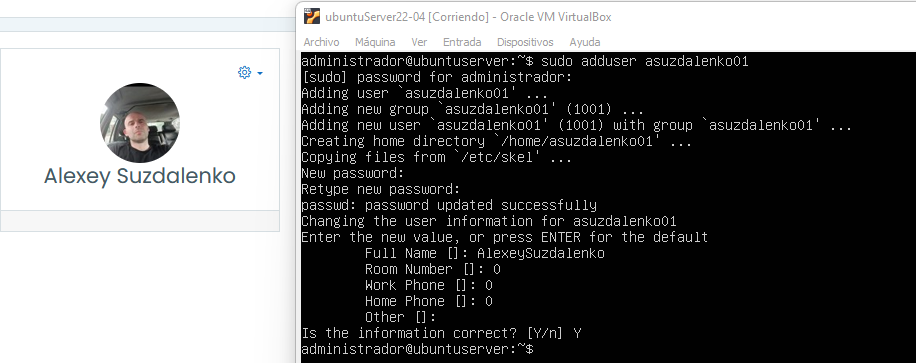
Tarea 6.

Partimos de una MV con Ubuntu Server 22-04

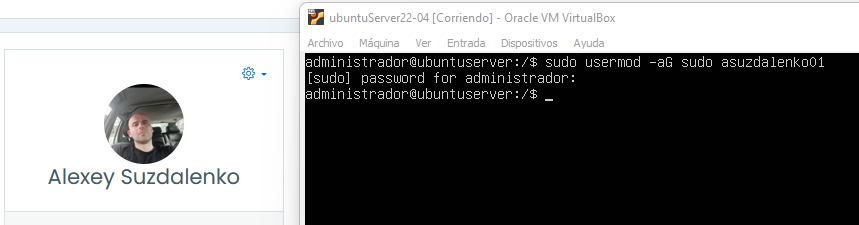
1. Creo mi usuario asuzdalenko01:

**sudo adduser asuzdalenko01**



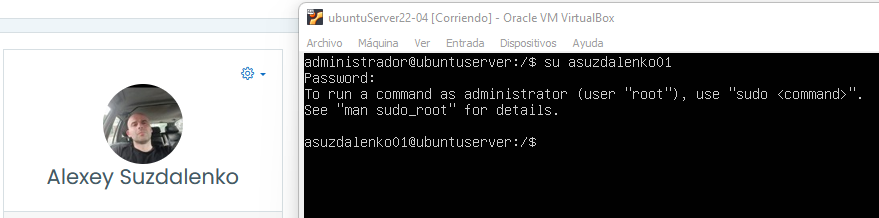
Agrego mi usuario al grupo sudoers:

**sudo usermod -aG sudo asuzdalenko01**



Inicio sesión con el:

**su asuzdalenko01**

****

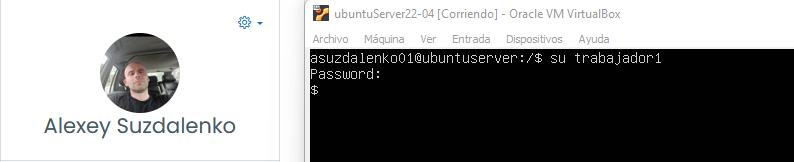
1. Crear usuario trabajador1 con una Shell /bin/sh

**sudo useradd -s /bin/sh trabajador1**

****

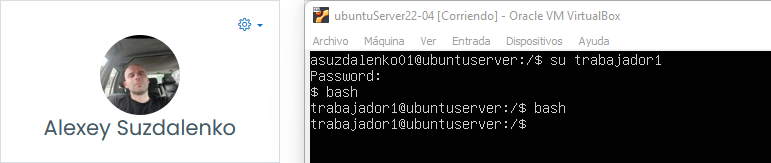
Inicio sesión con usuario trabajador1:

**su trabajador1**

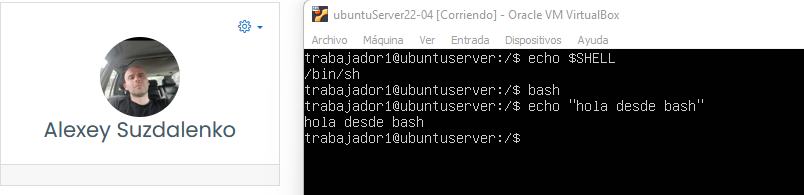
****

Actualmente bash funciona en trabajador1, y puedo comprobarlo:

bash



echo “hola desde bash”

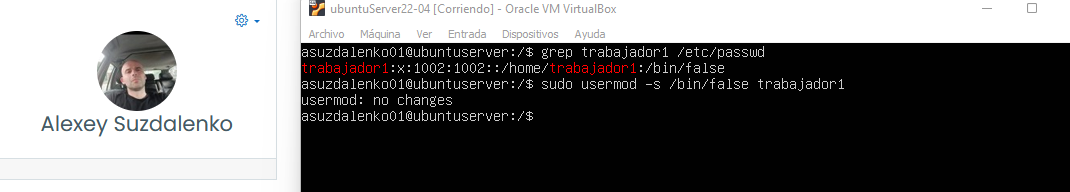


Con el comando

**sudo usermod -s /bin/false trabajador1** desactivo la ejecución de bash

y con **grep trabajador1 /etc/passwd** comprobó que el usuario trabajador1 tiene bash desactivado

las respuesta es :/bin/false -> trabajador1 no puede usar bash



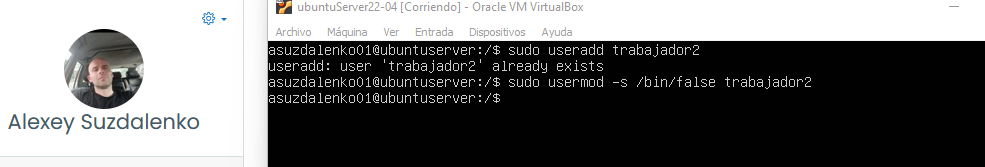
1. Crea el usuario trabajador2 y NO pueda loguearse. Muestra su UID

