'):

        #llama a la funcion crear_directorio_logs,
        crear_directorio_logs(usuario)

        #despues usa psutil para ver el espacio del disco y lo transforma a porcentaje
        espacio = psutil.disk_usage('/')
        porcentaje = espacio.percent

        #luego se llama a la funcion configurar registro con los datos del usuario
        configurar_registro(f'/home/{usuario}/logs/espacio.log')

        #y por ultimo se realiza el if que se encarga de los mensajes
        if porcentaje >= 80:
            logging.error(f'Espacio ocupado: {porcentaje}%  Error: El espacio ocupado es mayor que 80%')
        elif 60 <= porcentaje < 80:
            logging.warning(f'Espacio ocupado: {porcentaje}%  Warning: El espacio ocupado supera el 60%')
        else:
            logging.info(f'Espacio ocupado: {porcentaje}%  Info: Espacio libre suficiente')
    else:
        print("Error al introducir nombre de usuario o no existe")

#He comentado la llamada dado que al llamarlo desde Tarea5 la funcion se ejecutaba dos veces
#ver_espacio()
```

```
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ python3 Tarea4.py
Introduce tu nombre de usuario: brus
Se ha creado el directorio de logs en /home/brus/logs
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ cd
brus@brus-VirtualBox:~$ cd logs
brus@brus-VirtualBox:~/logs$ cat espacio.log
WARNING:root:Espacio ocupado: 69.4%  Warning: El espacio ocupado supera el 60%
brus@brus-VirtualBox:~/logs$
```

Tarea 5: define una función dentro del fichero “tarea4.py”y copia el código que creaste en la tarea 4. A continuación, crea el fichero “tarea5.py” e importa el fichero “tarea4.py” y llama a la función definida en él.

Como he mostrado en la tarea anterior en un principio sólo definí una función para todo el codigo pero finalmente lo dividi en tres funciones las cuales dos son llamadas por la función ver_espacio

Y en el fichero Tarea5.py llamamos a Tarea4 e importamos la función:

Tarea5.py

```
1  from Tarea4  import ver_espacio
2
3  ver_espacio()
```

Aquí podemos observar que a pesar de que el directorio ya está creado, el mensaje se agrega.

```
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ python3 Tarea5.py
Introduce tu nombre de usuario: brus
El directorio de logs en /home/brus/logs ya existe. No ha sido necesario crearlo.
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ cd
brus@brus-VirtualBox:~$ cd logs/
brus@brus-VirtualBox:~/logs$ cat espacio.log
WARNING:root:Espacio ocupado: 69.4%  Warning: El espacio ocupado supera el 60%
WARNING:root:Espacio ocupado: 69.4%  Warning: El espacio ocupado supera el 60%
brus@brus-VirtualBox:~/logs$
```

Tarea 6: Crea un servicio llamado “espacio.service” que llame al fichero creado en la “tarea5.py” cada 10 segundos.

Para crear el servicio vamos al siguiente directorio y como admin lo creamos.

```
brus@brus-VirtualBox:/lib/systemd/system$ sudo nano espacio.service
```

Para que se ejecute a los 10 segundos escribimos 10 en *RestartSec*. En *ExecStart* indicamos nuestra versión de python y a continuación la ruta del archivo a ejecutar

```
[Unit]
Description=Servicio para ejecutar Tarea5.py
After=network.target
StartLimitIntervalSec=0

[Service]
Type=simple
Restart=always
RestartSec=10
User=root
ExecStart=/usr/bin/python3 /home/brus/Documentos/test/ExamenASO/Tarea5.py

[Install]
WantedBy=multi-user.target
```

Al principio me dio un error porque tenia mal instalada la librería psutil

