

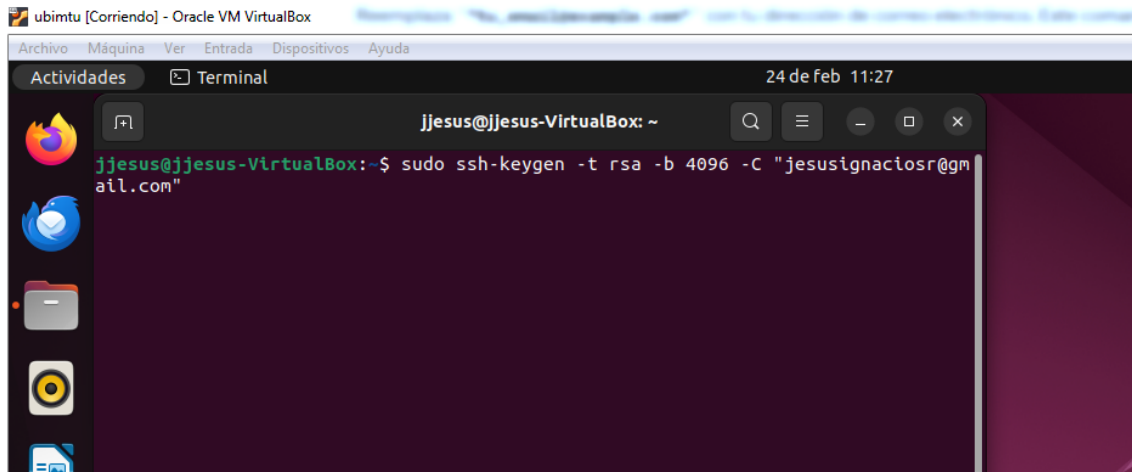
Nombre y apellidos: JESUS IGNACIO SANCHEZ RODRIGUEZ ASIRV2

EXAMEN PRÁCTICO 2EV

Elabora un documento donde expliques la realización de las siguientes tareas.

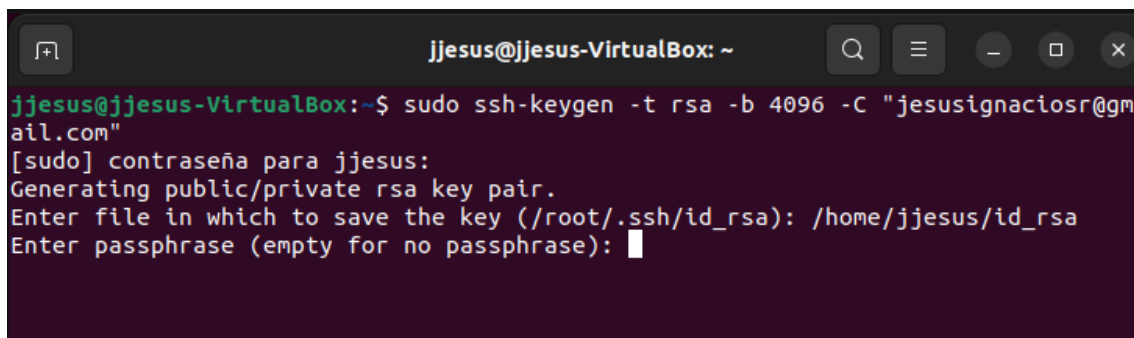
Tarea 1: Crea una clave SSH en tu sistema operativo y aporta tu clave publica, es decir, .pub

En la terminal de Ubuntu ejecutamos el siguiente comando para generar la clave SSH:



```
ubintu [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 24 de feb 11:27
jjesus@jjesus-VirtualBox: ~
jjesus@jjesus-VirtualBox:~$ sudo ssh-keygen -t rsa -b 4096 -C "jesusignaciosr@gmail.com"
```

Nos pide una ubicación para almacenar la clave, la ponemos



```
jjesus@jjesus-VirtualBox: ~
jjesus@jjesus-VirtualBox:~$ sudo ssh-keygen -t rsa -b 4096 -C "jesusignaciosr@gmail.com"
[sudo] contraseña para jjesus:
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /home/jjesus/id_rsa
Enter passphrase (empty for no passphrase):
```

Nos pedirá una frase de paso opcional para proteger la clave. Como no quiero establecer una la dejaré en blanco

```
jjesus@jjesus-VirtualBox: ~  
jjesus@jjesus-VirtualBox:~$ sudo ssh-keygen -t rsa -b 4096 -C "jesusignaciosr@gmail.com"  
[sudo] contraseña para jjesus:  
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa): /home/jjesus/id_rsa  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/jjesus/id_rsa  
Your public key has been saved in /home/jjesus/id_rsa.pub  
The key fingerprint is:  
SHA256:aQ6C/RLEiNAK/i+nNKKIsoSAGNHHF8rVZAmj+9okDkU jesusignaciosr@gmail.com  
The key's randomart image is:  
----[RSA 4096]-----+  
oo . ==o.  
+oo+o+.oo  
*...E.  
+. = . .  
o o * . S  
o o = +  
.o.o+ + .  
* o+.o  
*. *.  
-----[SHA256]-----+  
jjesus@jjesus-VirtualBox:~$
```

Para encontrar la clave publica generada nos vamos a donde indicamos para que se guardara, alli estará el archivo con extensión .pub

```
jjesus@jjesus-VirtualBox:~$ ls  
Descargas  Escritorio  id_rsa.pub  Música  Plantillas  snap  
Documentos id_rsa      Imágenes   open.py  Público     Videos
```

Para ver la clave publica usamos el siguiente comando

```
jjesus@jjesus-VirtualBox:~$ sudo cat id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADOC7dyoWdp1sLYqD9/BAVedD9AYkPVAEf/c2xjqnoh  
iJqeFHnn0sNkWGm4tZTW1HoCiaIN9amv5N4QX5pRNx/Xy/tenIw1/QTomEN+GYELmU9A8ub9KX0SU6Iu  
lvqA9xnXXZDZnjIXx9CR1h5XKz79RyPot7U0j6780FH/oVPR2QflxDdWZU2oJvmoYQXzw749HFmy0n0B  
/16pr9uIY2wzhYxUvyDcL+lQRz0wwQ8hCyVQbvqh2K2jmd90NpCgJS1R8AWHxEPQwaC2/UdXe+JL0/8n  
CJJRWAZsRjMu0+L3xFdZrBKyB5/o6hN4XJSFy7NInWy/OdIgcBqvG6J857Rtdhj/WF5Z6jqoNwNDcxat  
V/JNSpBCE/3EQxlHwTCAXngIT752YMVk1VyjjEg+ilH2dcYlVJ9DitBvKA9QV2rcUEjeQmYPPDtJQCoa  
sl1eVZW9jIVD/WeYk0Cy7i7JCM6TFv/9WnQPX8oLnWffiq0Xqfpqv/YHC05cJ1yvid4Y2G3XSNVvS+eB  
DUJ8cnJMF1zA2Bhevb4vpIwPLARXRf42wMeXLh+PplVOTR9jPbZpdcldcuPbZ0rt/wiFhz7a/nSnafv  
fRWhUDE3tnKtcdUfBgH2GxxUxNp30ASrIXvhUL6Y18gKbDJzaHG0qY7rj/AtcFJxABU1Jr0mDa1WZ2gb  
QQ== jesusignaciosr@gmail.com  
jjesus@jjesus-VirtualBox:~$
```

```
jjesus@jjesus-VirtualBox:~$ sudo mv id_rsa.pub jesus.pub
```

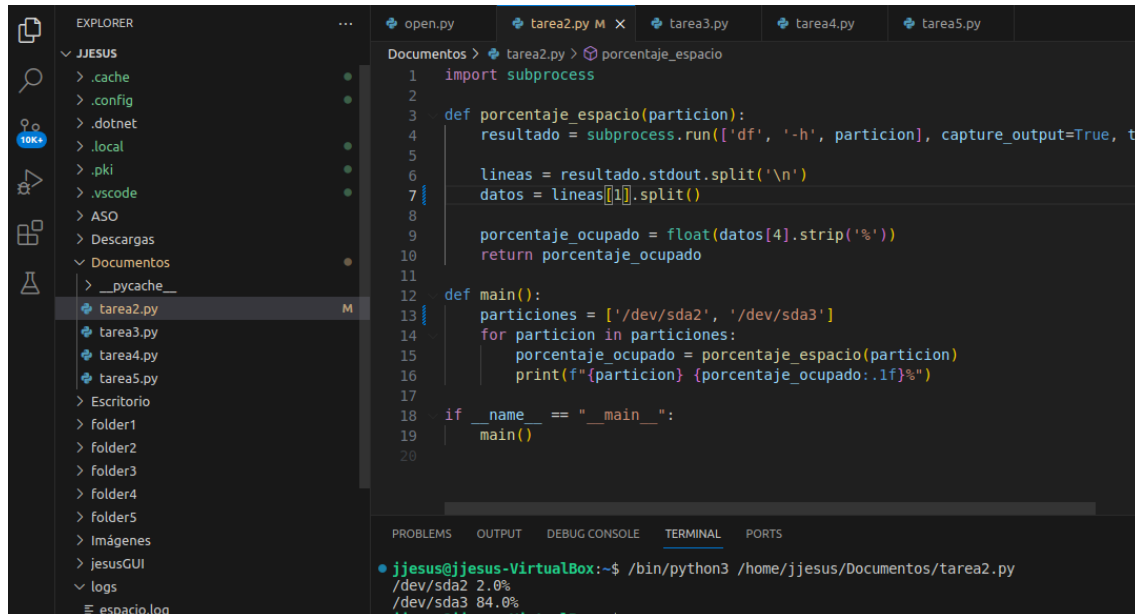
Tarea 2: Crea un programa con el nombre "tarea2.py" en Python que muestre por pantalla el porcentaje de espacio ocupado en cada una de las particiones de tu sistema, de forma que se muestre tal que así:

/dev/sda1 78,9%

/dev/sdb1 18,5%

Utilizamos dos funciones, una primera donde conseguimos la información sobre el espacio ocupado en las particiones del sistema en general, usaremos `df -h` dentro de esa función que muestra las diferentes particiones del sistema informático y el espacio ocupado en cada una.

En la segunda función especificamos las particiones de nuestro disco y hacemos un bucle `for` para que nos indique el espacio ocupado en cada una de ellas. Luego, imprimirá esta información en la consola.



The screenshot shows a Visual Studio Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'JJESUS' with various files and folders. The code editor shows a Python script named 'tarea2.py' with the following code:

```
1 import subprocess
2
3 def porcentaje_espacio(particion):
4     resultado = subprocess.run(['df', '-h', particion], capture_output=True, text=True)
5
6     lineas = resultado.stdout.split('\n')
7     datos = lineas[1].split()
8
9     porcentaje_ocupado = float(datos[4].strip('%'))
10    return porcentaje_ocupado
11
12 def main():
13    particiones = ['/dev/sda2', '/dev/sda3']
14    for particion in particiones:
15        porcentaje_ocupado = porcentaje_espacio(particion)
16        print(f"{particion} {porcentaje_ocupado:.1f}%")
17
18 if __name__ == "__main__":
19    main()
20
```

The terminal at the bottom shows the execution of the script:

```
● jjesus@jjesus-VirtualBox: ~$ /bin/python3 /home/jjesus/Documentos/tarea2.py
/dev/sda2 2.0%
/dev/sda3 84.0%
```

Tarea3: 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

Primero creamos un bucle `for` para que se creen las 5 carpetas, donde también les asignaremos un nombre. Dentro de ese bucle `for` para crear los 10 archivos con su contenido correspondiente.

```
1 import os
2
3 for i in range(1, 6):
4     folder_name = f"folder{i}"
5     os.makedirs(folder_name, exist_ok=True)
6
7     for j in range(1, 11):
8         file_name = f"{folder_name}/fichero{j}.txt"
9         with open(file_name, 'w') as file:
10             file.write(f"Este es el contenido del fichero {j}")
```

```
/bin/python3 /home/jjesus/Documentos/tarea3.py
jjesus@jjesus-VirtualBox:~$ /bin/python3 /home/jjesus/Documentos/tarea3.py
Traceback (most recent call last):
  File "/home/jjesus/Documentos/tarea3.py", line 9, in <module>
    with open(file_name, 'w') as file:
ValueError: invalid mode: 'W'
jjesus@jjesus-VirtualBox:~$ /bin/python3 /home/jjesus/Documentos/tarea3.py
jjesus@jjesus-VirtualBox:~$ /bin/python3 /home/jjesus/Documentos/tarea3.py
jjesus@jjesus-VirtualBox:~$
```

```
1 Este es el contenido del fichero 3
```

```
/bin/python3 /home/jjesus/Documentos/tarea3.py
jjesus@jjesus-VirtualBox:~$ /bin/python3 /home/jjesus/Documentos/tarea3.py
Traceback (most recent call last):
  File "/home/jjesus/Documentos/tarea3.py", line 9, in <module>
    with open(file_name, 'w') as file:
ValueError: invalid mode: 'W'
jjesus@jjesus-VirtualBox:~$ /bin/python3 /home/jjesus/Documentos/tarea3.py
jjesus@jjesus-VirtualBox:~$ /bin/python3 /home/jjesus/Documentos/tarea3.py
jjesus@jjesus-VirtualBox:~$
```

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/<nombre-usuario>/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

Primero hemos configurado el sistema de logging (especificando la ruta en filename) y luego hemos obtenido información sobre la partición raíz.

Luego se calcula el porcentaje de espacio ocupado en la partición raíz y luego registra un mensaje de logging en el archivo especificado según los criterios establecidos (error para más del 80%, advertencia para más del 60% y menos del 80%, e información para menos del 60%).

Help

open.py tarea2.py tarea3.py tarea4.py x espacio.log

```
Documentos > tarea4.py > ...
1 import logging
2 import shutil
3
4 logging.basicConfig(filename="/home/jjesus/logs/espacio.log", level=logging.INFO,
5 | | | | | format="%(asctime)s- %(levelname)s: %(message)s")
6
7 total, used, free = shutil.disk_usage("/")
8
9 percent_used = (used / total) * 100
10
11 if percent_used > 80:
12 | logging.error(f"Espacio ocupado en la particion raiz: {percent_used:.2f}%")
13 elif percent_used > 60:
14 | logging.warning(f"Espacio ocupado en la particion raiz: {percent_used:.2f}%")
15 else:
16 | logging.info(f"Espacio ocupado en la particion raiz: {percent_used:.2f}%")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● jjesus@jjesus-VirtualBox:~\$ /bin/python3 /home/jjesus/Documentos/tarea4.py
○ jjesus@jjesus-VirtualBox:~\$

bash
Python

espacio.log - Jjesus - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

- jjesus
 - .vscode
 - Descargas
 - Documentos
 - tarea2.py
 - tarea3.py
 - tarea4.py
 - Escritorio
 - folder1
 - folder2
 - folder3
 - folder4
 - folder5
 - fichero1.txt
 - fichero2.txt
 - fichero3.txt
 - fichero4.txt
 - fichero5.txt
 - fichero6.txt
 - fichero7.txt
 - fichero8.txt
 - fichero9.txt
 - fichero10.txt
 - Imágenes
 - logs
 - espacio.log

logs > espacio.log

```
1 2024-02-24 15:37:58,158- WARNING: Espacio ocupado en la particion raiz: 78.46%
2
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● jjesus@jjesus-VirtualBox:~\$ /bin/python3 /home/jjesus/Documentos/tarea4.py
○ jjesus@jjesus-VirtualBox:~\$

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.

Mosificamos el archivo *tarea4.py* introduciendo la funcion

```
Documentos > tarea4.py > ...
1 import logging
2 import shutil
3
4 def analizar_espacio():
5
6     logging.basicConfig(filename="/home/jjesus/logs/espacio.log", level=logging.INFO,
7                         format="%(asctime)s- %(levelname)s: %(message)s")
8
9     total, used, free = shutil.disk_usage("/")
10
11     percent_used = (used / total) * 100
12
13     if percent_used > 80:
14         logging.error(f"Espacio ocupado en la particion raiz: {percent_used:.2f}%")
15     elif percent_used > 60:
16         logging.warning(f"Espacio ocupado en la particion raiz: {percent_used:.2f}%")
17     else:
18         logging.info(f"Espacio ocupado en la particion raiz: {percent_used:.2f}%")
19
20 if __name__ == "__main__":
21     analizar_espacio()
```

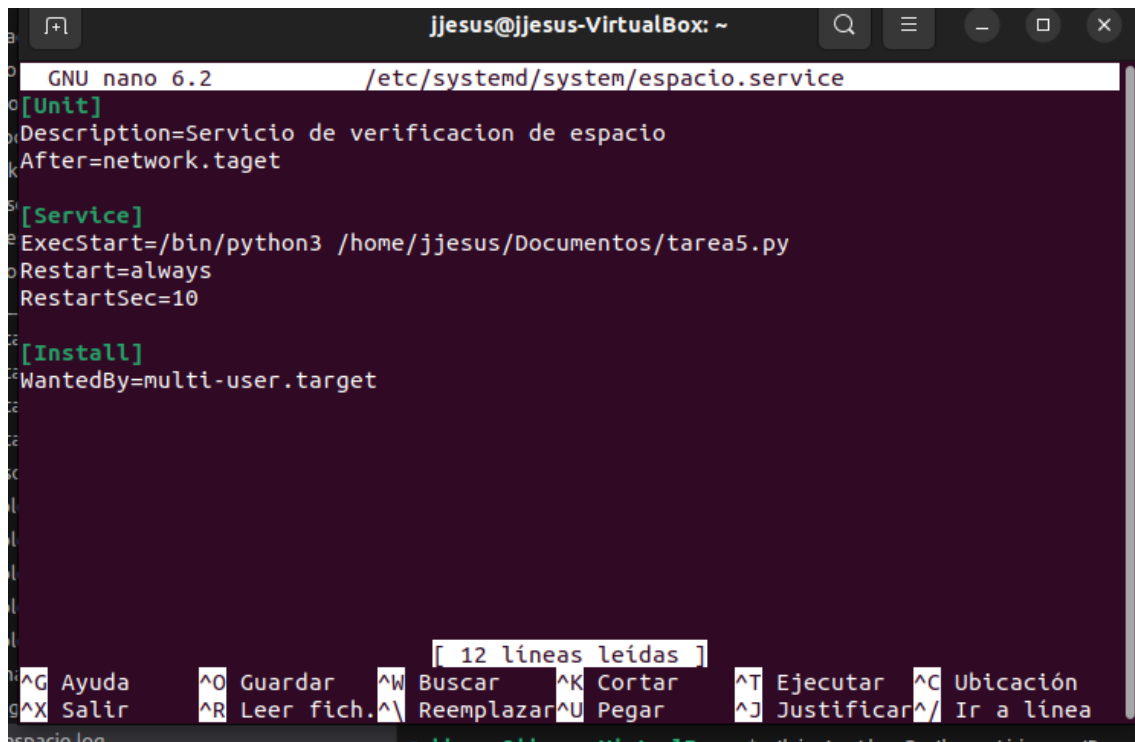
Creamos el archivo *tarea 5.py* con la siguiente configuracion; donde llamamos a la funcion *analizar_espacio()* desde *tarea 5.py* tras importarla desde *tarea4.py*

```
open.py  tarea2.py  tarea3.py  tarea4.py  tarea5.py x
Documentos > tarea5.py
1 from tarea4 import analizar_espacio
2
3 analizar_espacio()
4

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● jjesus@jjesus-VirtualBox:~$ /bin/python3 /home/jjesus/Documentos/tarea5.py
○ jjesus@jjesus-VirtualBox:~$
```

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

Para crear un servicio que llame al archivo *tarea5.py* cada 10 segundos, necesitarás crear un archivo de servicio en systemd. Creamos un archivo llamado *espacio.service* con privilegios de superusuario en el directorio */etc/systemd/system/*.



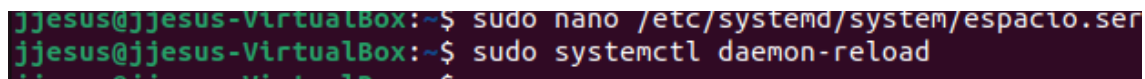
```
GNU nano 6.2 /etc/systemd/system/espacio.service
[Unit]
Description=Servicio de verificacion de espacio
After=network.target

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

[Install]
WantedBy=multi-user.target

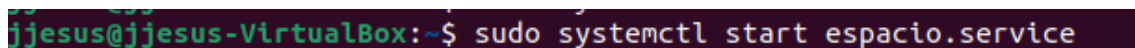
[ 12 líneas leídas ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación
^X Salir  ^R Leer fich. ^_ Reemplazar ^U Pegar  ^J Justificar ^_/ Ir a línea
```

Recargamos los archivos de systemd para que reconozca el nuevo servicio:

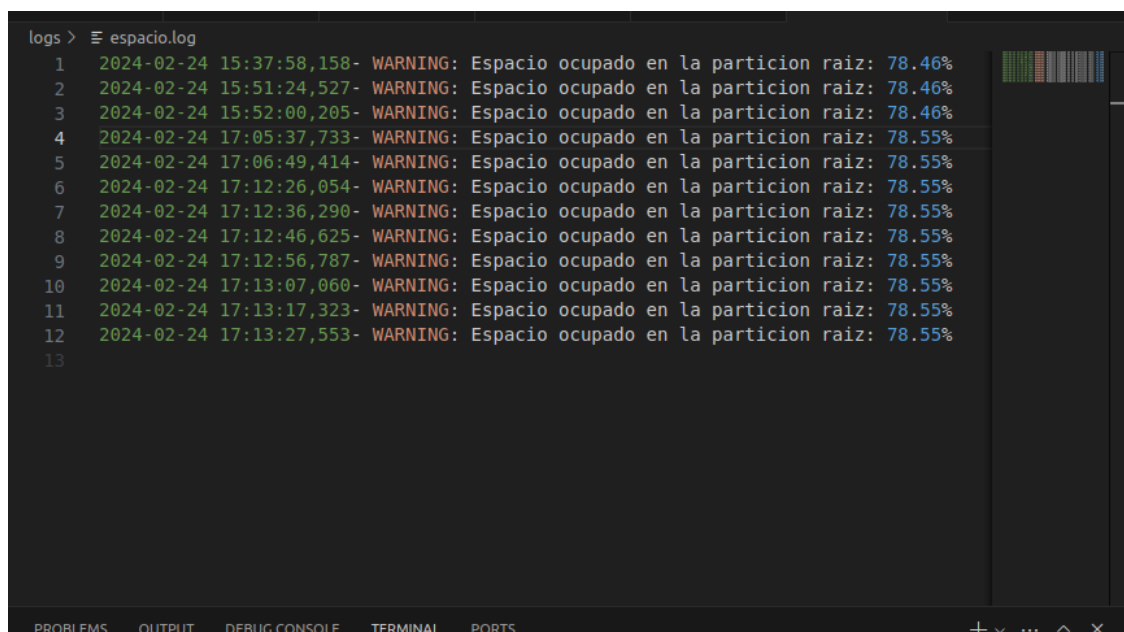


```
jjesus@jjesus-VirtualBox:~$ sudo nano /etc/systemd/system/espacio.service
jjesus@jjesus-VirtualBox:~$ sudo systemctl daemon-reload
jjesus@jjesus-VirtualBox:~$
```

Iniciamos el servicio *espacio.service*, pero la acción la realiza en el visual básico, en el archivo *espacio.log*



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



```
logs > espacio.log
1 2024-02-24 15:37:58,158- WARNING: Espacio ocupado en la particion raiz: 78.46%
2 2024-02-24 15:51:24,527- WARNING: Espacio ocupado en la particion raiz: 78.46%
3 2024-02-24 15:52:00,205- WARNING: Espacio ocupado en la particion raiz: 78.46%
4 2024-02-24 17:05:37,733- WARNING: Espacio ocupado en la particion raiz: 78.55%
5 2024-02-24 17:06:49,414- WARNING: Espacio ocupado en la particion raiz: 78.55%
6 2024-02-24 17:12:26,054- WARNING: Espacio ocupado en la particion raiz: 78.55%
7 2024-02-24 17:12:36,290- WARNING: Espacio ocupado en la particion raiz: 78.55%
8 2024-02-24 17:12:46,625- WARNING: Espacio ocupado en la particion raiz: 78.55%
9 2024-02-24 17:12:56,787- WARNING: Espacio ocupado en la particion raiz: 78.55%
10 2024-02-24 17:13:07,060- WARNING: Espacio ocupado en la particion raiz: 78.55%
11 2024-02-24 17:13:17,323- WARNING: Espacio ocupado en la particion raiz: 78.55%
12 2024-02-24 17:13:27,553- WARNING: Espacio ocupado en la particion raiz: 78.55%
13
```

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

ssh-ed25519

AAAAC3NzaC1lZDI1NTE5AAAAIOP+jPVj13h6gmYJbflcZlIpD7L3hrHD+Aeq75+DVYx

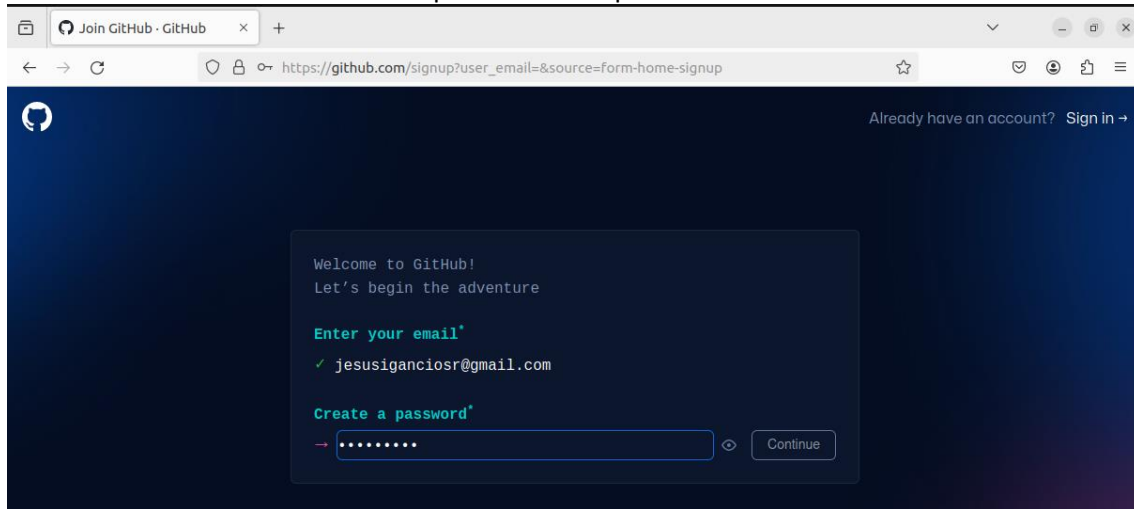
5 ies.fernandosanchez@gmail.com

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

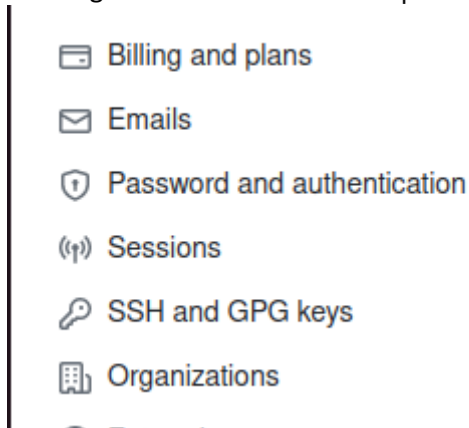
Añade tus dos repositorios a la propia entrega del AulaVirtual.

Github

Nos creamos una cuenta en Github para crear el repositorio



Accedemos y hacemos clic en la foto de perfil en la esquina superior derecha y seleccionamos "Settings". En el menú de la izquierda, haz clic en "SSH and GPG keys".



Hacemos clic 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 tu clave y por último hacemos clic en "Add SSH key".

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Add new SSH Key

Title

Mi clave de prueba

Key type

Authentication Key

Key

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADOC7dyoWdp1sLYqD9/BAVedD9AYkPVAEi/c2xjqnghiJqefHnn0sNkWGm4IZTW1HoCialN9amv5N4QX5pRNx/Xy/tenlw1/QTomEN+GYELmU9A8ub9KXOSU6lulvqA9xnXXZDZnjIX9CR1h5XKz79RyPot7U0j6780FH/oVPR2QfixDdWZU2oJvmoYQXzw749HFmy0n0B/16pr9uiY2wzhYxUvyDcL+IQRz0wwQ8hCyVObvqh2K2jmD9ONpCgJS1R8AWHxePQwaC2/UdXe+JL0/8nCJJRWAZsRjMu0+L3xFdZrBKyB5/o6hN4XJSFy7NlnWy/OdlgcBqvG6J857Rtdhj/WF5Z6jqoNwNDcxatV/JNSpBCE/3EQxiHwTCAXngIT752YMVk1VyijEg+IH2dcYIVJ9DitBvKA9QV2rcUEjeQmYPPDiJQCoas1eVZW9jIVD/WeYkOCy7i7JCM6TFv/9WnQPX8oLnWfFiq0Xq/pqv/YHC05cJ1yvid4Y2G3XSNVvS+eBDUJ8cnJMFZ1zA2Bhevb4vplwPLaRFRF42wMeXLh+PplVOTR9jPbZpdldcuPbZOrt/wiFhz7a/


Add SSH key

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys



Mi clave de prueba

SHA256: aQ6C/RLEiNAK/i+nNKKIsoSAgNHHF8rVZAmj+9okDKu

Added on Feb 24, 2024

Never used — Read/write

Delete

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

Procedemos a añadir la de Fernando siguiendo los mismos pasos que en el caso anterior

jesusignaciosr (jesusignaciosr)

Your personal account

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Add new SSH Key

Title

PERSONAL KEY FERNANDO

Key type

Authentication Key

Key

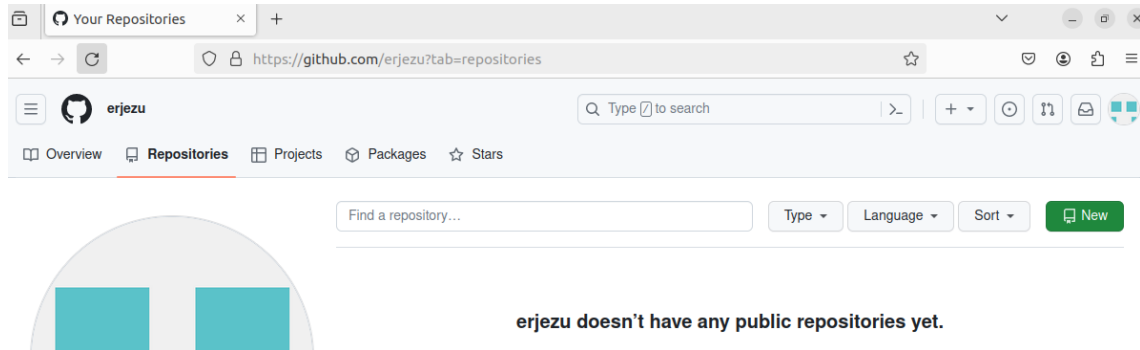
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOP+jPVj13h6gmYJbflcZllpD7L3hrHD+Aeq75+DVYx5ies.fernandosanchez@gmail.com

Add SSH key

Sin embargo en este caso nos da el siguiente error

Key is invalid. You must supply a key in OpenSSH public key format

Ahora vamos a crear un repositorio para subir los documentos. Clicamos en el boton verde donde pone New, en la pestaña de repositorios, para hacer uno nuevo



Ingresamos un nombre una descripcion (opcional), elegir si queremos que el repositorio sea público o privado. Marcar la casilla "Initialize this repository with a README" si deseas que GitHub cree un archivo README.md inicial para tu repositorio. Hacemos clic en el botón "Create repository".