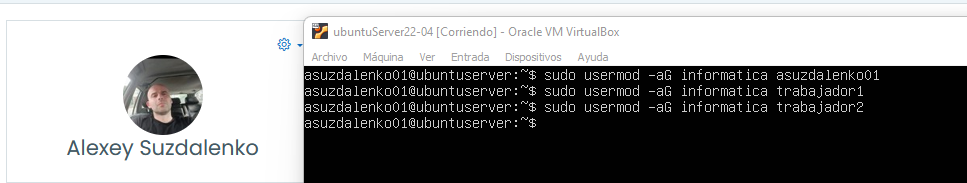
Crear usuario trabajador2

**sudo useradd trabajador2**

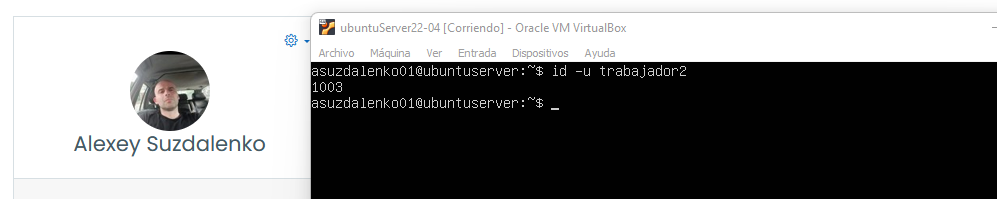
Que no pueda logearse

**sudo usermod -s /bin/false trabajador2**



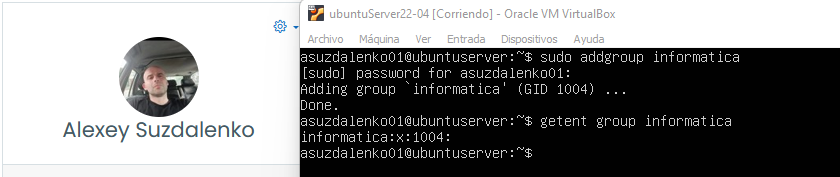
**id -u trabajador2**

Resultado 1003



1. Crear un grupo “informatica”

**sudo addgroup informatica**

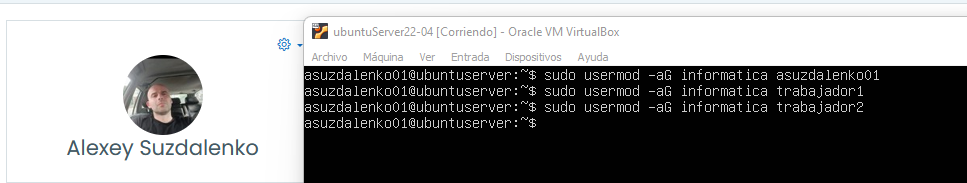
****

Agrego usuarios “asuzdalenko01”, “trabajador1”, “trabajador2” al grupo “informatica”

**sudo usermod -aG informatica asuzdalenko01**

**sudo usermod -aG informatica trabajador1**

**sudo usermod -aG informatica trabajador2**

****

1. Mostrar UID de usuario “asuzdalenko01” y el grupo al que pertenece “informatica”

**id -u asuzdalenko01** (id del usuario “asuzdalenko01)

**id -Gn asuzdalenko**  (mostrar grupo al que pertenece)

**id asuzdalenko01** (id de usuario y de grupo)

****

1. Crea el usuario trabajador3, cuyo “home” sea /usuarios/trabajador3 y pertenezca al grupo informática. Hazlo con un solo comando

**sudo useradd -m -d /usuarios/trabajador3 -G informatica trabajador3**

sudo: Ejecuta el comando con privilegios de superusuario.

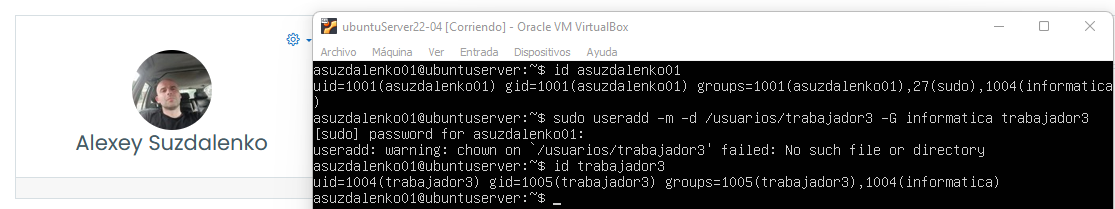
useradd: Comando para agregar un nuevo usuario.

-m: Crea el directorio de inicio si no existe.

-d /usuarios/trabajador3: Especifica el directorio de inicio del usuario.

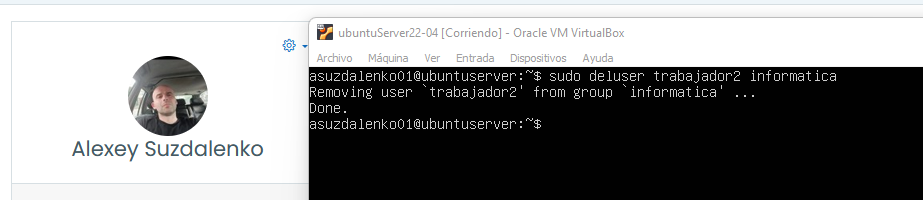
-G informatica: Añade al usuario al grupo "informatica".

trabajador3: Nombre del usuario que estás creando.



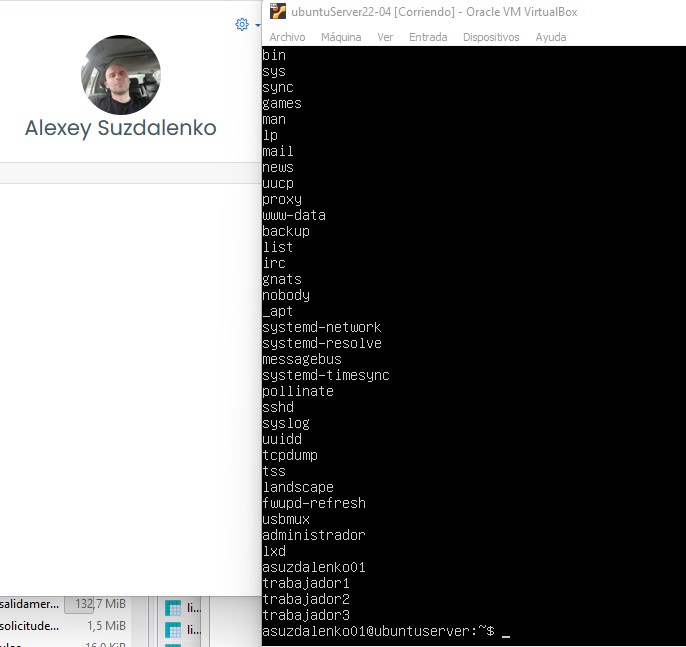
1. Saco a trabajador2 del grupo informatica

