

<b>Universidad Autónoma Tomás Frías</b> <b>Ingeniería de Sistemas</b>	<b>LABORATORIO 7</b>
<b>SIS 737</b> <b>SEGURIDAD DE SISTEMAS</b>	Copias de seguridad automáticas
<b>DOCENTE:</b> M.SC. J. ALEXANDER DURÁN M.	<b>RU:</b> 109457
<b>AUXILIAR:</b> UNIV. ALDRIN ROGER MIRANDA	
<b>NOMBRE:</b> UNIV. SAAVEDRA MARCA PATRICIA EMILIA	<b>CI:</b> 13870923
<b>FECHA DE ENTREGA:</b> 26/04/25	

# DESCRIPCIÓN

## Recursos:

Se utilizará:

- 1 máquina virtual: Ubuntu server, también puede usar la Ubuntu cliente 12.04.

Para Linux vamos a probar la herramienta **rsync**. Es una herramienta sencilla pero muy potente, que nos permite sincronizar directorios en una misma máquina o entre dos máquinas.

---

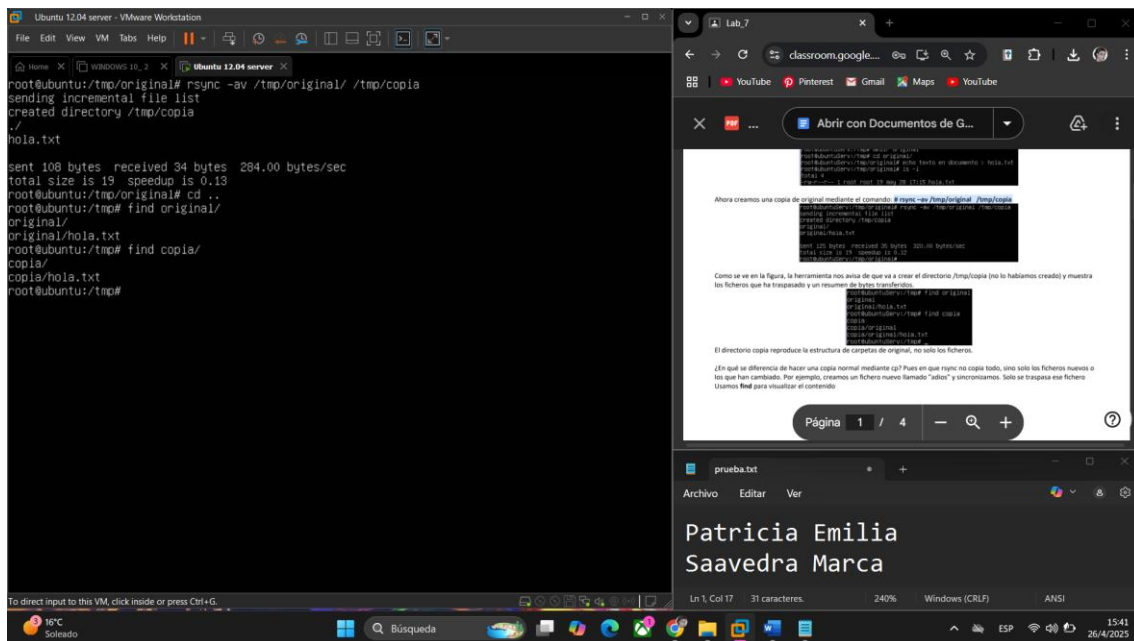
## PARTE 1

Entramos en nuestro Ubuntu Server 12.04 y nos ponemos en modo super-usuario con:

```
sudo -i
```

Nos vamos al `/tmp` y creamos un directorio llamado `original` con un fichero `hola.txt`.