```
brus@brus-VirtualBox:~/logs$ sudo systemctl status espacio.service
○ espacio.service - Servicio para ejecutar Tarea5.py
     Loaded: loaded (/lib/systemd/system/espacio.service; enabled; vendor prese>
     Active: inactive (dead) (Result: exit-code) since Mon 2024-02-26 13:01:25 >
    Process: 5557 ExecStart=/usr/bin/env python3 /home/brus/Documentos/test/Exa>
   Main PID: 5557 (code=exited, status=1/FAILURE)
        CPU: 50ms

feb 26 13:01:20 brus-VirtualBox systemd[1]: espacio.service: Main process exite>
feb 26 13:01:20 brus-VirtualBox systemd[1]: espacio.service: Failed with result>
feb 26 13:01:25 brus-VirtualBox systemd[1]: Stopped Servicio para ejecutar Tare>
lines 1-10/10 (END)
```

Y después me daba fallo porque al parecer los servicios no puede ejecutar inputs para que el usuario introduzca datos

```
brus@brus-VirtualBox:/var/log$ tail -n 10 syslog
Feb 26 18:29:53 brus-VirtualBox systemd[1]: Stopped Servicio para ejecutar Tarea5.py.
Feb 26 18:29:53 brus-VirtualBox systemd[1]: Started Servicio para ejecutar Tarea5.py.
Feb 26 18:29:53 brus-VirtualBox python3[5109]: Introduce tu nombre de usuario: Traceback (mo
st recent call last):
Feb 26 18:29:53 brus-VirtualBox python3[5109]:   File "/home/brus/Documentos/test/ExamenASO/
Tarea5.py", line 3, in <module>
Feb 26 18:29:53 brus-VirtualBox python3[5109]:     ver_espacio()
Feb 26 18:29:53 brus-VirtualBox python3[5109]:   File "/home/brus/Documentos/test/ExamenASO/
Tarea4.py", line 26, in ver_espacio
Feb 26 18:29:53 brus-VirtualBox python3[5109]:     usuario = input("Introduce tu nombre de u
suario: ")
Feb 26 18:29:53 brus-VirtualBox python3[5109]: EOFError: EOF when reading a line
Feb 26 18:29:53 brus-VirtualBox systemd[1]: espacio.service: Main process exited, code=exite
d, status=1/FAILURE
Feb 26 18:29:53 brus-VirtualBox systemd[1]: espacio.service: Failed with result 'exit-code'.
```

Así que lo que hice fue modificar la Tarea4 y en lugar de un input puse mi nombre de usuario.

