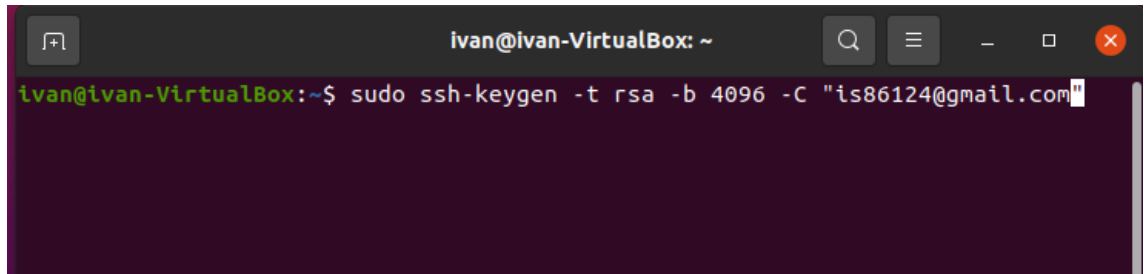


# Examen ASO ASIR 2V Iván Sánchez

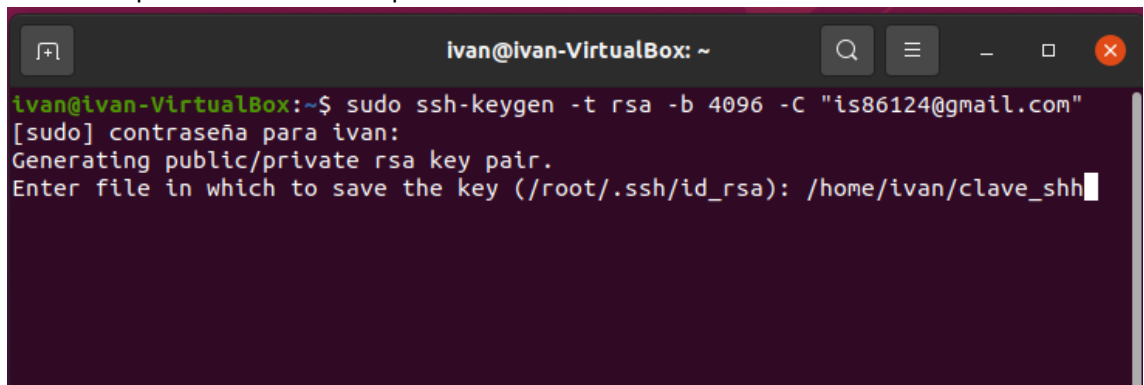
Tarea 1: Crea una Clave SSH en tu sistema operativo y aporta tu clave publica, es decir, <nombre>, pub

- En primer lugar en la terminal de Ubuntu ejecutamos el siguiente comando para generar la clave SSH



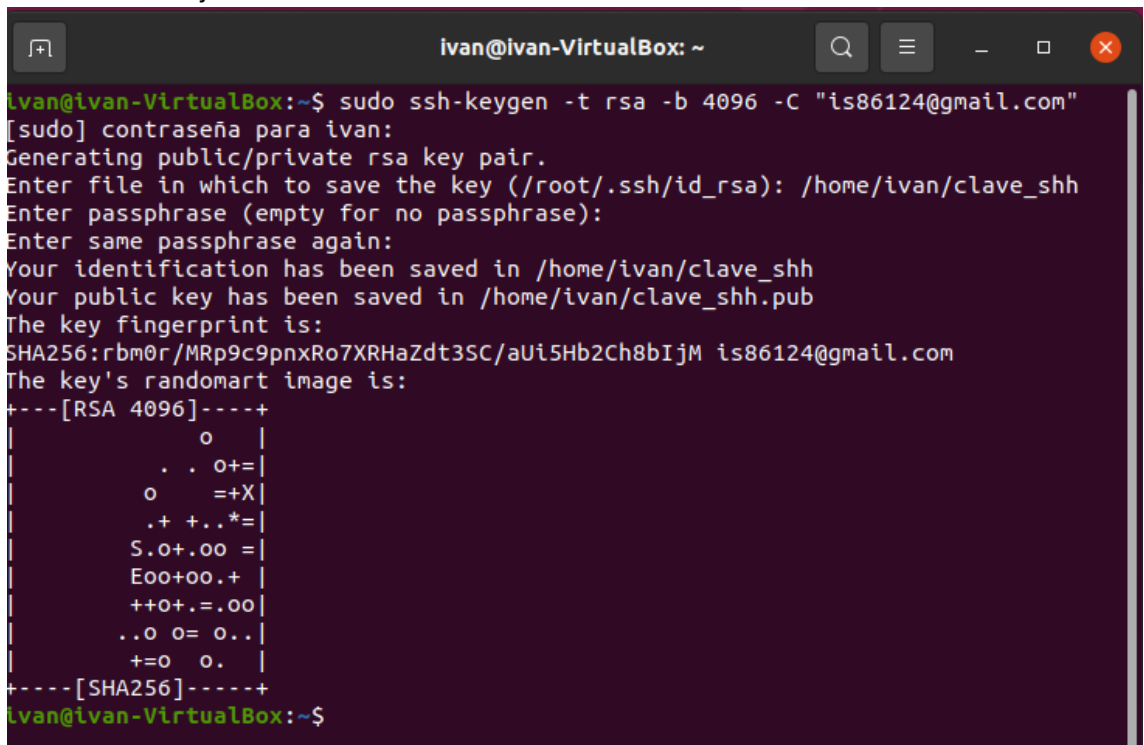
```
ivan@ivan-VirtualBox: ~  
ivan@ivan-VirtualBox:~$ sudo ssh-keygen -t rsa -b 4096 -C "is86124@gmail.com"
```