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * **Repository name ***

erjezu / ASO

ASO is available.

Great repository names are short and memorable. Need inspiration? How about **congenial-spoon** ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Create repository

Para poder subir documentos, primero clonamos el repositorio en la máquina local. Abrimos la terminal e instalamos git en nuestro equipo.

```
sudo: git: orden no encontrada
jjesus@jjesus-VirtualBox:~$ sudo apt install git
```

E introducimos el siguiente comando que seria *git clone* y la URL de nuestro repositorio

```
jjesus@jjesus-VirtualBox:~/Documentos$ git clone https://github.com/erjezu/ASO2
Clonando en 'ASO2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Recibiendo objetos: 100% (3/3), listo.
```

Toca olocar los documentos que deseamos subir dentro del directorio del repositorio que acabamos de clonar en tu máquina local. Primero nos vamos al directorio del proyecto con el siguiente comando:

```
jjesus@jjesus-VirtualBox:~/Documentos$ git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama pr
edeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para
usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda:
ayuda: git config --global init.defaultBranch <nombre>
ayuda:
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk'
y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada median
te este comando:
ayuda:
ayuda: git branch -m <nombre>
Iniciado repositorio Git vacío en /home/jjesus/Documentos/.git/
```

Ahora procedemos a añadir los documentos

```
Iniciado repositorio Git vacío en /home/jjesus/Documentos/.git/
jjesus@jjesus-VirtualBox:~/Documentos$ git add tarea2.py
jjesus@jjesus-VirtualBox:~/Documentos$ git add tarea3.py
jjesus@jjesus-VirtualBox:~/Documentos$ git add tarea4.py
jjesus@jjesus-VirtualBox:~/Documentos$ git add tarea5.py
```

Hacemos una confirmación de los cambios utilizando el siguiente comando:

```
jjesus@jjesus-VirtualBox:~/Documentos$ git commit -m "nuevo"
[master (commit-raíz) 1b6b888] nuevo
4 files changed, 53 insertions(+)
create mode 100644 tarea2.py
create mode 100644 tarea3.py
create mode 100644 tarea4.py
create mode 100755 tarea5.py
```

Eelgimos la rama del repositorio hacia la que subir los arxhivos

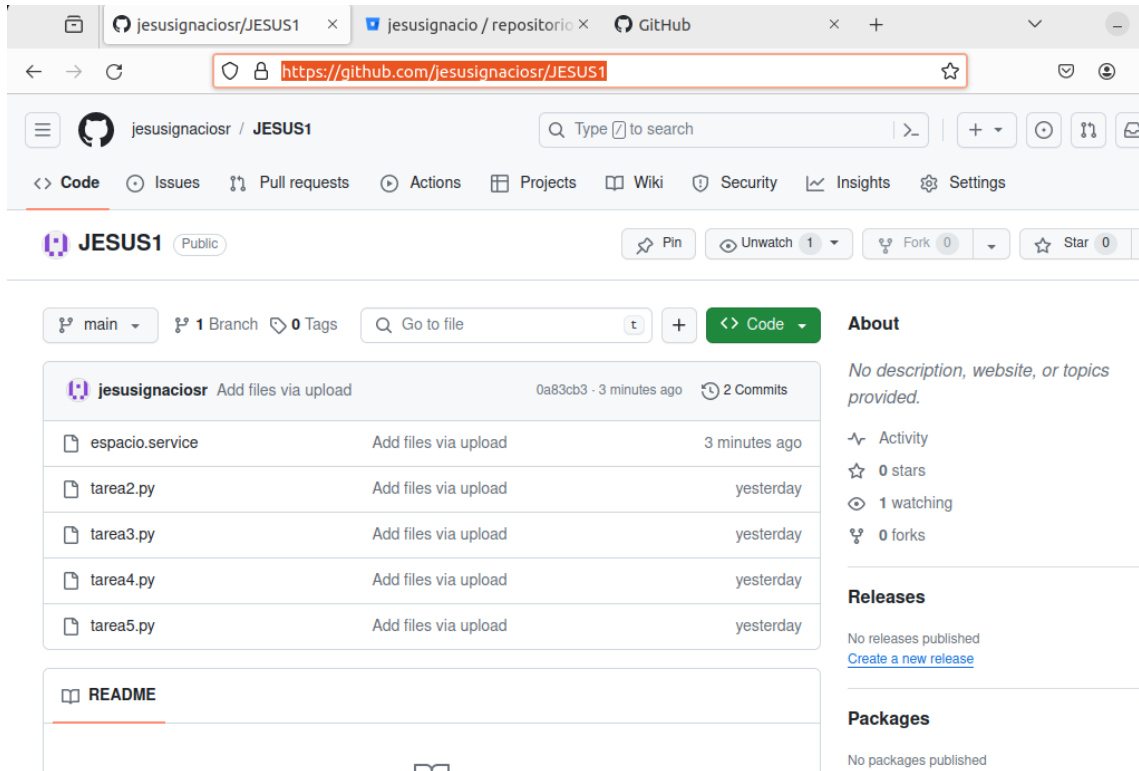
```
create mode 100755 tarea5.py
jjesus@jjesus-VirtualBox:~/Documentos$ git branch -M main
```

```
jjesus@jjesus-VirtualBox:~/Documentos$ git remote add origin https://github.com/
erjezu/ASO/pulls
```