```python
#Por ultimo la funcion ver_espacio comprueba si existe el direc
def ver_espacio():
    #usuario = input("Introduce tu nombre de usuario: ")
    usuario = 'brus'
    if os.path.exists(f'/home/{usuario}'):
```

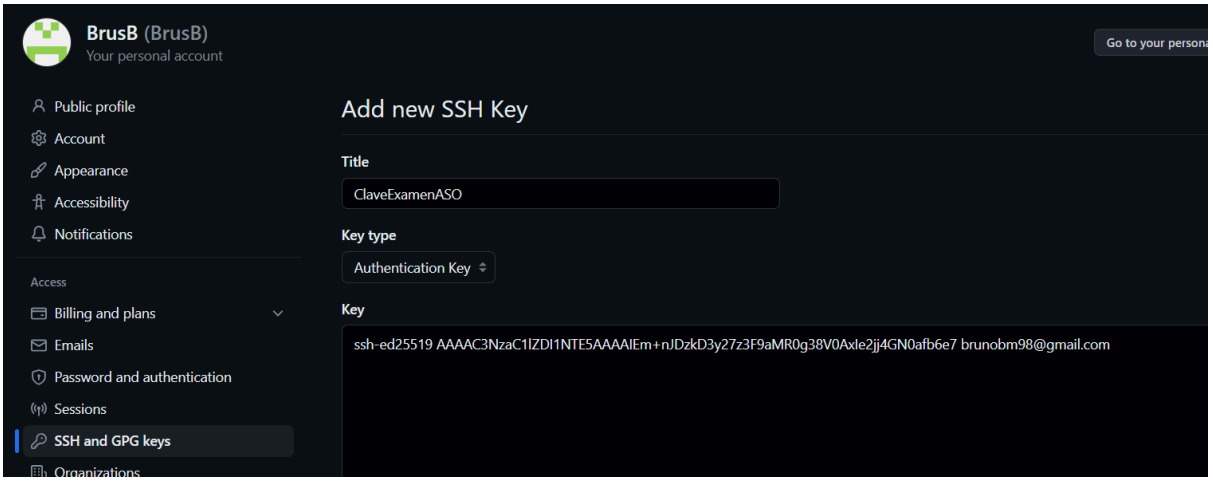
Y de esta forma ya funciona sin problema

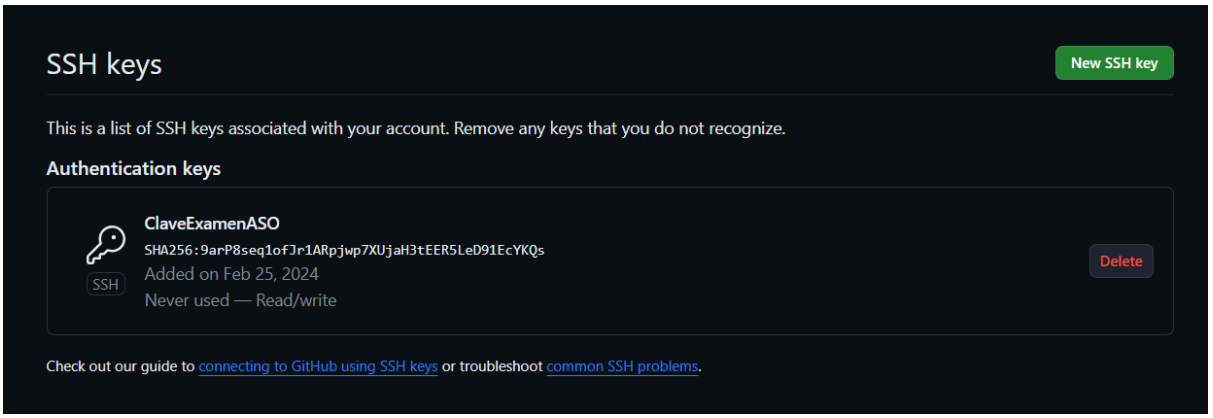
```
brus@brus-VirtualBox:~$ sudo systemctl restart espacio.service
brus@brus-VirtualBox:~$ sudo systemctl start espacio.service
brus@brus-VirtualBox:~$ sudo systemctl status espacio.service
● espacio.service - Servicio para ejecutar Tarea5.py
     Loaded: loaded (/lib/systemd/system/espacio.service; enabled; vendor preset: enabled)
     Active: activating (auto-restart) since Mon 2024-02-26 18:37:05 CET; 2s ago
    Process: 5290 ExecStart=/usr/bin/python3 /home/brus/Documentos/test/ExamenASO/Tarea5.py
   Main PID: 5290 (code=exited, status=0/SUCCESS)
        CPU: 27ms

feb 26 18:37:05 brus-VirtualBox systemd[1]: espacio.service: Deactivated successfully.
...skipping...
● espacio.service - Servicio para ejecutar Tarea5.py
     Loaded: loaded (/lib/systemd/system/espacio.service; enabled; vendor preset: enabled)
     Active: activating (auto-restart) since Mon 2024-02-26 18:37:05 CET; 2s ago
    Process: 5290 ExecStart=/usr/bin/python3 /home/brus/Documentos/test/ExamenASO/Tarea5.py
   Main PID: 5290 (code=exited, status=0/SUCCESS)
        CPU: 27ms
```