**sudo deluser trabajador2 informatica**

****

1. Mostrar todos los usuarios del sistema

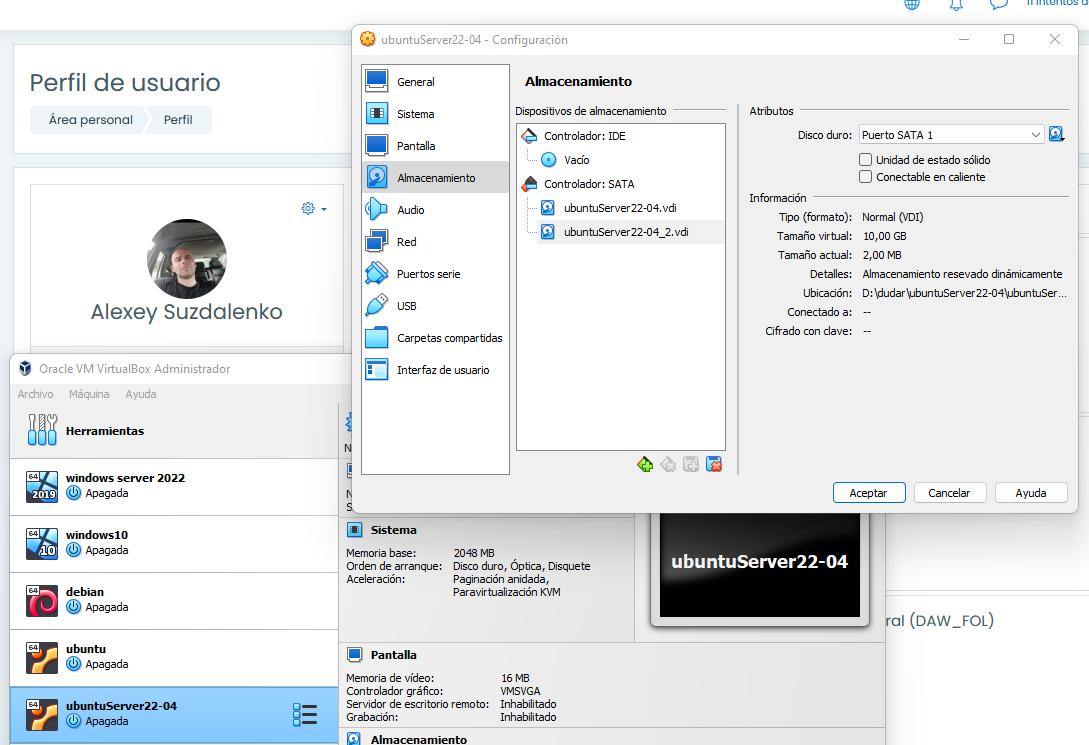
**cut -d: -f1 /etc/passwd**



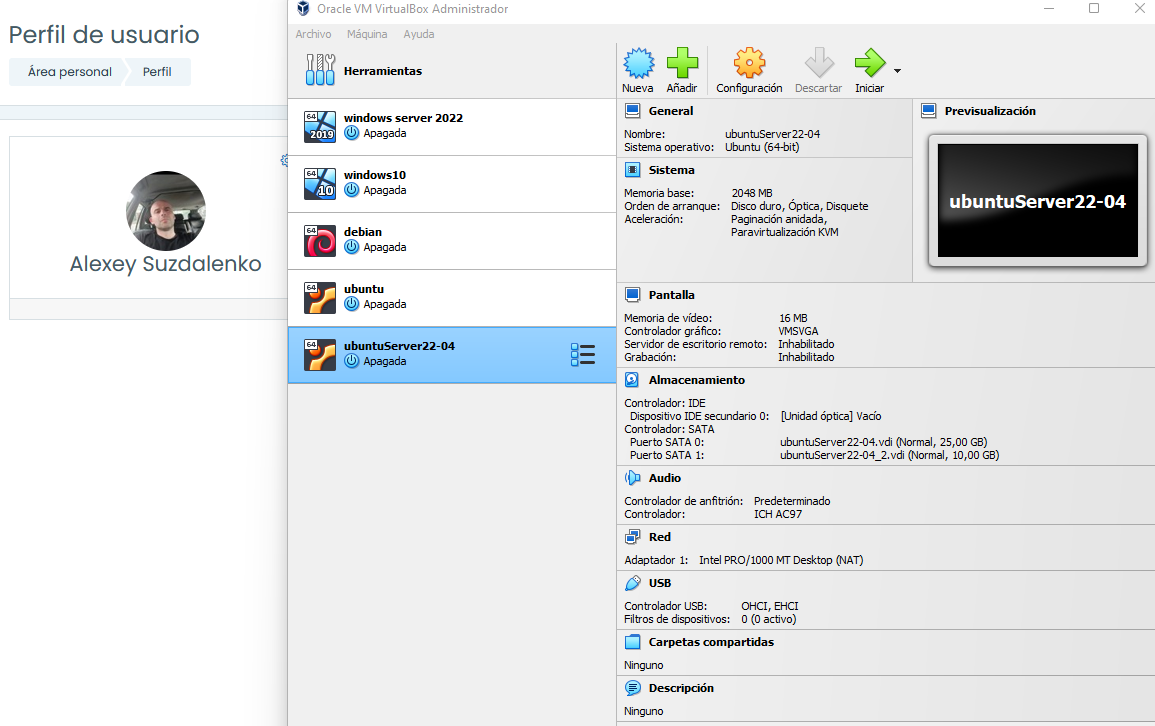
Se muestran todos los usuarios del sistema (incluidos con los que vienen con el sistema y los que no están destinados para el inicio de sesion)

**Actididad 2. FDISK Y FSTAB**

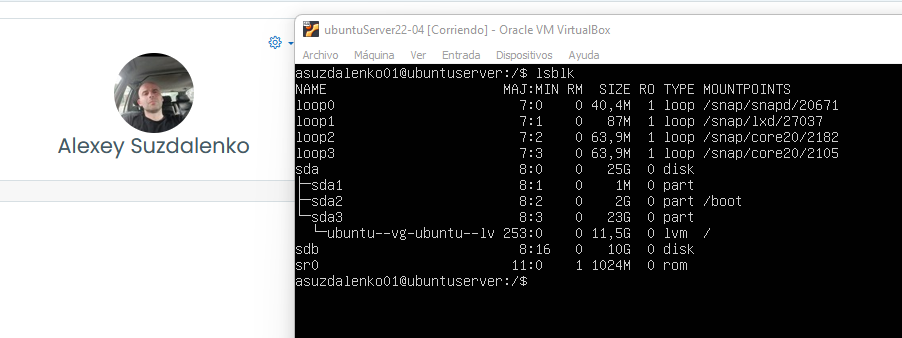
1. Añadimos un segundo disco de 10GB



En la siguiente captura se ve el disco de 10 GB asignado:



2 Para mostrar que discos tengo en le servidor usare **lsblk,** el disco con el que voy a trabajar es “sdb”, por ahora no tiene ninguna partición.