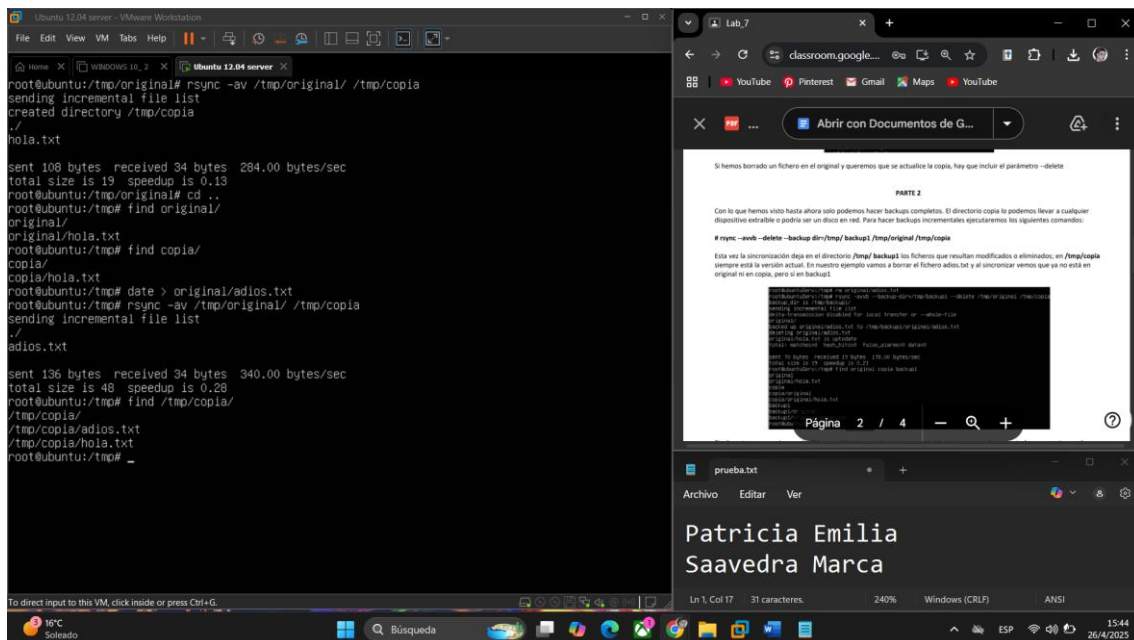


El directorio copia reproduce la estructura de carpetas de original, no solo los ficheros.

¿En qué se diferencia de hacer una copia normal mediante `cp`?  
Pues en que **rsync** no copia todo, sino solo los ficheros nuevos o los que han cambiado.

Por ejemplo, creamos un fichero nuevo llamado "adios" y sincronizamos. Solo se traspasa ese fichero.

Usamos `find` para visualizar el contenido.



Si hemos borrado un fichero en el original y queremos que se actualice la copia, hay que incluir el parámetro `--delete`.

## 1. Borrar `adios.txt` en el directorio original:

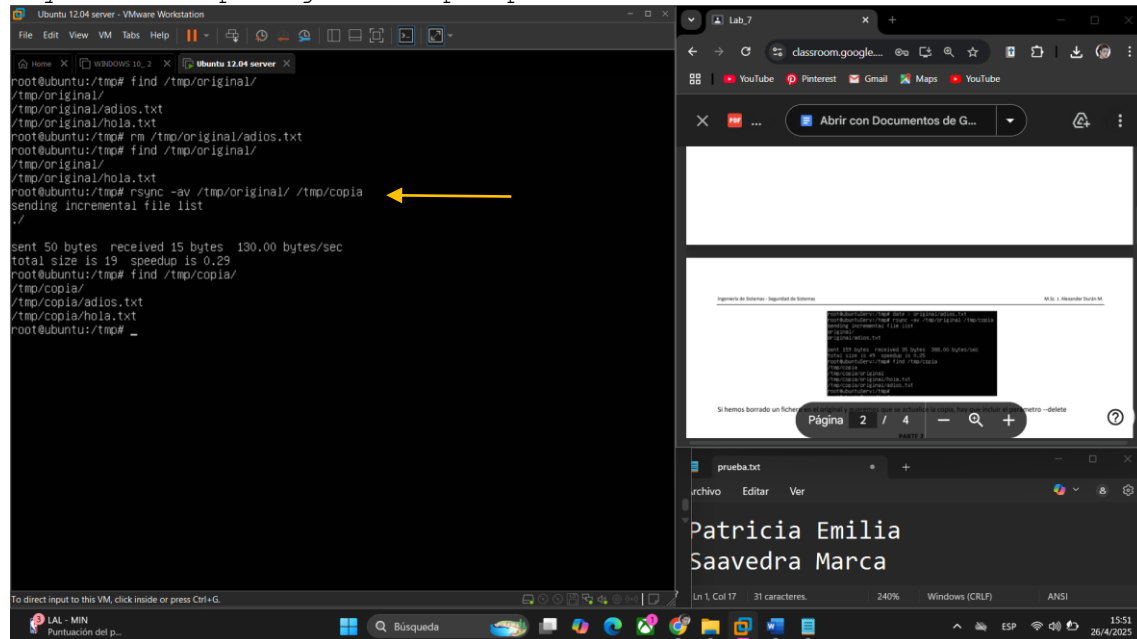
```
rm /tmp/original/adios.txt
```

Ahora en `/tmp/original` solo queda:

- `hola.txt`

## 2. Sincronizar otra vez SIN `--delete`:

```
rsync -av /tmp/original /tmp/copia
```



## Resultado:

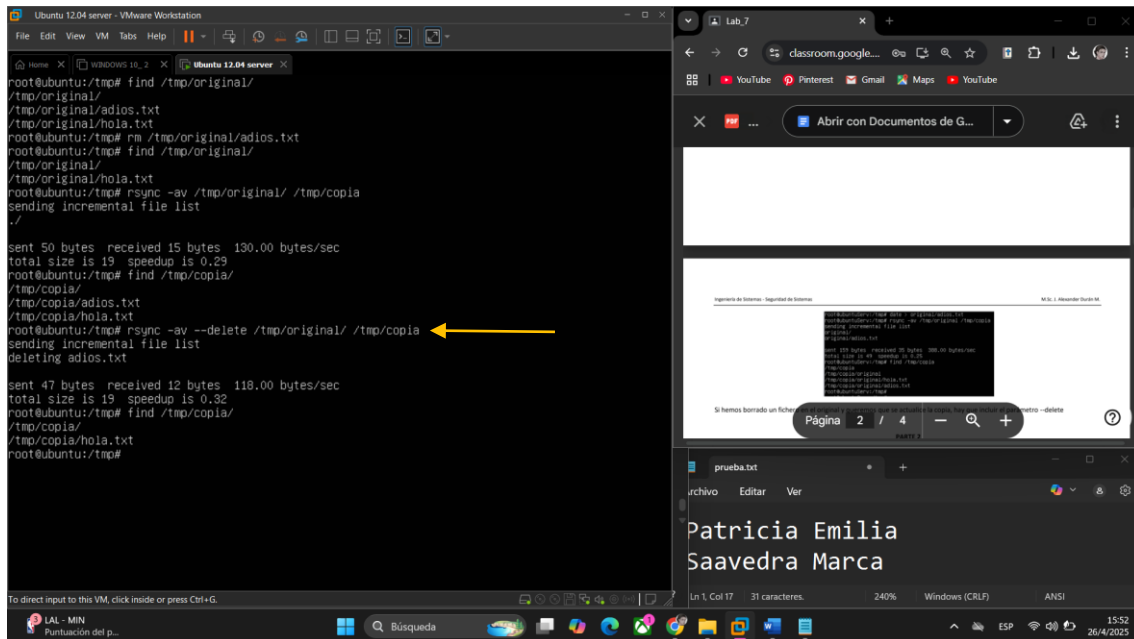
En `/tmp/copia` todavía tienes:

- `hola.txt`
- `adios.txt`

(porque `rsync` no borra archivos si no le pides explícitamente.)

## 3. Sincronizar ahora CON `--delete`:

```
rsync -av --delete /tmp/original /tmp/copia
```



## Resultado final:

En /tmp/copia ahora solo queda:

- hola.txt

*(adios.txt fue eliminado de /tmp/copia porque ya no existe en /tmp/original.)*

---

## PARTE 2

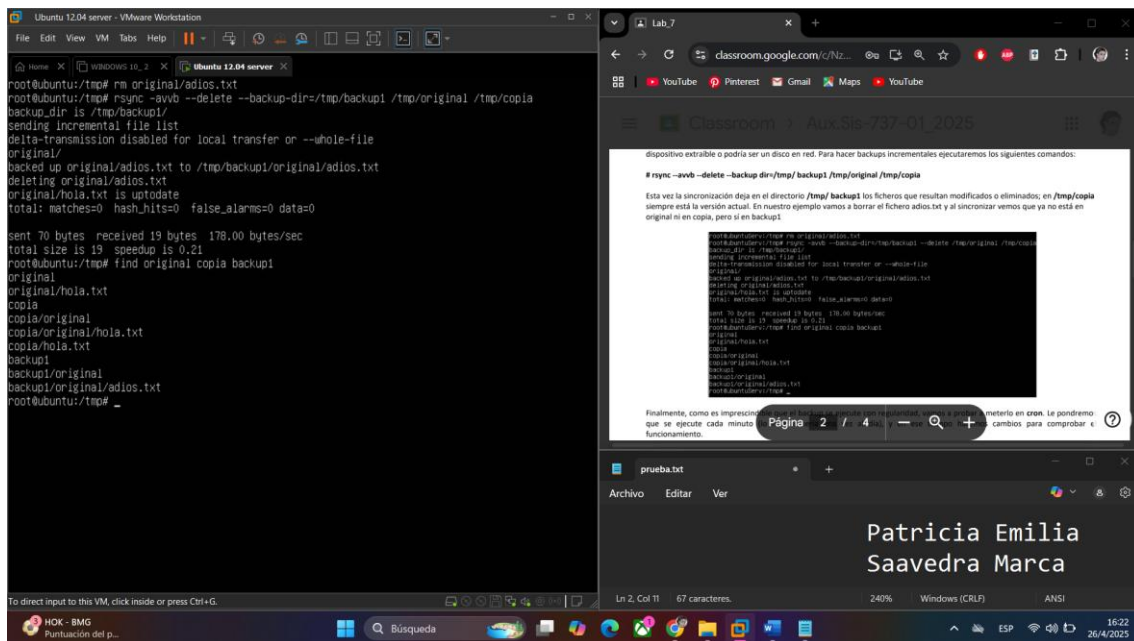
Con lo que hemos visto hasta ahora solo podemos hacer backups completos.

El directorio copia lo podemos llevar a cualquier dispositivo extraíble o podría ser un disco en red.

Para hacer backups incrementales ejecutaremos los siguientes comandos:

```
rsync -avvb --delete --backup-dir=/tmp/backup1 /tmp/original /tmp/copia
```

Esta vez la sincronización deja en el directorio /tmp/backup1 los ficheros que resultan modificados o eliminados; en /tmp/copia siempre está la versión actual.



En nuestro ejemplo vamos a borrar el fichero `adios.txt` y al sincronizar vemos que ya no está en `original` ni en `copia`, pero sí en `backup1`.

Finalmente, como es imprescindible que el backup se ejecute con regularidad, vamos a probar a meterlo en `cron`.

Le pondremos que se ejecute cada minuto (lo normal sería una vez al día), y en ese tiempo haremos cambios para comprobar el funcionamiento.

Crearemos un script llamado `mibackup.sh` (darle permisos de lectura, escritura y ejecución con `chmod +x /tmp/mibackup.sh`) que invocaremos desde el `cron`.

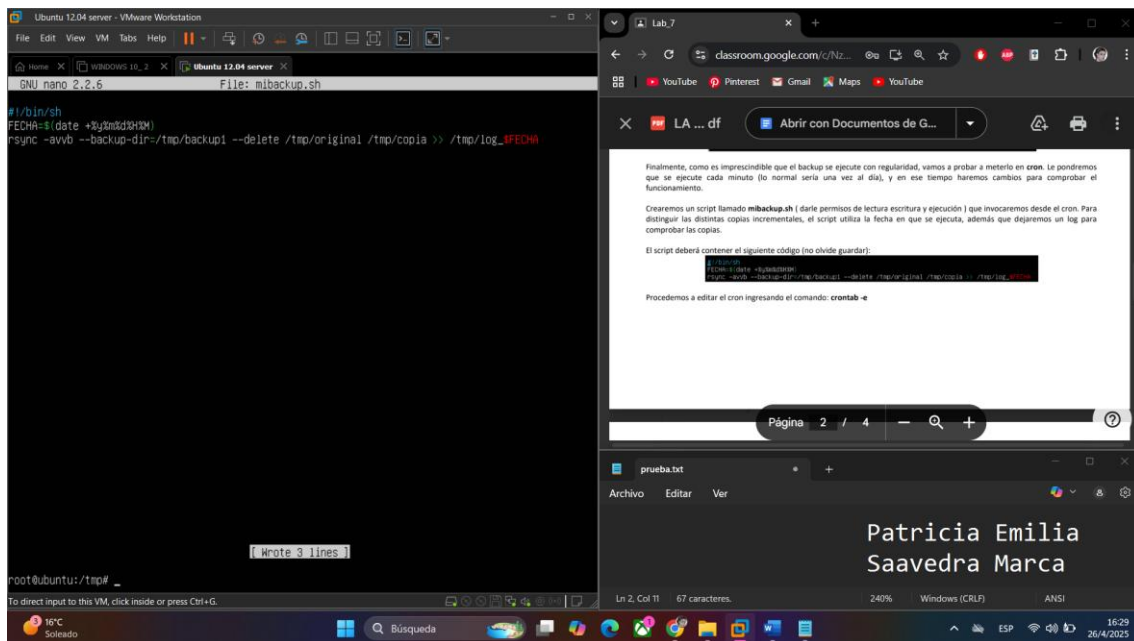
Para distinguir las distintas copias incrementales, el script utiliza la fecha en que se ejecuta, además que dejaremos un log para comprobar las copias.

En el script escribiremos:

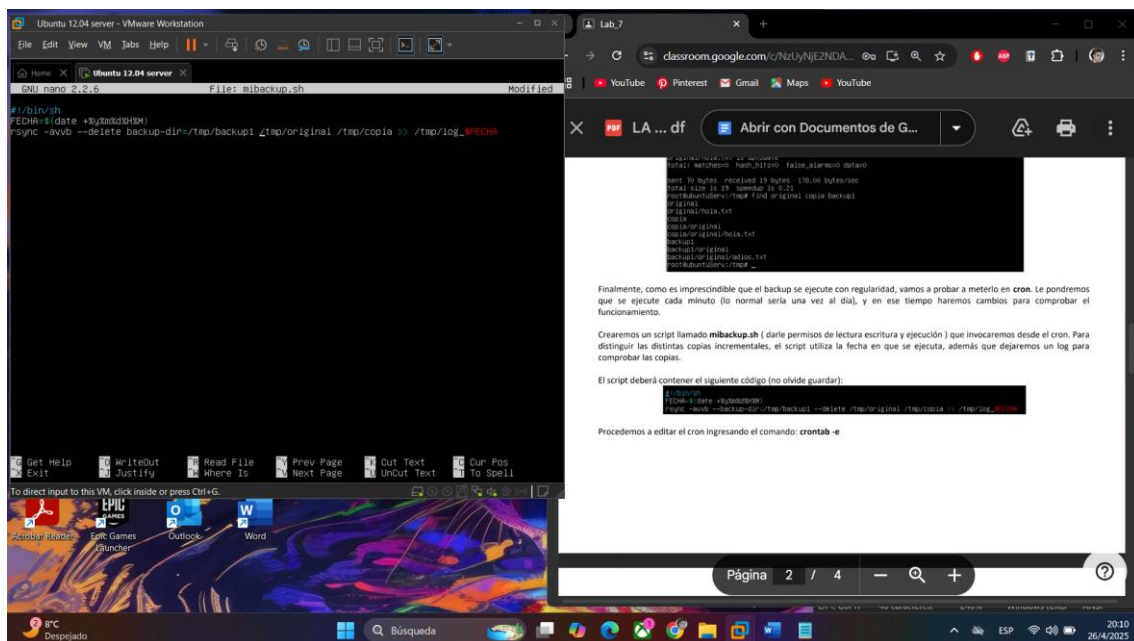
```
#!/bin/bash

FECHA=$(date +%Y%m%d%H%M)

rsync -av --delete --backup-dir=/tmp/backup1 /tmp/original
/tmp/copia >> /tmp/log_backup_$FECHA
```



mmm...



El script deberá contener el siguiente código (no olvide guardar):

Procedemos a editar el `cron` ingresando el comando:

`crontab -e`

para luego poner al final:

`* * * * */tmp/mibackup.sh`



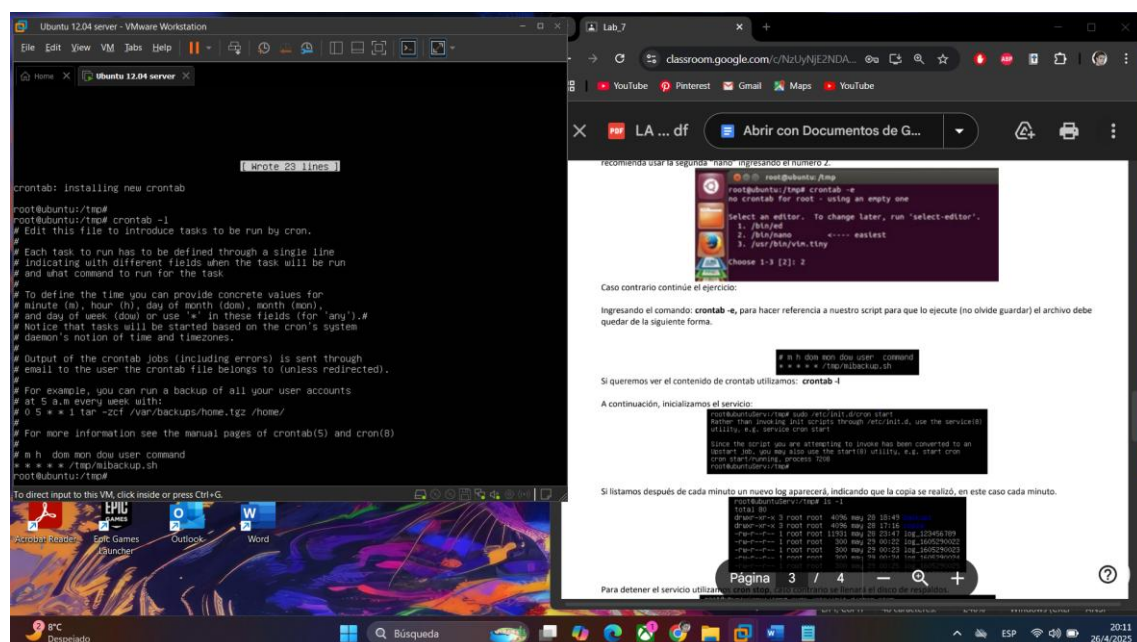
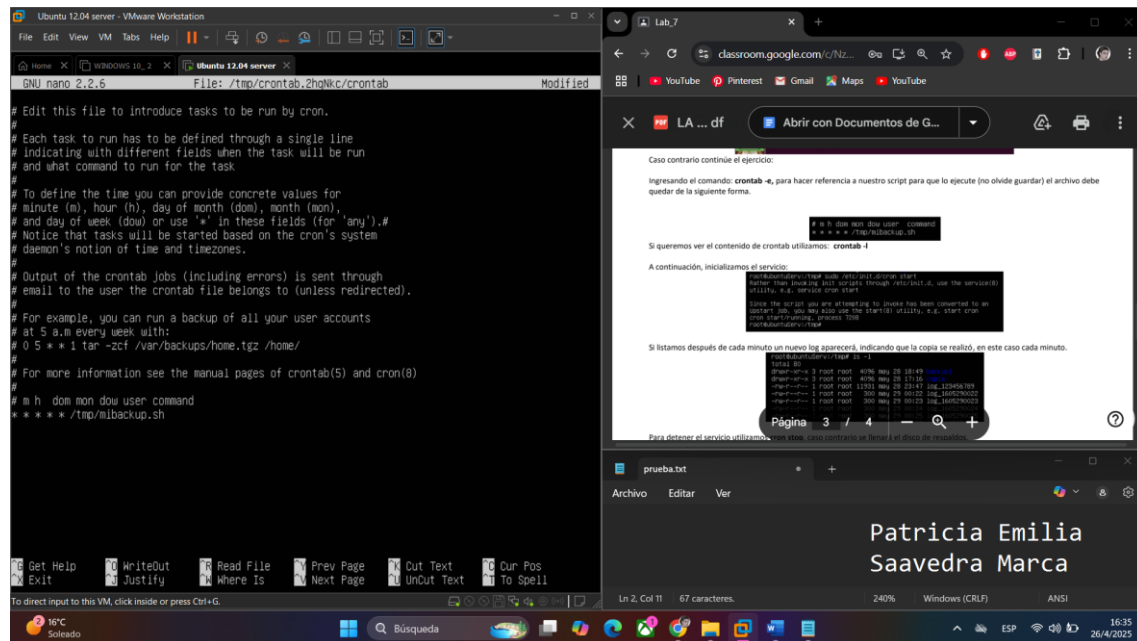
## IMPORTANTE:

Si no está utilizando Ubuntu Server, puede presentarse la siguiente pantalla, donde le pedirá elegir el editor, se recomienda usar la segunda opción “nano” ingresando el número 2.

Caso contrario, continúe el ejercicio ingresando el comando: `crontab -e`

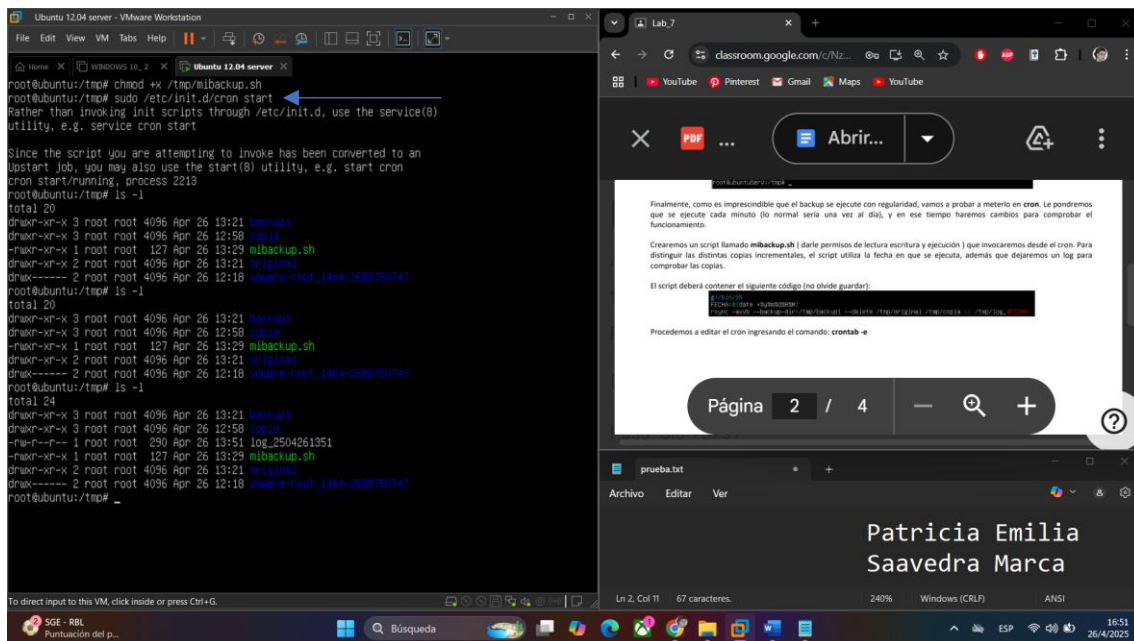
Para hacer referencia a nuestro script para que lo ejecute (no olvide guardar), el archivo debe quedar de la siguiente forma.