```
brus@brus-VirtualBox:~$ cd logs/
brus@brus-VirtualBox:~/logs$ cat espacio.log
WARNING:root:Espacio ocupado: 69.5%  Warning: El espacio ocupado supera el 60%
WARNING:root:Espacio ocupado: 69.5%  Warning: El espacio ocupado supera el 60%
WARNING:root:Espacio ocupado: 69.5%  Warning: El espacio ocupado supera el 60%
WARNING:root:Espacio ocupado: 69.5%  Warning: El espacio ocupado supera el 60%
WARNING:root:Espacio ocupado: 69.5%  Warning: El espacio ocupado supera el 60%
WARNING:root:Espacio ocupado: 69.5%  Warning: El espacio ocupado supera el 60%
WARNING:root:Espacio ocupado: 69.5%  Warning: El espacio ocupado supera el 60%
WARNING:root:Espacio ocupado: 69.5%  Warning: El espacio ocupado supera el 60%
```

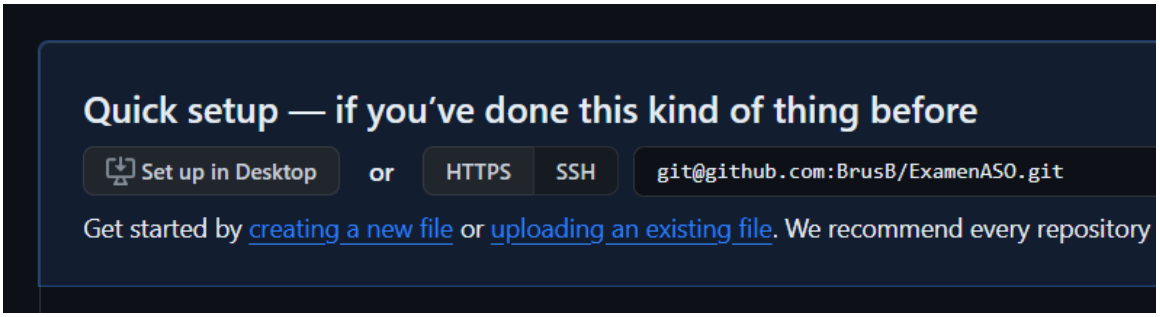
Tarea 7: Crea un repositorio en github y otro en bitbucket y añade allí tu clave pública SSH, así como la mia: ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOP+jPVj13h6gmYJbflcZlIpD7L3hrHD+Aeq75+DVYx 5 ies.fernandosanchez@gmail.com
Sube en estos dos repositorios todas las tareas y documenta la creación e incorporación de las claves públicas en un documento PDF, así como los comandos ejecutados para las subidas de los ficheros a cada uno de los documentos. Sube el documento PDF a continuación al AulaVirtual de la asignatura. Añade tus dos repositorios a la propia entrega del AulaVirtual.

A continuación para agregar nuestra clave pública vamos a *Settings,* hacemos clic en el menú de *SSH AND GPG keys* y le damos a *New SSH key*. Una vez ahí le ponemos un nombre para identificar nuestra clave y la pegamos.





Para crear el repositorio simplemente vamos a nuestra cuenta de GitHub y hacemos clic en “Create a new repository”.
Al crearlo nos aparece la opción de configurarlo a través de HTTPS o SSH, le damos a SSH y copiamos la dirección de nuestro repositorio.



Después vamos al directorio donde queramos tener el repositorio y lo clonamos utilizando el comando *git clone* seguido de la URL SSH y ya tendríamos el repositorio conectado, en mi caso ExamenASO.

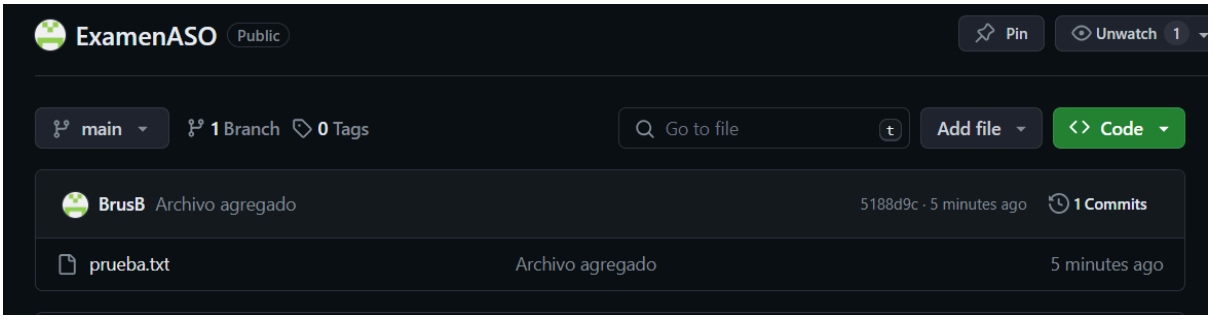
```
brus@brus-VirtualBox:~/Documentos/test$ git clone git@github.com:BrusB/ExamenASO.git
Clonando en 'ExamenASO'...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
warning: Pareces haber clonado un repositorio sin contenido.
```

A continuación he creado un txt para comprobar que funciona. Después con el comando *git add* para agregar el archivo, luego el comando *git commit* para que confirme los cambios y el comando *git push origin main* que lo envía a el repositorio remoto (*main)*.
Para ejecutar los comandos primero tuve que identificarme con mi correo electrónico utilizando el comando *git config --global user.email "brunobm98@gmail.com".*