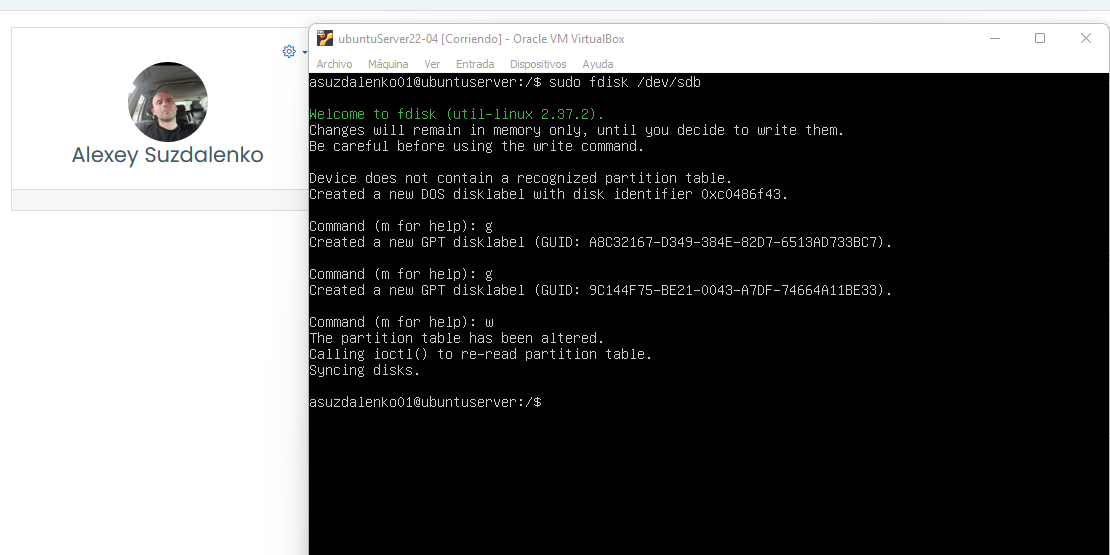


3 crear tabla de las particiones GPT

**sudo fdisk /dev/sdb**

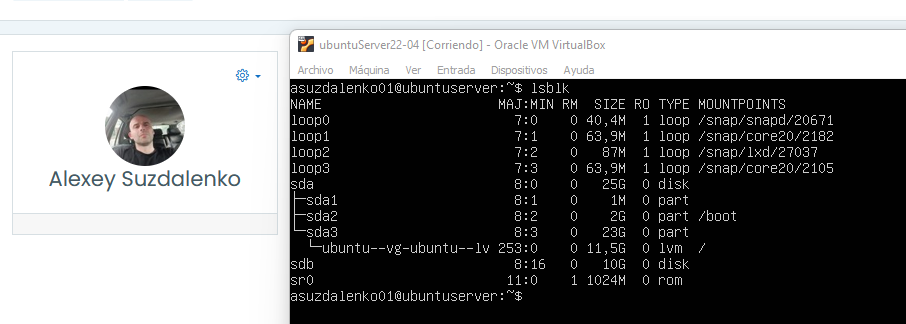
Presiono **g** para cambiar a la tabla de particiones GPT.

Luego, presiona **w** para escribir los cambios.

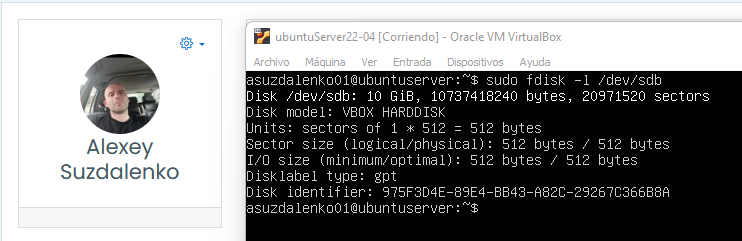
****

**lsblk**

Podemos observar que seguimos teniendo el disco sdb sin ninguna partición por ahora

****

Y **sudo** **fdisk -l /dev/sdb** compruebo el estado del disco sdb

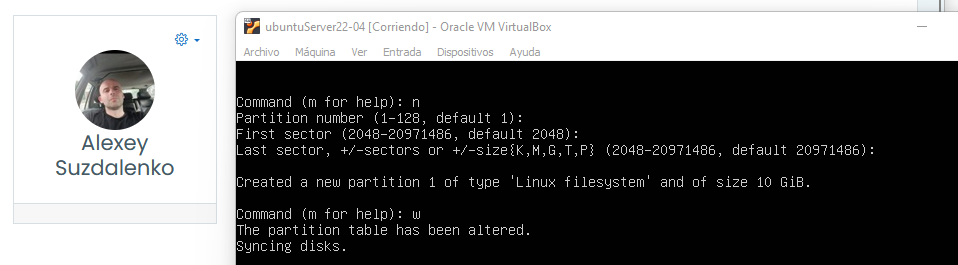


**4** Crearpartición con formato ext4 con todo el tamaño (10 GB)

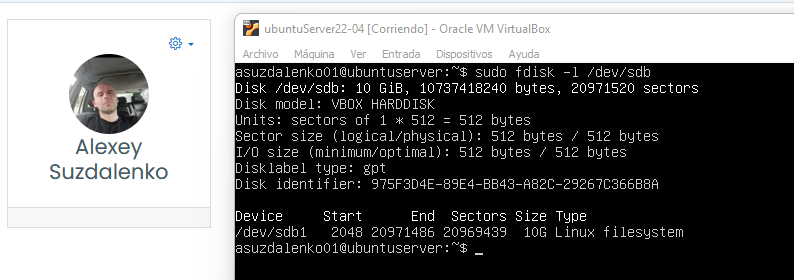
Primero creare una partición **sudo fdisk /dev/sdb**

n – para crear nueva partición

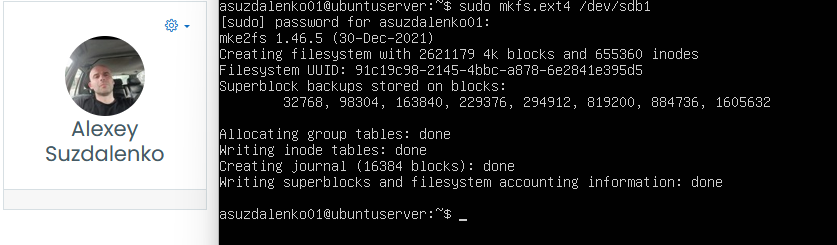
w – para guardar los cambios

****

Compruebo que tengo una partición creada **sudo fdisk -l /dev/sdb**

****

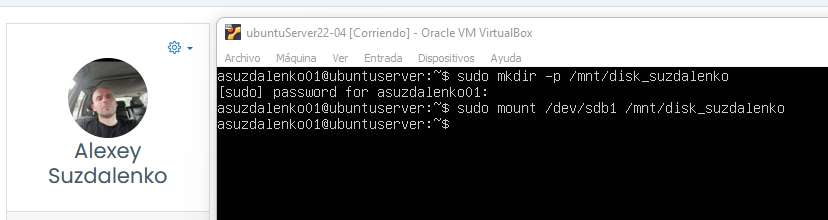
Ahora formateo la partición **sudo mkfs.ext4 /dev/sdb1**

****

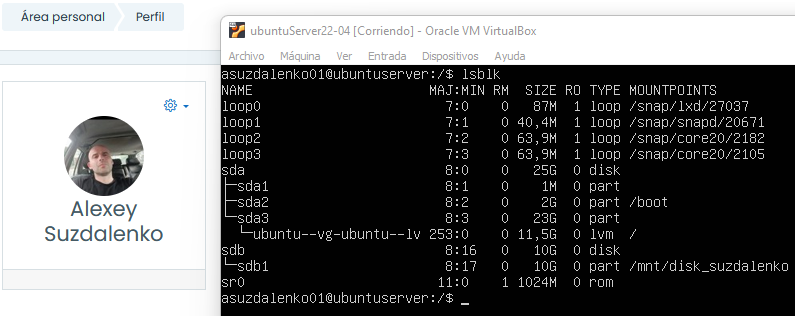
**5.** montamos la partición en la carpeta /mnt/disk\_suzdalenko

Creo carpeta en /mnt/disk\_suzdalenko **sudo mkdir -p /mnt/disk\_suzdalenko**

Y monto el disco **sudo mount /dev/sdb1 /mnt/disk\_suzdalenko**

****

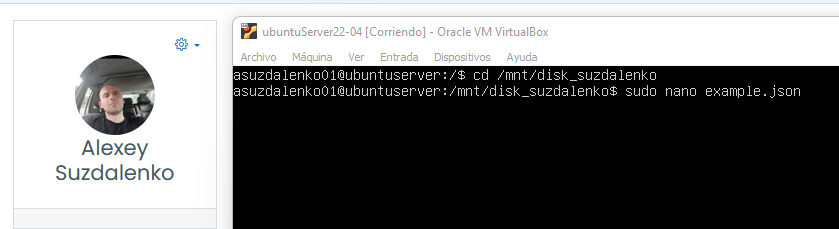
Puedo comprobar que el disco está montado **lsblk**

****

6. Creamos un archivo con nano en la carpeta /mnt/disk\_tuapellido para comprobar que tenemos permisos de escritura

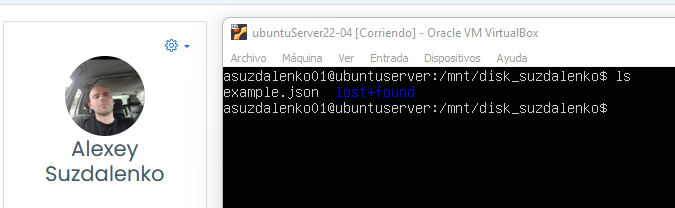
**cd /mnt/disk\_suzdalenko** (vamos a la ruta indicada)

**sudo nano example.json** (creamos archivo example.json)



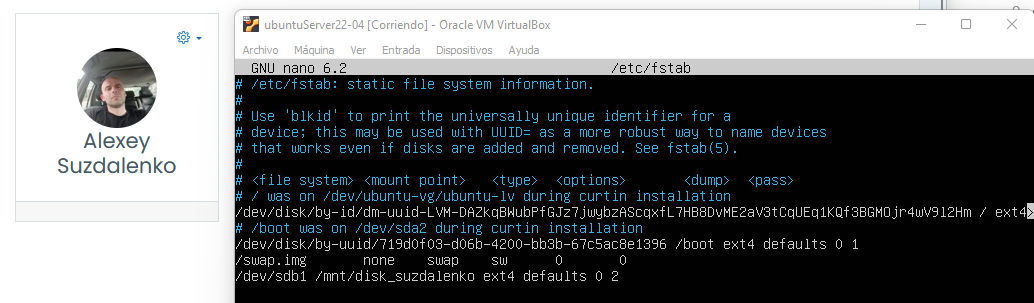
Mostramos contenido de la carpeta para ver que el archivo “example.json” se ha creado

**ls**

****

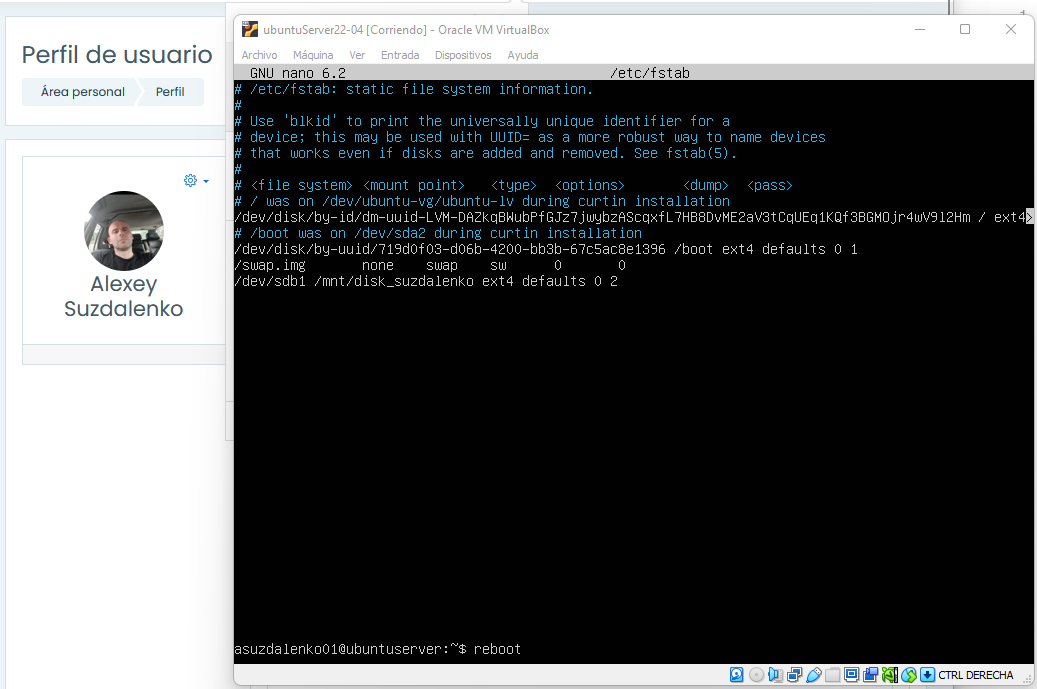
**7.** Editar el fichero fstab para que al reiniciar la máquina el disco se monte de forma automática

**sudo nano /etc/fstab**

****

En este archivo he añadido la ultima linea **/dev/sdb1 /mnt/disk\_suzdalenko ext4 defaults 0 2**, gracias a eso el reiniciar el servidor Ubuntu se montará el disco automáticamente.

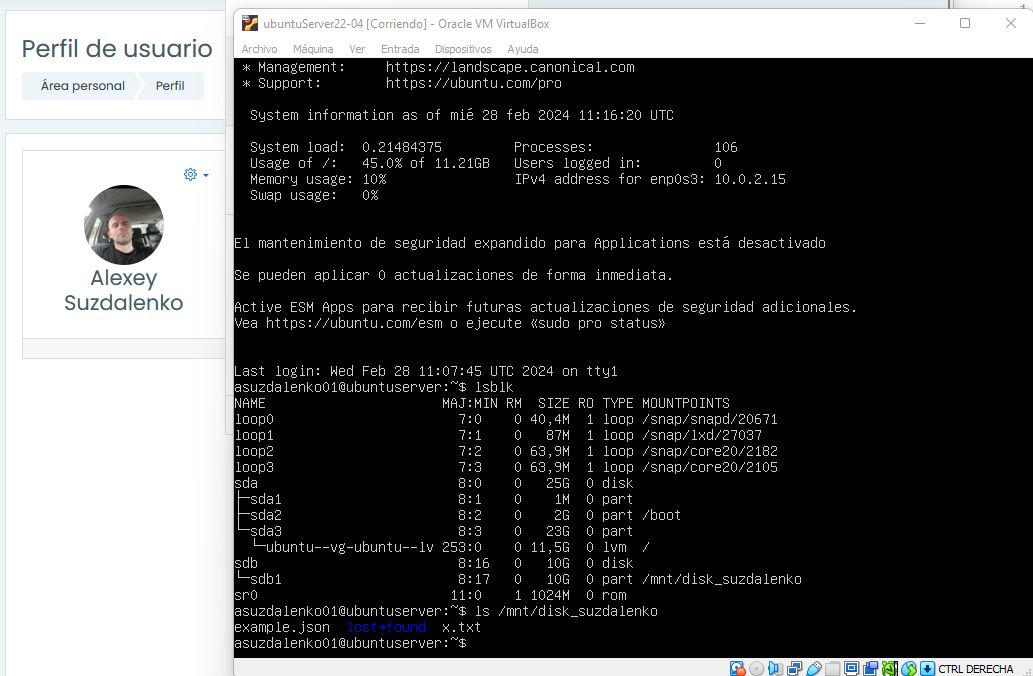
Vamos a comprobarlo, ejecutamos **reboot**

****

**8** reiniciamos el Ubuntu server, ejecuto siguientes comandos:

**lsblk**

**ls /mnt/disk\_suzdalenko**

****

Comprobamos que SI el disco se ha montado automáticamente y tiene archivos que yo había creado previamente x.txt, example.txt

**ACTIVIDAD 3.**

**chmod 750 tarea.txt**

Despues de ejecutar este comando los permisos de archivo tarea.txt seria

**rwxr-x---**

Propietario (u): **rwx** (lectura, escritura y ejecución)

Grupo (g): **r-x** (lectura y ejecución)

Otros (o): --- (sin permisos)

**chmod o+rw tarea.txt**

Despues de ejecutar este comando la cadena de permisos seria

**rwxr-xrw-**

se añade lectura y escritura para otros (o+rw). Entonces, la cadena de permisos se convierte en **rwxr-xrw-,** donde el propietario tiene lectura, escritura y ejecución (rwx), el grupo tiene lectura y ejecución (r-x), y otros tienen lectura y escritura (rw)

**chmod go-w tarea.txt**

Despues de ejecutar este comando la cadena de permisos de archivo tarea.txt seria

**rwxr-x--w-**

Se quita el permiso de escritura para el grupo y otros (go-w). La cadena de permisos resultante es **rwxr-x--w-**, donde el propietario tiene lectura, escritura y ejecución (rwx), el grupo tiene lectura y ejecución (r-x), y otros tienen solo permisos de escritura (--w).

**chmod u+x,g+w tarea.txt**

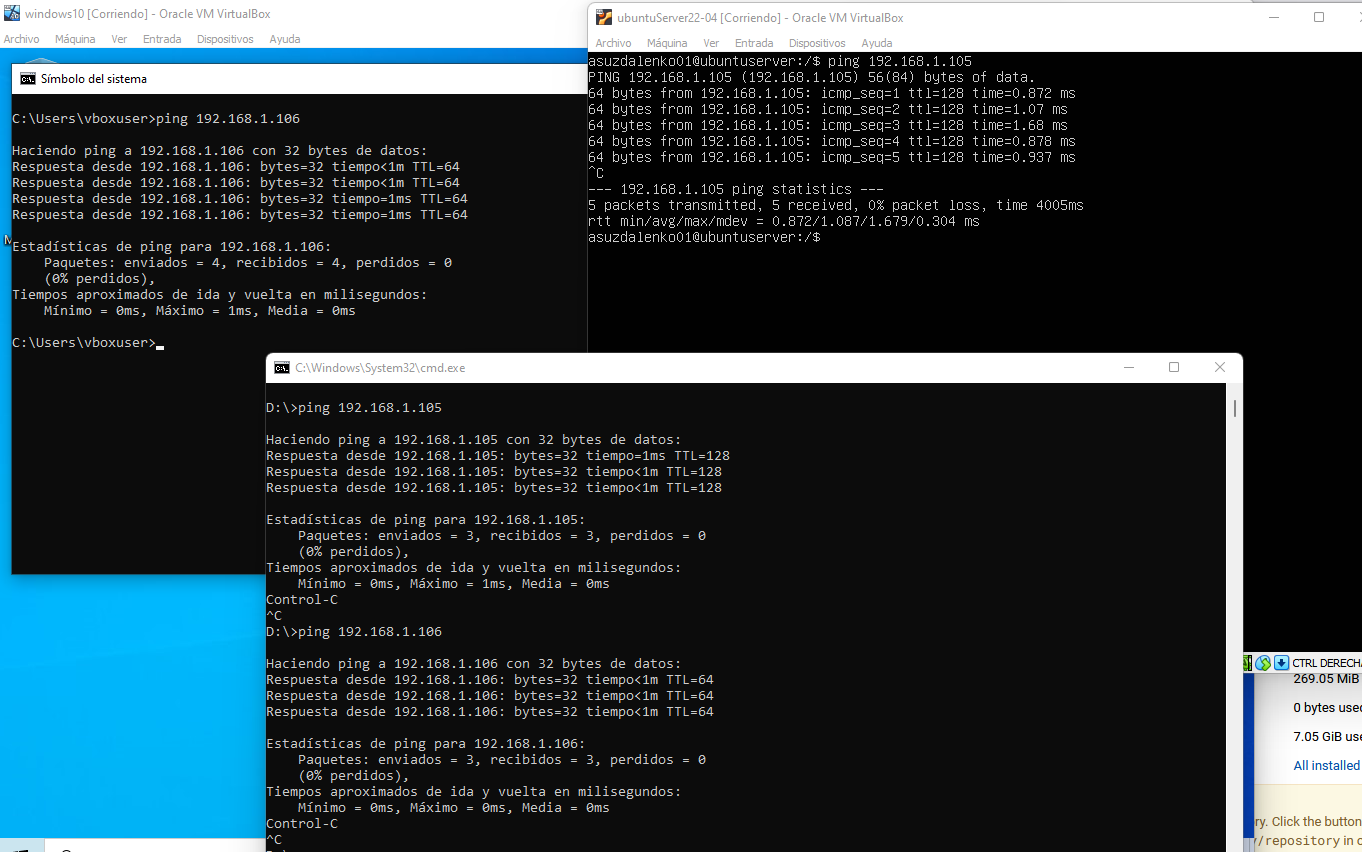
Despues de ejecutar este comando la cadena de permisos de archivo tarea.txt seria

**rwxr--r--**

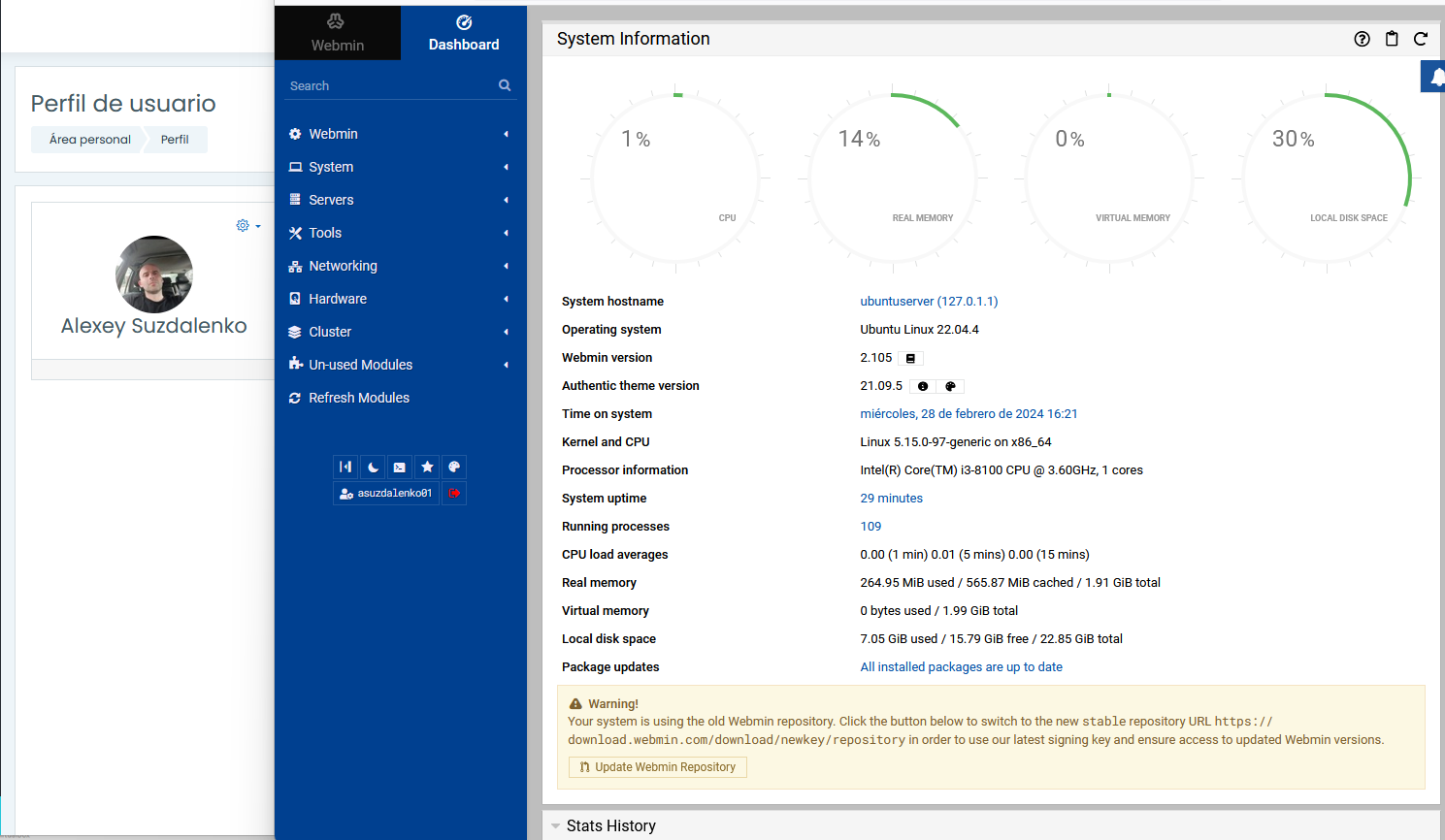
**Después de** chmod u+x,g+w tarea.txt**:** Se añade permiso de ejecución para el propietario (u+x) y permiso de escritura para el grupo (g+w). La cadena de permisos se convierte en **rwxr--r--,** donde el propietario tiene lectura, escritura y ejecución (rwx), el grupo tiene lectura y escritura (r--), y otros solo tienen permisos de lectura (r--).

**ACTIVIDAD 4**

Resulta que para usar webmin, según he entendido hace falta que la máquina virtual Ubuntu server este visible desde fuera o sea desde la maquina de anfitrión. Esto me ha llevado un tiempo y al final he conseguido que Ubuntu server(virtual), Windows 10(virtual) y la maquina anfitrión ven unas a otras y pueden comunicarse:



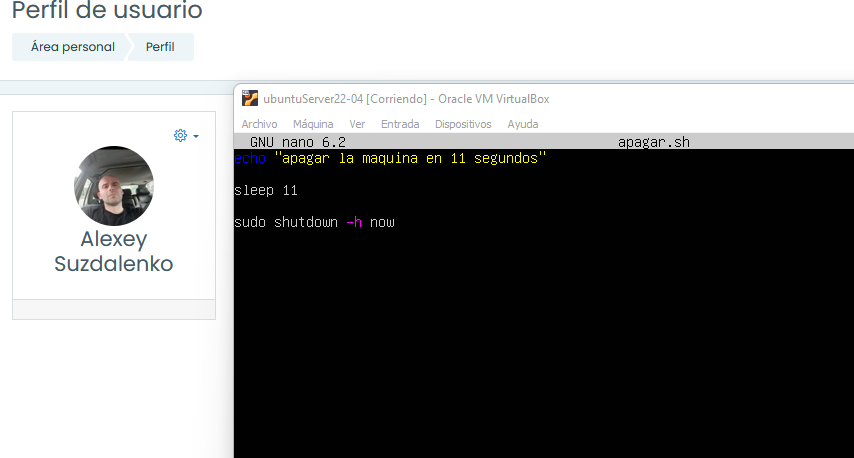
A partir de aquí, instalando webmin en Ubuntu server, ya puedo abrirle en navegador desde la maquina anfitrión.



**ACTIVIDAD 5**

**1** crea un script que apague la máquina. Confirma que funciona

**nano apagar.sh**

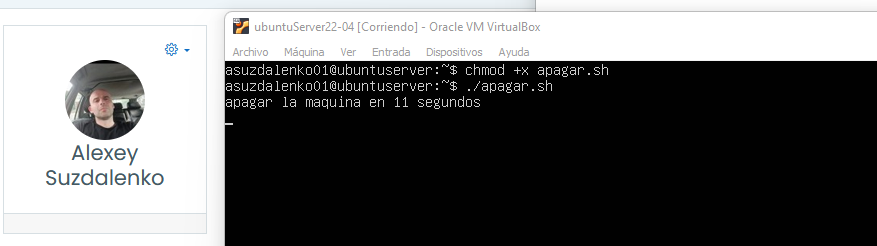
****

dar permisos de ejecución del escript

**chmod +x apagar.sh**

****

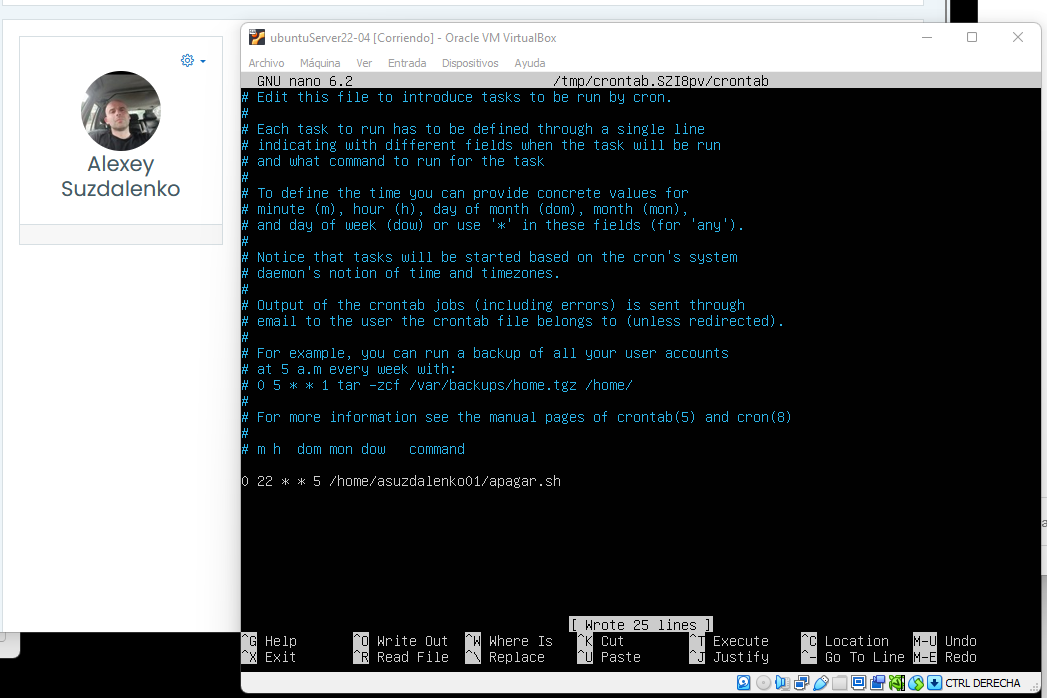
Probamos que funciona, ejecutamos **./apagar.sh**

****

Y seguido la maquina fue apagada

**2** realiza lo necesario para programar que el script se ejecuta todos los viernes a las 22:00

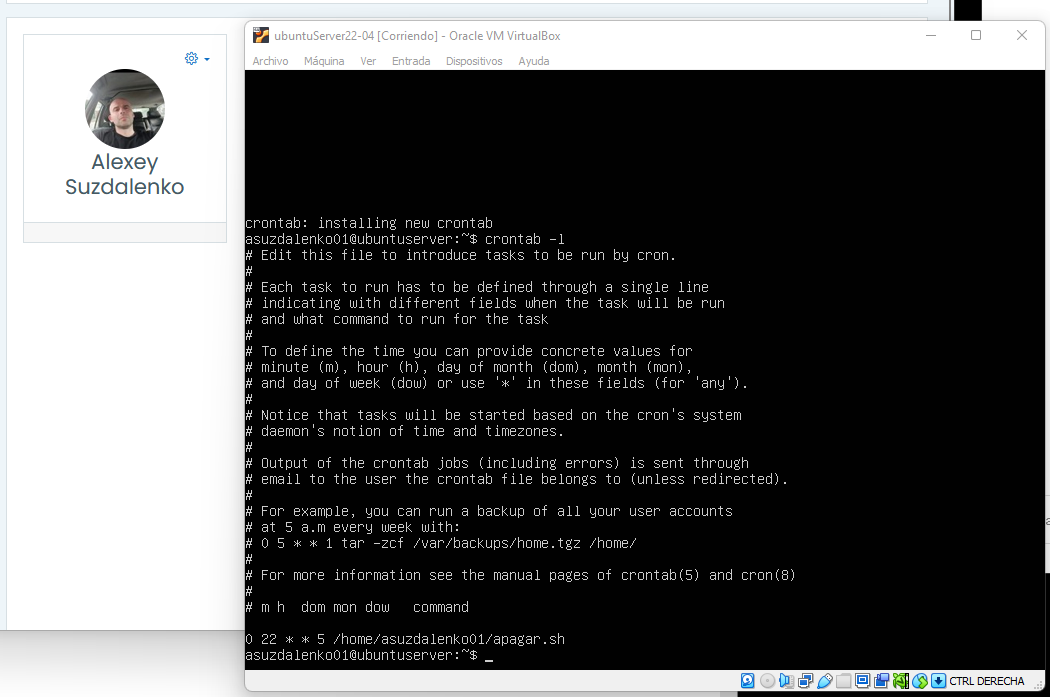
Abro el archivo crontab para editar **crontab -e**

Y escribo al final del archivo **0 22 \* \* 5 /ruta/del/script/apagar.sh** 

Esto programará el script para que se ejecute a las 22:00 (10:00 PM) cada viernes (día de la semana 5)

Para verificar que el archivo crontab esta modificado correctamente ejecuto

**crontab -l**

****