- Ahora nos pedirá una ubicación para almacenar la clave



```
ivan@ivan-VirtualBox:~$ sudo ssh-keygen -t rsa -b 4096 -C "is86124@gmail.com"  
[sudo] contraseña para ivan:  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa): /home/ivan/clave_sshh
```

- Nos pedirá que si queremos una contraseña para proteger la clave, como es opcional en mi caso lo dejare en blanco



```
ivan@ivan-VirtualBox:~$ sudo ssh-keygen -t rsa -b 4096 -C "is86124@gmail.com"  
[sudo] contraseña para ivan:  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa): /home/ivan/clave_sshh  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ivan/clave_sshh  
Your public key has been saved in /home/ivan/clave_sshh.pub  
The key fingerprint is:  
SHA256:rbm0r/MRp9c9pnxRo7XRHaZdt3SC/aui5Hb2Ch8bIjM is86124@gmail.com  
The key's randomart image is:  
+---[RSA 4096]---+  
|                 |  
|  o               |  
| . . o+=         |  
| o   =+X         |  
| .+ +..*=        |  
| S.O+.OO =       |  
| Eoo+oo.+        |  
| ++o+.=.OO       |  
| ..O O= O..      |  
| +o  O.          |  
+---[SHA256]-----+  
ivan@ivan-VirtualBox:~$
```

- Para encontrar la clave publica generado nos vamos a la carpeta previamente indicada donde la guardamos, allí estará con extensio.pub

```
ivan@ivan-VirtualBox:~$ ls
clave_ssh      Descargas      Escritorio     logs           Plantillas     Videos
clave_ssh.pub  Documentos     Imágenes      Música         Público
```

- Para ver la clave publica usamos el siguiente comando

```
ivan@ivan-VirtualBox: ~
ivan@ivan-VirtualBox:~$ cat clave_ssh.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQDH9Xv6qPhDoi+BYp6xYdLxD1Xg6jeHLd/0NurvG51k
FRXCwV0cLR43POMMDNFS+5gMiNRaWvMkqDK0HPecMh0EM6nsQCh11Y4HX790jbbEceQDYt6CeI4AB4fc
RCtAXjxH0tjJ4a9IuoJ6cy5lIpchfKFGT+lvj2yoVKgdd95989njEt8XqP97NzHF5SDU3eNMMUvA1TLZ
J20g+w/DS7wptIq/HYTIrHKAKRMzjMr+JEyOSnQ0+OC42X+SrAaGQaVQ70c1sSAT8enjRN0EB9rk1Tcu
FlNxo0mBJDv5/83w0Lmr3FnkngpMLUu9qXSRIxVHQgM7nngBF2bBLfHER04y4jppTYVOne7eDDqEvVL7
Fv4ey8g/8YtYTb0AC4bBwHi94lKu182YWSzEoJkRieweQnv8952YgX7yFpTJCSL4CyKJiYc4FecKMP0L
LWuU9Vd02rJTH0yhLkpNu6LIuGMZJRfMMc08N4hnmVH8qUGS8c1K01Zqb2Z2qfm/ozIBaeUpipeRdRKT
Vf1xV9NsW3YmkLOotyHuG438vEk103fcqLXBt6Uv09ff5iJiDhLAGNeFTNkxPrWlyKm6ErQCWZb6ccf0
mk80Gy8+zv2Fw1nwE2ncjLjmtH8MSfn7V1u1gVUHFswocRH4J4oPlqluxqhYcbc9T1JfP4YtveQY3yaG
NQ== is86124@gmail.com
ivan@ivan-VirtualBox:~$
```

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 asi:

```
/dev/sda1 78,9%
```

```
/dev/sdb1 18,5%
```

- En primer lugar, instalaremos el visual studio code
  - sudo apt update
  - sudo apt install software-properties-common apt-transport-https wget
  - wget -q https://packages.microsoft.com/keys/microsoft.asc -O- | sudo apt-key add -
  - sudo add-apt-repository "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main"
  - sudo apt update
  - sudo apt install code
- Una vez tengamos instalado el visual studio code instalremos python3
  - sudo apt update
  - sudo apt install python3
- Una vez instalado python3 instalaremos el paquete psutil que sirve para obtener información sobre la partición raíz ('/') del sistema y calcular el porcentaje de espacio ocupado en ella.
  - sudo apt update
  - sudo apt install python3-pip
  - pip3 install psutil

- Ahora nos iremos al visual studio code y crearemos el script de Python y lo ejecutaremos

```

1 import psutil
2
3 def obtener_porcentaje_espacio(particion):
4     espacio = psutil.disk_usage(particion)
5     porcentaje = espacio.percent
6     return porcentaje
7
8 def main():
9     particiones = ["/dev/sda1"] # Lista de las particiones que quieres verificar
10    for particion in particiones:
11        porcentaje = obtener_porcentaje_espacio(particion)
12        print(f"{particion} (porcentaje: {porcentaje})")
13
14 if __name__ == "__main__":
15     main()

```

```

[Running] python3 -u "/home/ivan/Escritorio/tarea2.py"
/dev/sda1 0.6%
[Done] exited with code=0 in 0.132 seconds

```

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, creando dentro de ellos 10 ficheros con el siguiente nombre y siguiente contenido:  
 nombre fichero: fichero1.txt  
 contenido: Este es el contenido del fichero 1

- Nos iremos al visual studio code y crearemos el script de Python y lo ejecutaremos

```

1 import os
2
3 def crear_carpeta_y_ficheros():
4     for i in range(1, 6): # Itera 5 veces
5         nombre_carpeta = f"folder{i}"
6         os.makedirs(nombre_carpeta, exist_ok=True) # Crea la carpeta si no existe
7         for j in range(1, 11): # Crea 10 archivos dentro de cada carpeta
8             nombre_fichero = f"{nombre_carpeta}/fichero{j}.txt"
9             with open(nombre_fichero, "w") as archivo:
10                 archivo.write(f"Este es el contenido del fichero {j}\n")
11             print(f"Se crearon los archivos en la carpeta {nombre_carpeta}")
12
13 def main():
14     crear_carpeta_y_ficheros()
15
16 if __name__ == "__main__":
17     main()

```

```

[Running] python3 -u "/home/ivan/Escritorio/tarea3.py"
Se crearon los archivos en la carpeta folder1
Se crearon los archivos en la carpeta folder2
Se crearon los archivos en la carpeta folder3
Se crearon los archivos en la carpeta folder4
Se crearon los archivos en la carpeta folder5
[Done] exited with code=0 in 0.217 seconds

```

Tarea 4: Desarrolla el programa con el nombre "tarea4.py" donde se analice el espacio disponible en la partición correctamente a la raíz ("/"), sacando un mensaje de logging mediante una libreria logging en el fichero /home/ivan/logs/espacio.log

Si el espacio ocupado es mayor que 80% se usara un mensaje de error

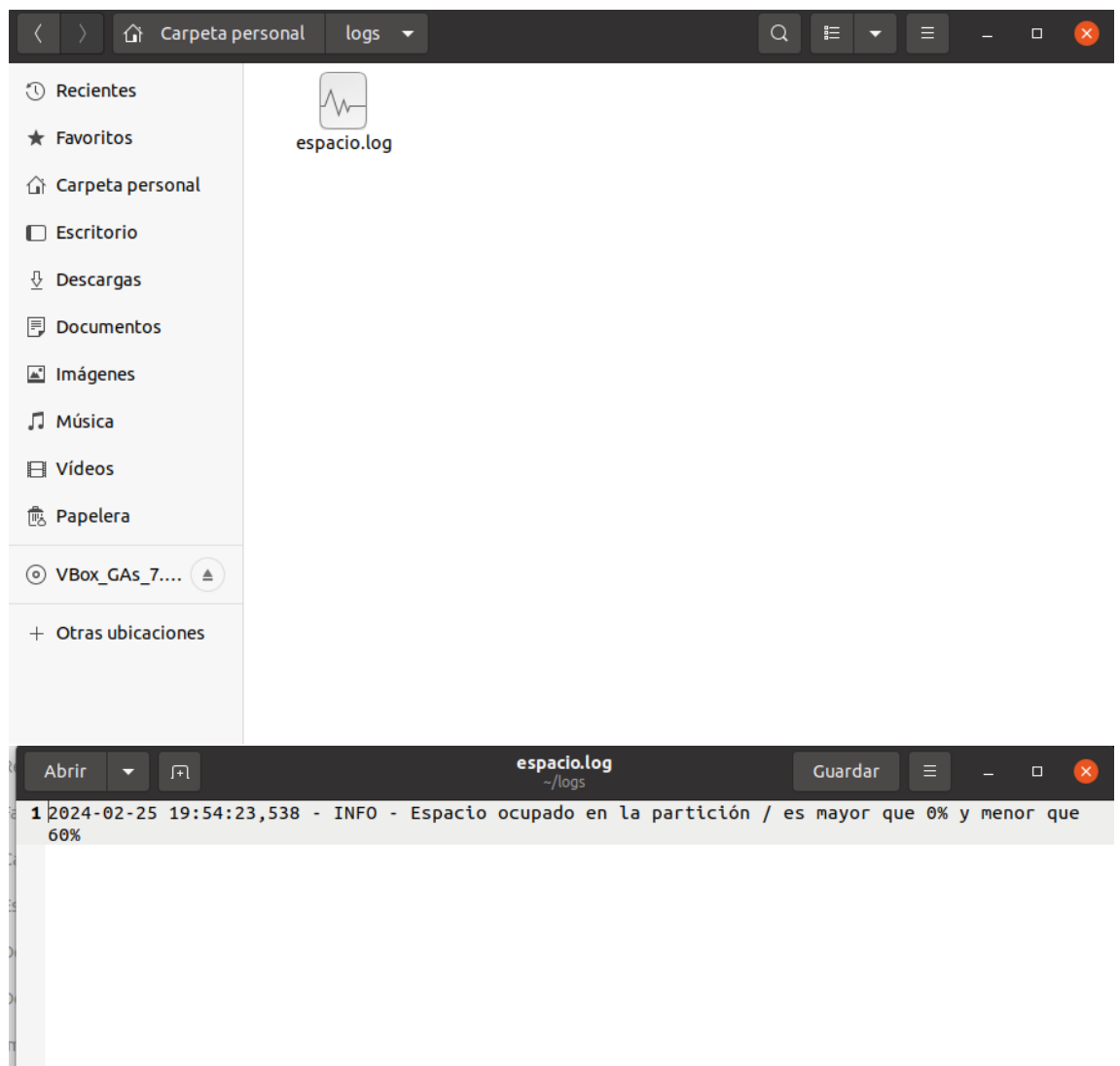
Si el espacio ocupado es mayor que 60% y menor que 80% se usara un mensaje de warning

Si el espacio ocupado es mayor que 0% y menos que 60% se usara un mensaje de info

- Nos iremos al visual studio code y crearemos un script de Python y lo ejecutaremos

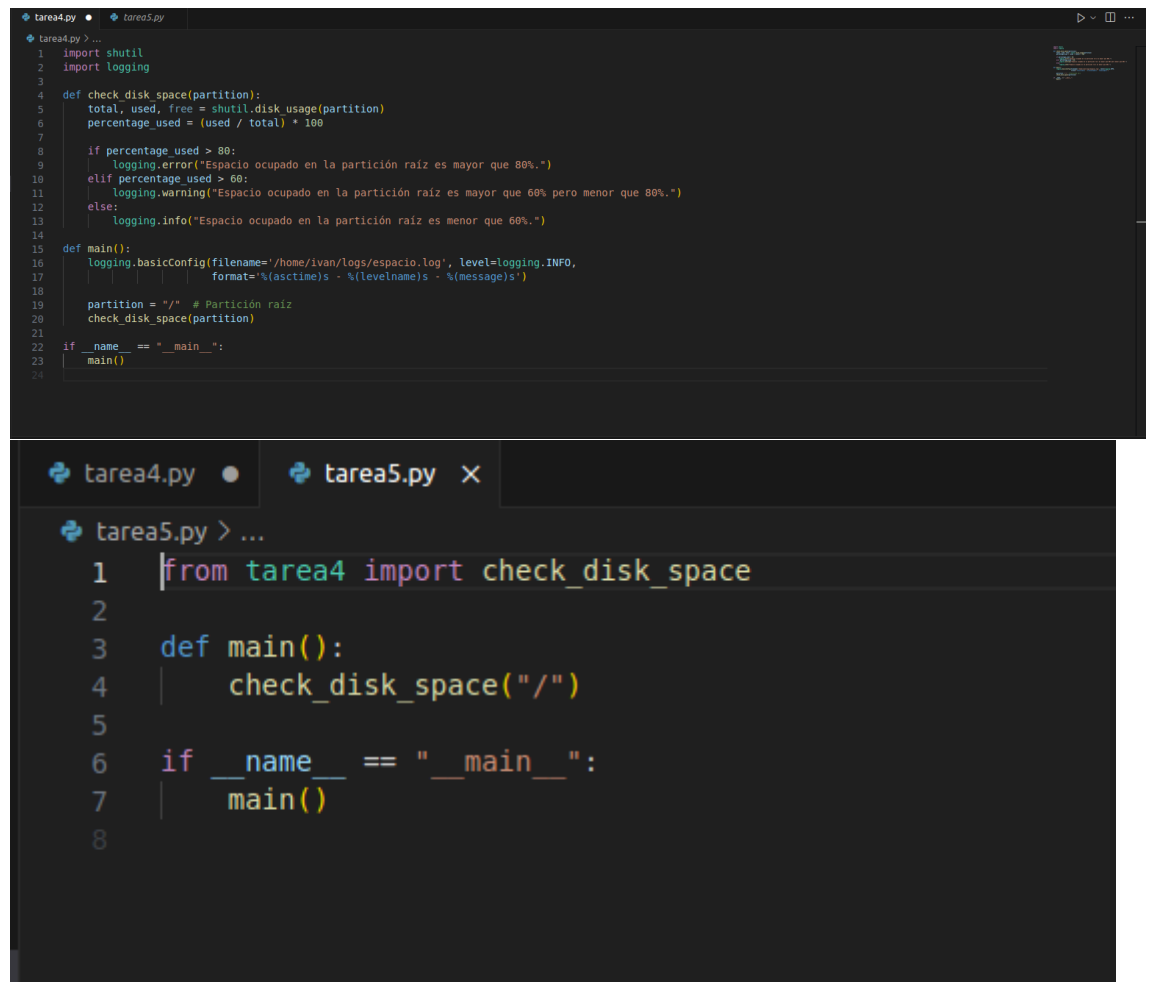
```
tarea4.py x
tarea4.py > ...
1 import shutil
2 import logging
3
4 # Configuración básica del registro
5 logging.basicConfig(filename='/home/ivan/logs/espacio.log', level=logging.INFO,
6                     format='%(asctime)s - %(levelname)s - %(message)s')
7
8 # Función para verificar el espacio en disco y registrar mensajes según el nivel de ocupación
9 def check_disk_space(partition):
10     total, used, free = shutil.disk_usage(partition)
11     percentage_used = (used / total) * 100
12
13     if percentage_used > 80:
14         logging.error("Espacio ocupado en la partición raíz es mayor que 80%.")
15     elif percentage_used > 60:
16         logging.warning("Espacio ocupado en la partición raíz es mayor que 60% pero menor que 80%.")
17     else:
18         logging.info("Espacio ocupado en la partición raíz es menor que 60%.")
19
20 def main():
21     partition = "/" # Partición raíz
22     check_disk_space(partition)
23
24 if __name__ == "__main__":
25     main()
26
```

- Si nos vamos a la carpeta de logs veremos un archivo que ha creado volcando el resultado del script



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.