```
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ ls
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ nano prueba.txt
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git add prueba.txt
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git commit -m "Archivo agregado"
Identidad del autor desconocido

*** Por favor cuéntame quién eres.

Ejecuta

  git config --global user.email "you@example.com"
  git config --global user.name "Tu Nombre"

para configurar la identidad por defecto de tu cuenta.
Omite --global para configurar tu identidad solo en este repositorio.

fatal: no es posible auto-detectar la dirección de correo (se obtuvo 'brus@brus-Virt
ualBox.(none)')
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git config --global user.email "br
unobm98@gmail.com"
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git commit -m "Archivo agregado"
[main (commit-raíz) 5188d9c] Archivo agregado
 1 file changed, 1 insertion(+)
 create mode 100644 prueba.txt
```

```
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git push origin main
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 232 bytes | 232.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To github.com:BrusB/ExamenASO.git
 * [new branch]      main -> main
```

Y ya tendríamos nuestro archivo subido al repositorio.



Por último aquí dejo como he agregado los archivos de python al repositorio:

```
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git init
Reinicializado el repositorio Git existente en /home/brus/Documentos/test/ExamenASO/.git/
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git config --global user.name 'brus'
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git add __pycache__/
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git add *.py
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git commmit -m 'Añadiendo archivos de examen'
git: 'commmit' no es un comando de git. Mira 'git --help'.

El comando más similar es
        commit
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git commit -m 'Añadiendo archivos de examen'
[main 9765ed6] Añadiendo archivos de examen
 6 files changed, 145 insertions(+)
 create mode 100644 Tarea2.py
 create mode 100644 Tarea3.py
 create mode 100644 Tarea4.py
 create mode 100755 Tarea5.py
 create mode 100644 __pycache__/Tarea4.cpython-310.pyc
 create mode 100644 cuatro.py
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$ git push origin main
Enumerando objetos: 10, listo.
Contando objetos: 100% (10/10), listo.
Comprimiendo objetos: 100% (8/8), listo.
Escribiendo objetos: 100% (9/9), 3.88 KiB | 1.94 MiB/s, listo.
Total 9 (delta 0), reusados 0 (delta 0), pack-reusados 0
To github.com:BrusB/ExamenASO.git
   5188d9c..9765ed6  main -> main
brus@brus-VirtualBox:~/Documentos/test/ExamenASO$
```

Adjunto el enlace del repositorio: https://github.com/BrusB/ExamenASO

Al intentar agregar tu clave me ha dado el siguiente error: