



Nombre y apellidos: Ayoub Bouhachami Lazaar

Fecha: 24/02/2024

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, <nombre>.pub

Comando: `ssh-keygen -t 25995 -b 4096 -C pruebaposfix@gmail.com`

```
alumno@alumno-VirtualBox:~$ ssh-keygen -t ed25519 -b 4096 -C "pruebaposfix@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/alumno/.ssh/id_ed25519): ayoub2
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ayoub2
Your public key has been saved in ayoub2.pub
The key fingerprint is:
SHA256:70idgI3+gm1EDcsr7ht5P+UIpdhX69X/0l4bEl5fJYM pruebaposfix@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|
| . +      E o . |
| + o..    o. |
| + + .S. . . . |
| o.*+.oo . o o o |
|.o+=o+=.. + .o |
| o++oooo. +.o |
|.oo.o+.o .o=. |
+----[SHA256]-----+
alumno@alumno-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 así:

`/dev/sda1 78,9%`

`/dev/sdb1 18,5%`

```
alumno@alumno-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
alumno@alumno-VirtualBox: ~ x alumno@alumno-VirtualBox: ~ x +
GNU nano 6.2 tarea2.py
import psutil

def obtener_porcentaje_espacio(particion):
    disco = psutil.disk_usage(particion)
    porcentaje_ocupado = disco.percent
    return porcentaje_ocupado

def mostrar_porcentaje_espacio(particiones):
    for particion in particiones:
        porcentaje = obtener_porcentaje_espacio(particion)
        print(f"{particion} {porcentaje:.1f}%")

particiones = ['/dev/sda1', '/dev/sda2', '/dev/sda3']
mostrar_porcentaje_espacio(particiones)
```

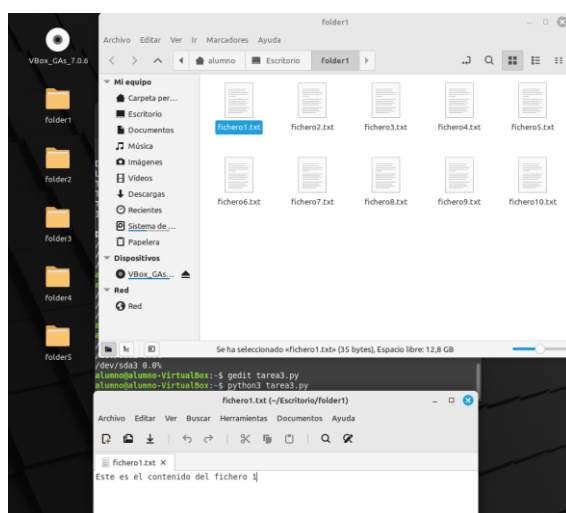
```
alumno@alumno-VirtualBox:~$ python3 tarea2.py
/dev/sda1 0.0%
/dev/sda2 0.0%
/dev/sda3 0.0%
alumno@alumno-VirtualBox:~$
```

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

```
Abrir  +  tarea3.py  Guardar  -  +  x
1 import os
2
3 def crear_carpetas_y_ficheros(ruta_absoluta):
4     for i in range(1, 6):
5         carpeta = os.path.join(ruta_absoluta, f"folder{i}")
6         os.makedirs(carpeta, exist_ok=True)
7         for j in range(1, 11):
8             nombre_fichero = os.path.join(carpeta, f"fichero{j}.txt")
9             with open(nombre_fichero, 'w') as f:
10                 f.write("Este es el contenido del fichero 1\n")
11
12 ruta_absoluta = "/home/alumno/Esitorio"
13
14 crear_carpetas_y_ficheros(ruta_absoluta)
15
16
```



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

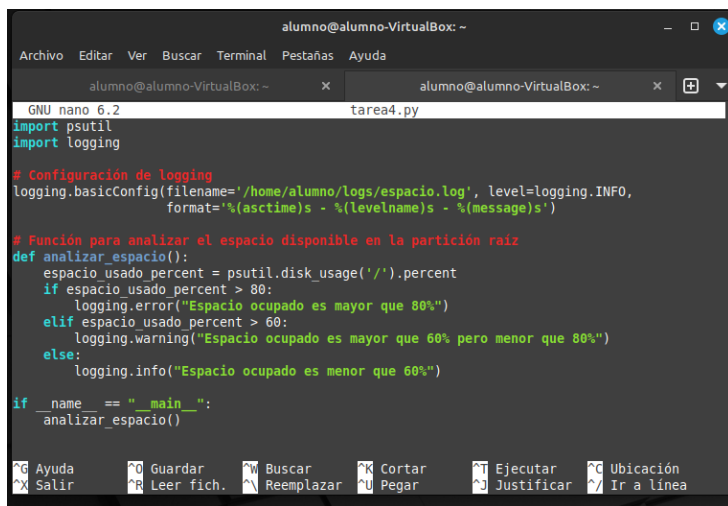
Si el espacio ocupado es mayor que 80% se usara un mensaje de error.

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

```
alumno@alumno-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda

alumno@alumno-VirtualBox: ~ x alumno@alumno-VirtualBox: ~ + - x
alumno@alumno-VirtualBox:~$ mkdir logs
alumno@alumno-VirtualBox:~$ rm log
rm: no se puede borrar 'log': Es un directorio
alumno@alumno-VirtualBox:~$ rmdir log
alumno@alumno-VirtualBox:~$ cd logs/
alumno@alumno-VirtualBox:~/logs$ touch espacio.log
alumno@alumno-VirtualBox:~/logs$ cd
alumno@alumno-VirtualBox:~$
```



The screenshot shows a terminal window titled 'alumno@alumno-VirtualBox: ~'. The nano editor is open with a file named 'tarea4.py'. The code in the file is as follows:

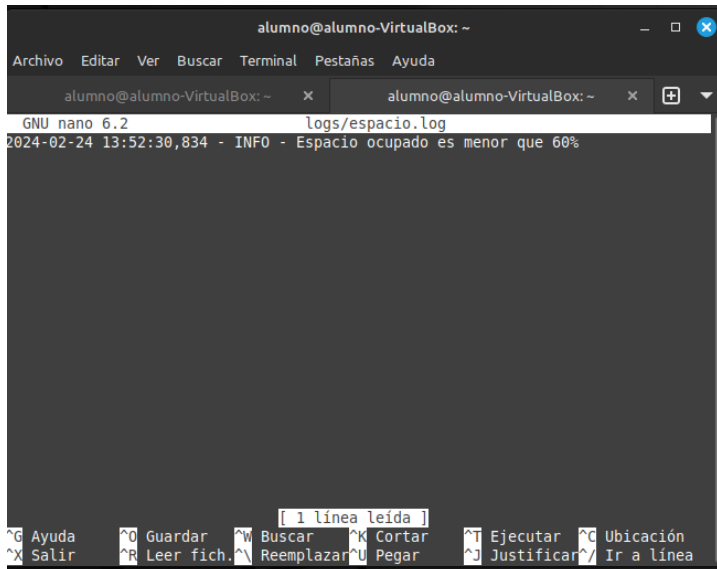
```
GNU nano 6.2 tarea4.py
import psutil
import logging

# Configuración de logging
logging.basicConfig(filename='/home/alumno/logs/espacio.log', level=logging.INFO,
                    format='%(asctime)s - %(levelname)s - %(message)s')

# Función para analizar el espacio disponible en la partición raíz
def analizar_espacio():
    espacio_usado_percent = psutil.disk_usage('/').percent
    if espacio_usado_percent > 80:
        logging.error("Espacio ocupado es mayor que 80%")
    elif espacio_usado_percent > 60:
        logging.warning("Espacio ocupado es mayor que 60% pero menor que 80%")
    else:
        logging.info("Espacio ocupado es menor que 60%")