- En primer lugar, moveremos el código existente en "tarea4.py" a una función dentro del mismo archivo, luego crearemos un nuevo archivo "tarea5.py" que importará "tarea4.py" y llamará a esa función.



The image shows a code editor with two tabs: 'tarea4.py' and 'tarea5.py'. The 'tarea4.py' tab is active, showing the following code:

```
1 import shutil
2 import logging
3
4 def check_disk_space(partition):
5     total, used, free = shutil.disk_usage(partition)
6     percentage_used = (used / total) * 100
7
8     if percentage_used > 80:
9         logging.error("Espacio ocupado en la partición raíz es mayor que 80%.")
10    elif percentage_used > 60:
11        logging.warning("Espacio ocupado en la partición raíz es mayor que 60% pero menor que 80%.")
12    else:
13        logging.info("Espacio ocupado en la partición raíz es menor que 60%.")
14
15 def main():
16     logging.basicConfig(filename='/home/ivan/logs/espacio.log', level=logging.INFO,
17                         format='%(asctime)s - %(levelname)s - %(message)s')
18
19     partition = "/" # Partición raíz
20     check_disk_space(partition)
21
22 if __name__ == "__main__":
23     main()
```

The 'tarea5.py' tab is also visible, showing the following code:

```
1 from tarea4 import check_disk_space
2
3 def main():
4     check_disk_space("/")
5
6 if __name__ == "__main__":
7     main()
```

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

- En primer lugar abriremos un editor de texto y crearemos un archivo llamado "espacio.service". Puedes ubicarlo en /etc/systemd/system/ para que sea accesible por systemd.

```
ivan@ivan-VirtualBox: ~
GNU nano 4.8 /etc/systemd/system/espacio.service
[Unit]
Description=Servicio para verificar espacio ocupado en partición raíz
After=network.target

[Service]
ExecStart=/bin/python3 /home/ivan/Escritorio/tarea5.py
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target

[ 11 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^T Ortografía ^_ Ir a línea
```

- Después de crear el archivo del servicio, necesitaremos recargar systemd para que reconozca los cambios.

```
ivan@ivan-VirtualBox: ~$ sudo nano /etc/systemd/system/espacio.service
[sudo] contraseña para ivan:
ivan@ivan-VirtualBox:~$ sudo systemctl daemon-reload
ivan@ivan-VirtualBox:~$
```

- Ahora iniciaremos el servicio espacio.service, pero la acción será realizada en el visual basic, en el archivo espacio.log

```
ivan@ivan-VirtualBox:~$ sudo systemctl start espacio.service
ivan@ivan-VirtualBox:~$
```

```
home > ivan > logs > espacio.log
1 2024-02-25 20:23:54,833 - INFO - Espacio ocupado en la partición / es mayor que 0% y menor que 60%
2 2024-02-26 12:20:59,846 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
3 2024-02-26 12:25:57,840 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
4 2024-02-26 12:26:08,078 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
5 2024-02-26 12:26:12,678 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
6 2024-02-26 12:26:18,330 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
7 2024-02-26 12:26:28,588 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
8 2024-02-26 12:26:38,861 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
9 2024-02-26 12:26:49,080 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
10 2024-02-26 12:26:59,324 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
11 2024-02-26 12:27:09,580 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
12 2024-02-26 12:27:19,837 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
13 2024-02-26 12:27:30,080 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
14 2024-02-26 12:27:40,325 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
15 2024-02-26 12:27:50,584 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
16 2024-02-26 12:28:00,830 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
17 2024-02-26 12:28:11,103 - INFO - Espacio ocupado en la partición raíz es menor que 60%.
```

Tarea 7: Crea un repositorio en github y otro en bitbucket y añade allí tu clave publica SSH, así como la mia:

ssh-ed25519

AAAAC3NzaC1lZDI1NTE5AAAAIOP+jPVj13h6gmYJbflcZllpD7L3hrHD+Aeq75+DVYx

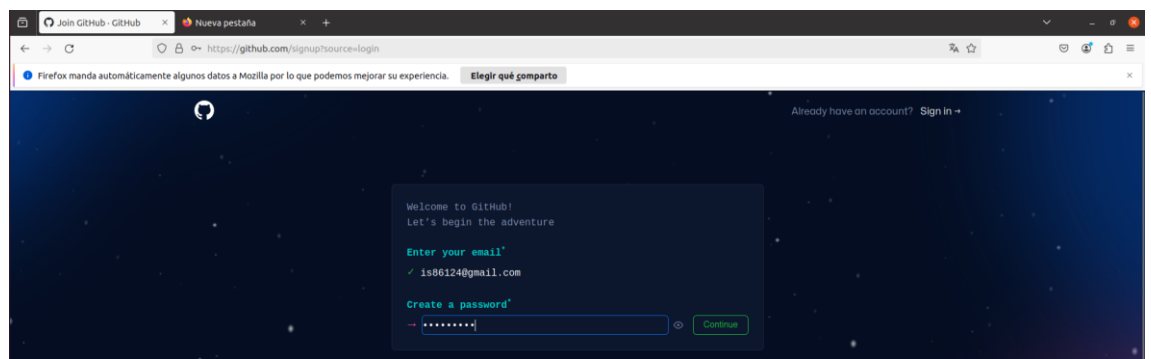
5 [ies.fernandosanchez@gmail.com](mailto:ies.fernandosanchez@gmail.com)

Sube en estos dos repositorios todas las tareas y documenta la creación e incorporación de las claves publicas 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.

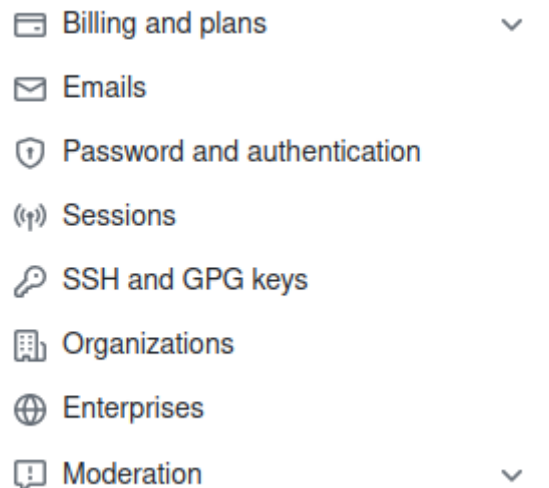
Añade tus dos repositorios a la propia entrega del AulaVirtual.

## Github

- En primer lugar nos creamos una cuenta de Github para crear el repositorio



- Accederemos y hacemos click en la foto de perfil en la esquina superior derecha y seleccionamos settings. En el menú de la izquierda, hacemos click en SSH and GPG keys.



- Hacemos click en New SSH KEY

### SSH keys

New SSH key

There are no SSH keys associated with your account.

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

- Pegamos nuestra clave SSH en el campo Key, le damos un nombre descriptivo a la clave y por ultimo hacemos click en Add SSH Key

### Add new SSH Key

Title

MI clave SSH

Key type

Authentication Key ↕

Key

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQDH9Xv6qPhDoi+BYp6xYdLxD1Xg6jeHLd/
0NUrvG51kFRXCwVOclR43POMMDNFS+5gMiNRaWvMkqDK0HPecMh0EM6nsQCh11Y4HX79OjbbEceQDYt6CeI4AB4fcRCtAXjxHOtjJ4a9luo6
cy5llpchfKFgT+IVj2yoVKgdd95989njEt8XqP97NzHF5SDU3eNMmUvA1TLzJ2Og+w/DS7wptlq/
HYTlrHKAkRMzIMr+JEyOSnQ0+OC42X+SrAaGQaVQ70c1sSAT8enJRNOEB9rk1TcuFlNxoOmBJDv5/83w0Lmr3FnkngpMIUu9qXSRIxVHQgM7n
ngBF2bBIFHERO4y4jppTYVOne7eDDqEvVL7Fv4ey8g/
8YtYTbOAC4bBwHi94IKu182YWSzEoJkRieweQnv8952YgX7yFpTJCSL4CyKJiYc4FecKMP0LLWeU9Vd02rJTH0yhLkpNu6LluGMZJRfMMc08N4
hnmVH8qUGS8c1K01Zqb2Z2qfm/
oziBAeUpipeRdRKTvfxV9NsW3YmkLOotyHuG438vEk1O3fcqLXBt6Uv09fF5jIDhLAGNeFTNkxPrWIYkm6ErQCWZb6ccfOmk8OGy8+zv2Fw1n
wE2ncjLjmtH8MSfn7V1u1gVUHfswocRH4J4oPlqluxqhYcbc9T1JIP4YtvEQY3yaGNQ== is86124@gmail.com
```

Add SSH key

- Ahora iremos a crear un repositorio para subir los documentos, hacemos en settings, repositories y crearemos uno nuevo (create one)

### Repository default branch

Choose the default branch for your new personal repositories. You might want to change the default name due to different workflows, or because your integrations still require "master" as the default branch name. You can always change the default branch name on individual repositories. [Learn more about default branches.](#)

main

Update

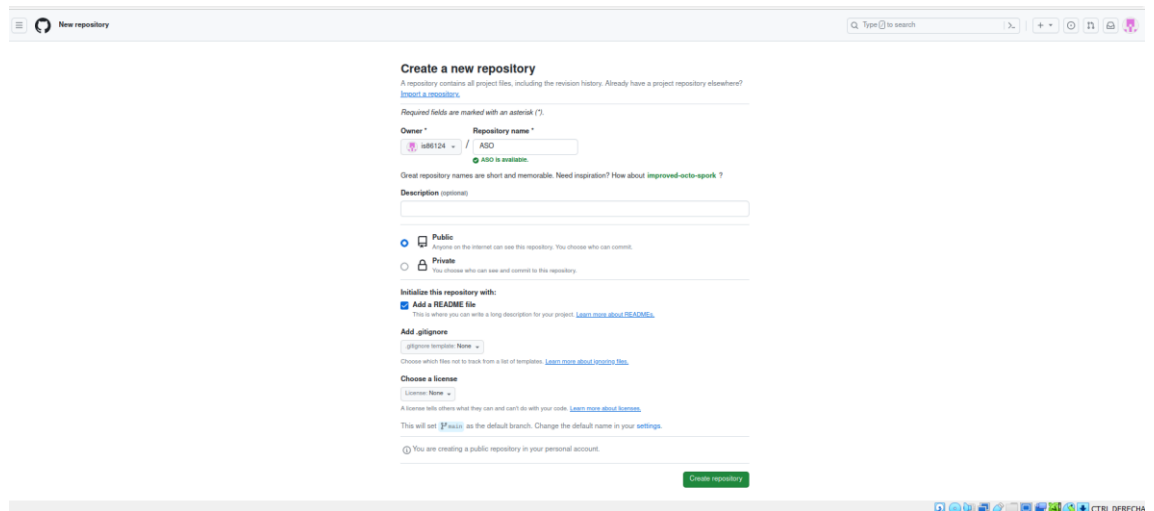
### Repositories

 Repositories  Deleted repositories

You don't have any repositories yet. Would you like to [create one](#)?

- Una vez en la pestaña de creación, ingresaremos un nombre, una descripción (opcional), elegiremos si queremos que el repositorio sea publico o privado. Marcar la casilla Initialize this repository with a README, si deseas que Github cree un archivo README.md inicial para tu repositorio. Una vez rellenado esos campos le daremos al botón Create Repository





- Ahora para poder subir documentos al repositorio en la maquina local, instalaremos github en el propio equipo