Por ultimo subimos los archivos al repositorio reomoto con este comando

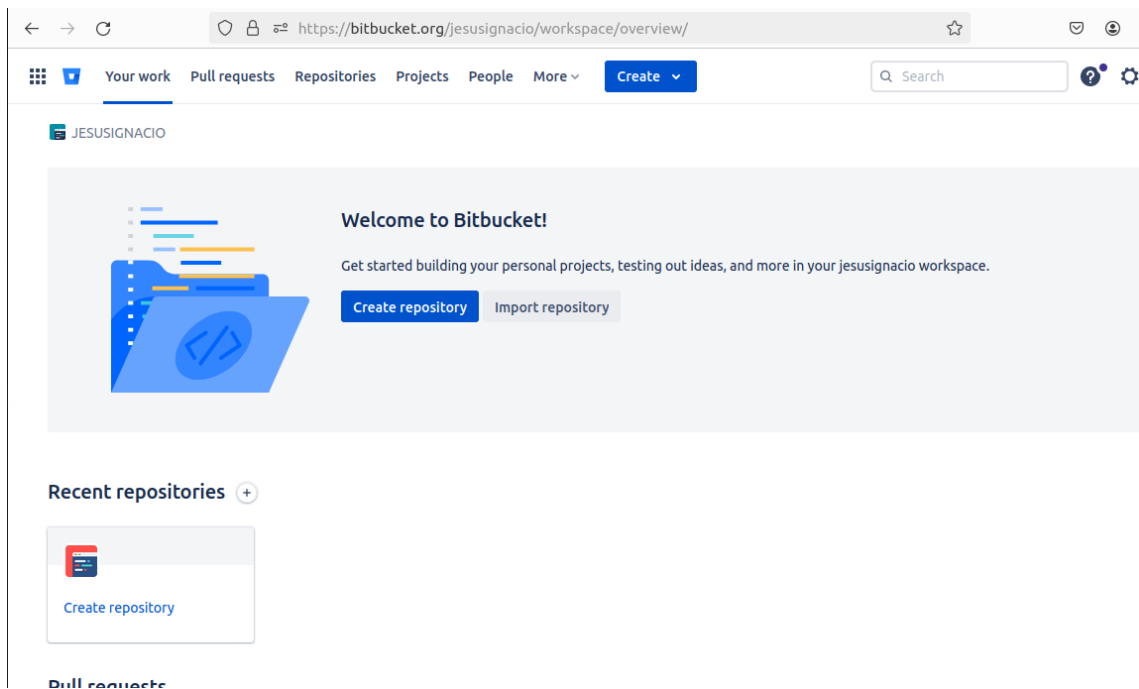
```
fatal: Authentication failed for https://github.com:jesusignaciosr/JESUS1$ git push origin main
```

Ante los problemas que me estaba dando este repositorio para subir los archivos he creado uno nuevo llamado JESUS1, que me ha dejado subir los archivos



Bitbucket

Como en el caso anterior nos creamos una cuenta en Bitbucket



Accedemos y hacemos clic en la foto de perfil en la esquina superior derecha y seleccionamos "Settings". En el menú de la izquierda, haz clic en "SSH and GPG keys".

SECURITY

SSH keys

Two-step verification

Sessions

Audit log

SSH keys

Use SSH to avoid password prom

Add key

Key

Add SSH key

Label

PERSONAL KEY

Key*

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDOC7dYoWdp1sLYqD9/
BAVedD9AYkPVAEf/
c2xjqnohiJqeFHnn0sNkWGm4tZTW1HoCiaIN9amv5N4QX5pRNx/Xy/tenlw1/
QTomEN+GYELmU9A8ub9KXOSU6IulvqA9xnXXZDZnjIX9CR1h5XKz79RyPot
7U0j6780FH/oVPR2QflxDdWZU2oJvm0yQXzw749HFmy0n0B/
16pr9uiY2wzhYxUvyDcL+lQRz0wwQ8hCyVQbvqh2K2jmD9ONpCgJS1R8AWh
XePQwaC2/UdXe+JL0/8nCJJRWAZsRjMu0+L3xFdZrBKyB5/
o6hN4XJSFy7NlnWy/OdlgcBqvG6J857Rtdhj/WF5Z6jqoNwNDcxatV/

Don't have a key?

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

Already have a key?

Copy and paste your key here with `cat ~/.ssh/id_rsa.pub | xclip -sel clip`.

Problems adding a key?

Add key

Cancel

SSH keys

Use SSH to avoid password prompts when you push code to Bitbucket. Learn how to [generate an SSH key](#).


Add key

Key	Added	Last used
PERSONAL KEY	just now	Never

Para pasar los docuemntos al repositorio primero creamos un repositorio wn la maquina

Create a new repository


[Import repository](#)

Workspace  JESUSIGNACIO


Project name*

Repository name*

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

Include .gitignore? Yes (recommended) 

[Advanced settings](#)

Create repository

Cancel

El siguiente paso seria hacer una clonacion del repositorio en la terminal de Ubuntu, pero a la hora de introducir la clave me da problemas a pesar de que ingreso la correcta

Las URLs de los repositorios son:

<https://github.com/jesusignaciosr/JESUS1>

<https://bitbucket.org/jesusignacio/repositorio1/src/main/>