if __name__ == "__main__":
    analizar_espacio()
```

The bottom of the window shows a status bar with keyboard shortcuts: ^G Ayuda, ^X Salir, ^O Guardar, ^R Leer fich., ^W Buscar, ^\ Reemplazar, ^K Cortar, ^U Pegar, ^T Ejecutar, ^C Ubicación, ^J Justificar, ^_ Ir a línea.

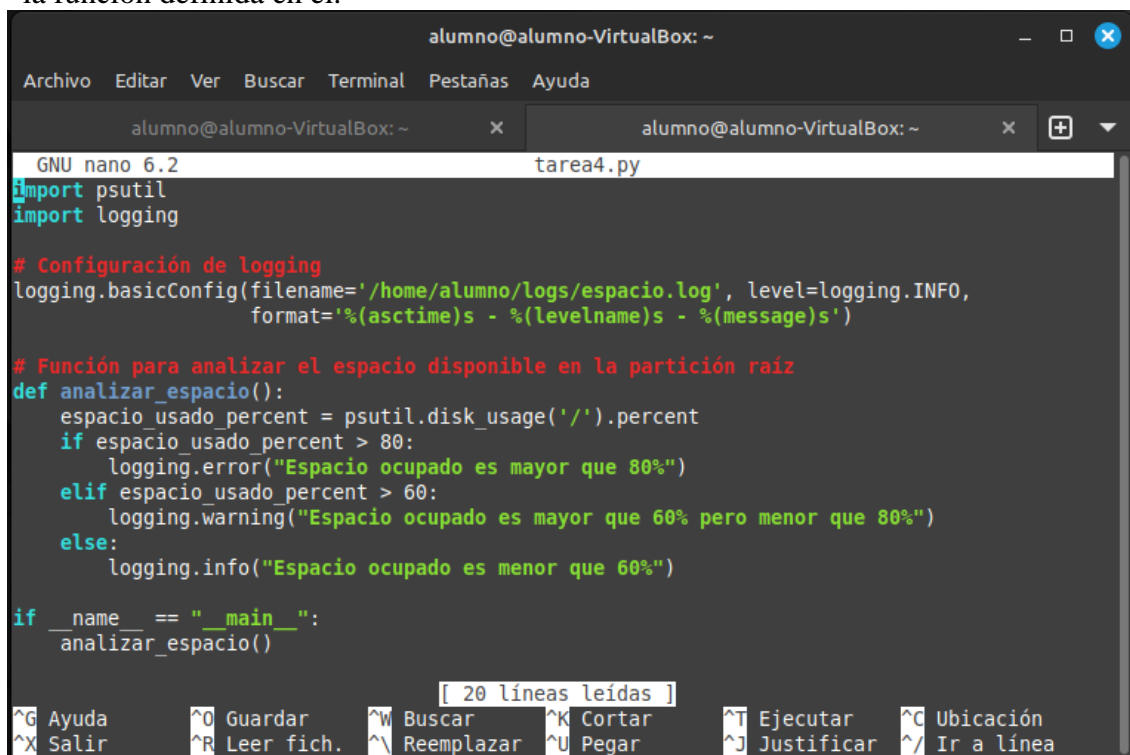


The screenshot shows the same terminal window, but now the nano editor is open with a file named 'logs/espacio.log'. The content of the file is:

```
GNU nano 6.2 logs/espacio.log
2024-02-24 13:52:30,834 - INFO - Espacio ocupado es menor que 60%
```

The status bar at the bottom shows: ^G Ayuda, ^X Salir, ^O Guardar, ^R Leer fich., ^W Buscar, ^\ Reemplazar, ^K Cortar, ^U Pegar, ^T Ejecutar, ^C Ubicación, ^J Justificar, ^_ Ir a línea. A status message '[1 línea leída]' is visible above the status bar.

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.



The screenshot shows the same terminal window, with the nano editor open to 'tarea4.py'. The code is identical to the first screenshot:

```
GNU nano 6.2 tarea4.py
import psutil
import logging

# Configuración de logging
logging.basicConfig(filename='/home/alumno/logs/espacio.log', level=logging.INFO,
                    format='%(asctime)s - %(levelname)s - %(message)s')

# Función para analizar el espacio disponible en la partición raíz
def analizar_espacio():
    espacio_usado_percent = psutil.disk_usage('/').percent
    if espacio_usado_percent > 80:
        logging.error("Espacio ocupado es mayor que 80%")
    elif espacio_usado_percent > 60:
        logging.warning("Espacio ocupado es mayor que 60% pero menor que 80%")
    else:
        logging.info("Espacio ocupado es menor que 60%")

if __name__ == "__main__":
    analizar_espacio()
```

The status bar at the bottom shows the same keyboard shortcuts as before. A status message '[20 líneas leídas]' is visible above the status bar.