```
ivan@ivan-VirtualBox:~$ sudo apt install git
[sudo] contraseña para ivan: 
```

- A continuacion introduciremos el siguiente comando que sirve para clonar en nuestra maquina local el repositorio

```
ivan@ivan-VirtualBox:~/Escritorio$ git clone https://github.com/is86124/ASO
Clonando en 'ASO'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (3/3), 853 bytes | 426.00 KiB/s, listo.
ivan@ivan-VirtualBox:~/Escritorio$
```

- Ahora podremos a colocar los documentos que queramos subir dentro del directorio del repositorio que acabamos de clonar en la maquina virtual, pero para ello nos vamos al directorio del proyecto con el siguiente comando

```
ivan@ivan-VirtualBox: ~/Escritorio
ivan@ivan-VirtualBox:~/Escritorio$ git init
Iniciado repositorio Git vacío en /home/ivan/Escritorio/.git/
ivan@ivan-VirtualBox:~/Escritorio$
```

- Ahora añadiremos los documentos

```
ivan@ivan-VirtualBox:~/Escritorio$ git add tarea2.py
ivan@ivan-VirtualBox:~/Escritorio$ git add tarea3.py
ivan@ivan-VirtualBox:~/Escritorio$ git add tarea4.py
ivan@ivan-VirtualBox:~/Escritorio$ git add tarea5.py
```

- Ahora haremos una comprobación de los archivos

```
ivan@ivan-VirtualBox:~/Escritorio$ git commit -m "repositorio"
[master (commit-raíz) 72114c6] repositorio
4 files changed, 64 insertions(+)
create mode 100644 tarea2.py
create mode 100644 tarea3.py
create mode 100644 tarea4.py
create mode 100644 tarea5.py
ivan@ivan-VirtualBox:~/Escritorio$
```

- Ahora elegiremos la rama de los repositorios hacia la que subir los archivos

```
ivan@ivan-VirtualBox:~/Escritorio$ git branch -M ASO
ivan@ivan-VirtualBox:~/Escritorio$ git remote add origin https://github.com/is86124/ASO/pulls
```

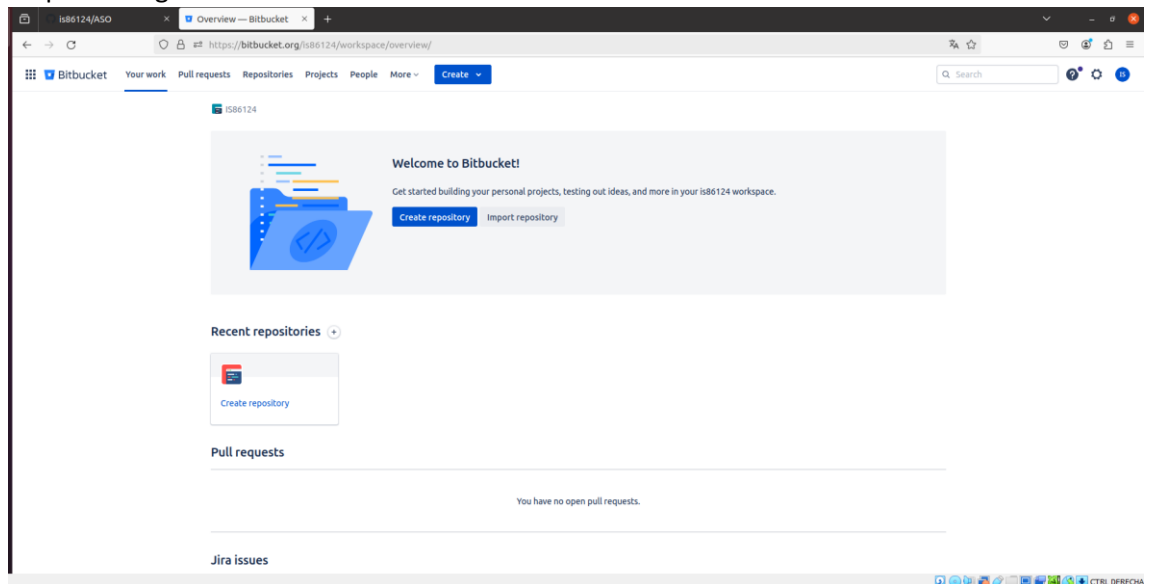
- Por ultimo subiremos los archivos al repositorio remoto con este comando