```
alumno@alumno-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
alumno@alumno-VirtualBox: ~ x alumno@alumno-VirtualBox: ~ x +
GNU nano 6.2 tarea5.py *
from tarea4 import analizar_espacio
# Llamar a la función definida en tarea4.py
analizar_espacio()
```

```
alumno@alumno-VirtualBox: ~$ python3 tarea5.py
alumno@alumno-VirtualBox: ~$ nano logs/espacio.log
alumno@alumno-VirtualBox: ~$
```

```
alumno@alumno-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
alumno@alumno-VirtualBox: ~ x alumno@alumno-VirtualBox: ~ x +
GNU nano 6.2 logs/espacio.log
2024-02-24 13:52:30,834 - INFO - Espacio ocupado es menor que 60%
2024-02-24 14:09:17,440 - INFO - Espacio ocupado es menor que 60%
```

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

```
alumno@alumno-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
alumno@alumno-VirtualBox: ~ x alumno@alumno-VirtualBox: ~ x +
GNU nano 6.2 /etc/systemd/system/espacio.service
[Unit]
Description=Ejecutar tarea5.py cada 10 segundos

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

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

Me falto el comando `sudo systemctl daemon-reload`.

```
alumno@alumno-VirtualBox: ~$ sudo systemctl enable espacio.service
Created symlink /etc/systemd/system/multi-user.target.wants/espacio.service → /etc/systemd/system/espacio.service.
alumno@alumno-VirtualBox: ~$ sudo systemctl start espacio.service
^Z^X^C
alumno@alumno-VirtualBox: ~$ sudo systemctl start espacio.service
```

```

alumno@alumno-VirtualBox:~$ tail -10f /home/alumno/logs/espacio.log
2024-02-24 19:12:42,484 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:13:20,655 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:13:42,490 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:14:04,964 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:14:35,693 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:15:42,486 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:16:10,563 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:16:37,476 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:17:22,754 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:26:27,132 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:26:37,232 - INFO - Espacio ocupado es menor que 60%
2024-02-24 19:26:47,479 - INFO - Espacio ocupado es menor que 60%
^Z
[3]+  Detenido                  tail -10f /home/alumno/logs/espacio.log
alumno@alumno-VirtualBox:~$

```

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

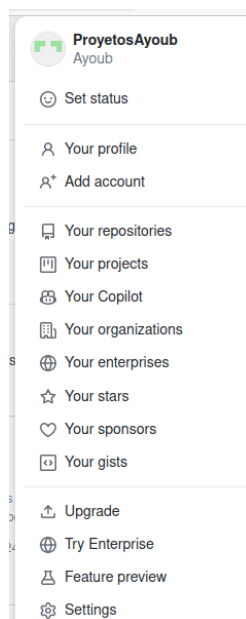
ssh-ed25519

AAAAC3NzaC1lZDI1NTE5AAAAIOP+jPVj13h6gmYJbfIcZlIpD7L3hrHD+Aeq75+DVYx


5 ies.fernandosanchez@gmail.com

En GitHub no podemos poner una contraseña en interfaz gráfica pero por comandos seguramente se pueda así que ese sería la única parte que me faltaría.

- Iremos a la derecha de nuestra pagina y le damos clic en nuestro icono y nos situaremos en Settings.



- Iremos al apartado de SSH and GPG Keys.

**Ayoub (ProyetosAyoub)**
Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

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](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.


Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

☐ Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key. Note that this will include your existing unsigned commits. [Learn about vigilant mode.](#)

- Le ponemos un título para reconocer.

**Ayoub (ProyetosAyoub)**
Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Add new SSH Key

Title

Key type


Authentication Key

Key

Begin with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

- Ponemos la contraseña que nos creo con el primer comando y que esta en un fichero llamado <nombre>.pub.

**Ayoub (ProyetosAyoub)**
Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions


Code, planning, and automation

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



ClavePublicoExamenAso
SHA256:70idgI3+gm1EDcsr7ht5P+UIpdhX69X/014bE15fJYM
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](#).

Bitbucket Your work Pull requests Repositories Projects People More Create

ExamenAso

ProjectosAyoub / ExamenAso / ExamenAso / Repository settings

Access keys



Use [access keys](#) to gain read-only access to this repository. Learn more about using SSH keys.

Add key

GENERAL

- Back
- Repository details
- Repository permissions
- Username aliases

SECURITY

- Access keys
- Access tokens

WORKFLOW

- Branch restrictions

Add SSH key

Label

ClavePublicaExamenASO

Key *

```
ssh-ed25519 |AAAC3NzaC1lZDI1NTE5AAAAIP6l9pq0CzYAzfB/
mrvrjYNVpTRgXyAaUHDmpuh4UPfW
```

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 | pbcopy
```



Problems adding a key?

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

Add SSH key

Cancel

Repositorio 1: <https://github.com/ProyetosAyoub/ExamenASO>

Repositorio 2: <https://bitbucket.org/proyectosayoub/examenaso>

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.