```
ivan@ivan-VirtualBox:~/Escritorio$ git push origin ASO
Username for 'https://github.com': is86124
Password for 'https://is86124@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Autenticación falló para 'https://github.com/is86124/ASO.git/'
```

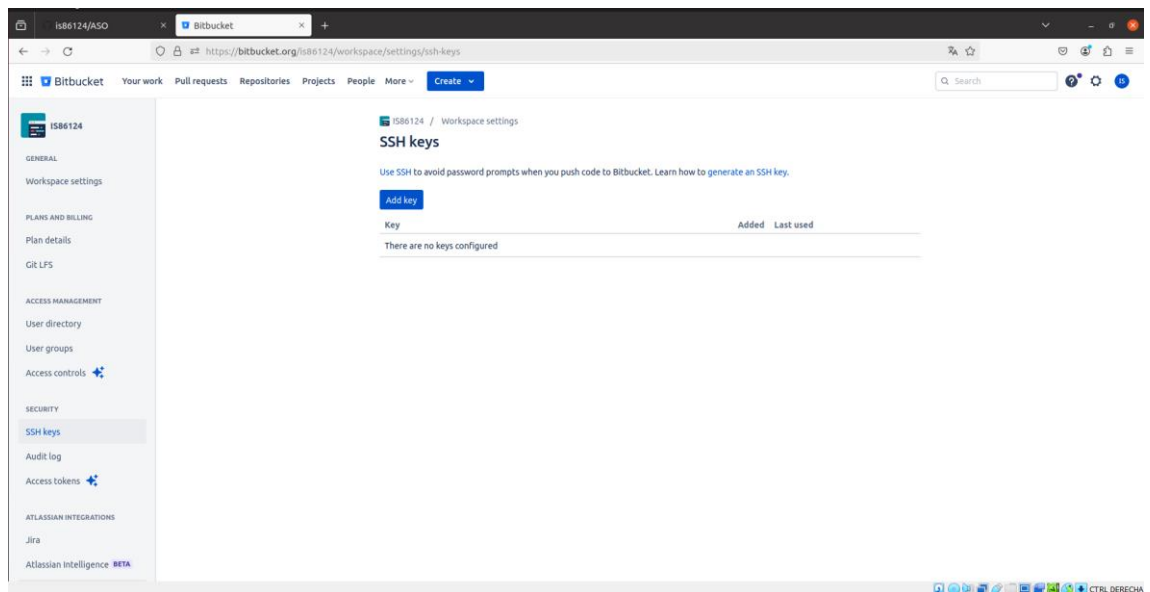
Al subir los archivos al repositorio me da fallo de autenticación por los que te los subo de forma manual.

## Bitbucket

- En primer lugar como antes crearemos una cuenta



- Accederemos y haremos clickk en la foto de perfil en la esquina superior derecha y seleccionamos settings, en el menu de la izquierda haz click en SSH and GPG Keys



## Add SSH key

Label

PERSONAL KEY

Key\*

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQADH9Xv6qPhDoi+BYp6xYdLxD1Xg
6jeHLd/
0NUrvG51kFRXCwVOclR43POMMDNFS+5gMiNRaWvMkqDK0HPecMh0EM6n
sQCh11Y4HX79OjbbEceQDYt6CeI4AB4fcRCtAXjxHOTjJ4a9luoj6cy5llpchfKFg
T+lvj2yoVKgdd95989njEt8XqP97NzHF5SDU3eNMmUvA1TLzJ2Og+w/
DS7wptiq/
HYTlrHKAKRMzjMr+JEyOSnQ0+OC42X+SrAaGQaVQ70c1sSAT8enjRNOEB9rk
```

### Don't have a key?

Learn how to [generate an SSH key](#).

### Already have a key?

Copy and paste your key here .

### Problems adding a key?

Read our [troubleshooting page](#) for common issues.


Add key


Cancel

- Ahora procederemos a crear un repositorio

## Create a new repository

[Import repository](#)

Workspace  IS86124

Project\*  ASO1

Repository name\* ASO1

Access level ☒ Private repository  
Uncheck to make this repository public. Public repositories typically contain open-source code and can be viewed by anyone.

Include a README? Yes, with a tutorial (for begin...)

Default branch name e.g., 'main'

Include .gitignore? Yes (recommended)

[Advanced settings](#)

[Create repository](#) [Cancel](#)

- A la hora de clonar el servidor e introducir las credenciales me da fallo a pesa de las credenciales que introduzco son las correctas

```
ivan@ivan-VirtualBox:~$ git clone https://bitbucket.org/is86124/aso1/src/main/
Clonando en 'main'...
Username for 'https://bitbucket.org': is86124
Password for 'https://is86124@bitbucket.org':
remote: Invalid credentials
fatal: Autenticación falló para 'https://bitbucket.org/is86124/aso1/src/main/'
```