Si queremos ver el contenido de `crontab`, utilizamos: `crontab -l`

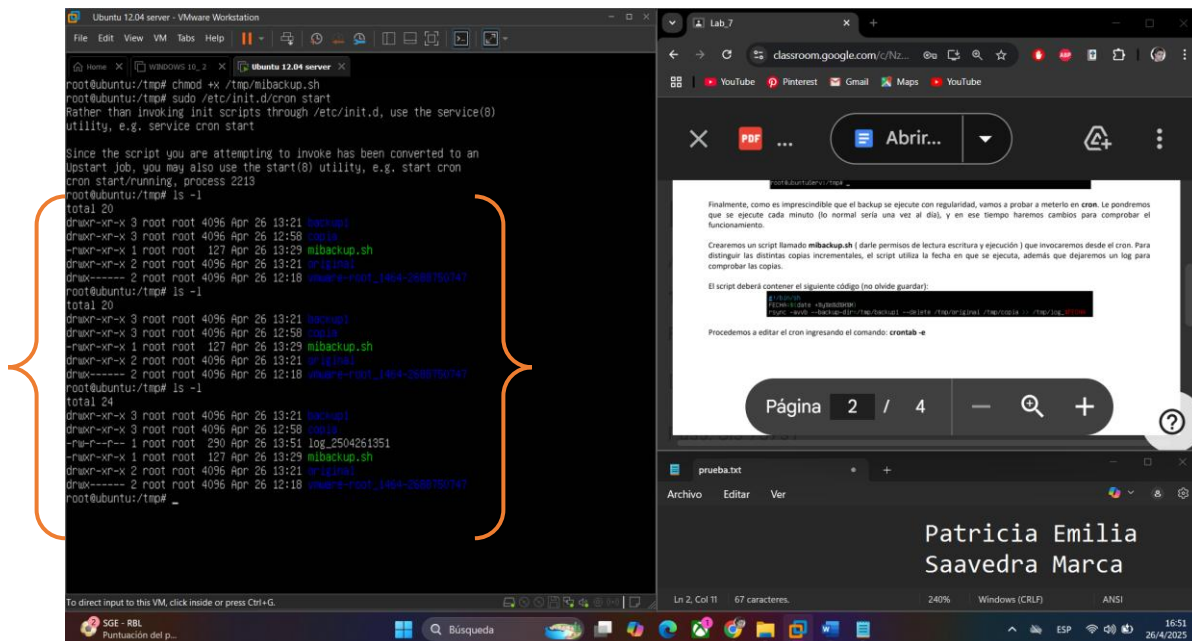


A continuación, inicializamos el servicio: `sudo /etc/init.d/cron start`

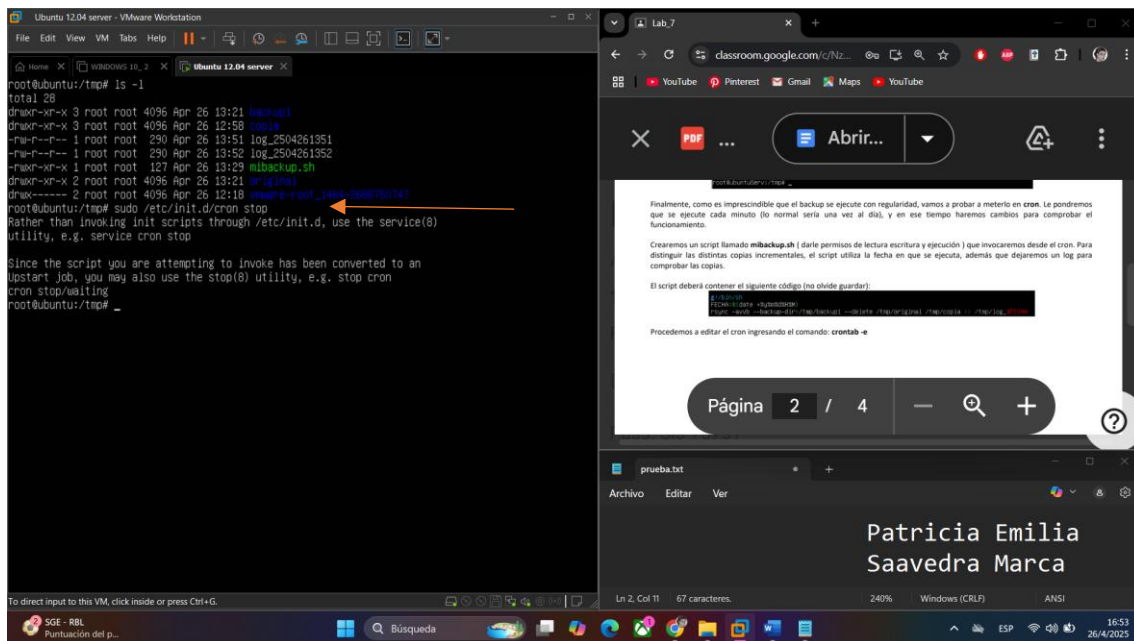




Si listamos con **ls -l** después de cada minuto un nuevo log aparecerá, indicando que la copia se realizó, en este caso cada minuto.



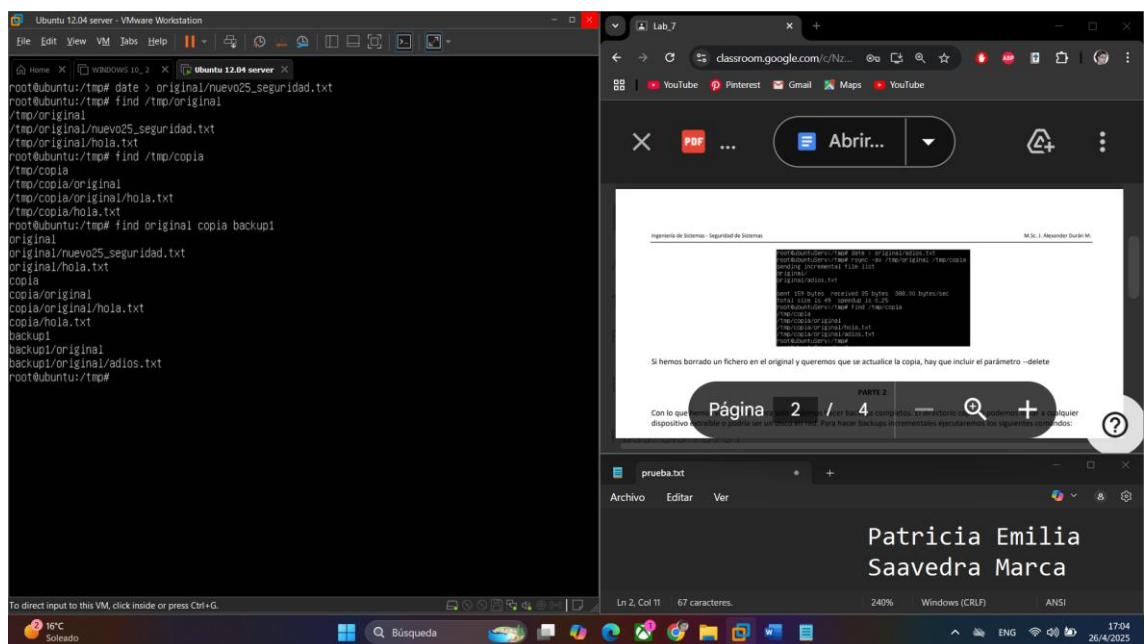
Para detener el servicio utilizamos: **cron stop**



Caso contrario se llenará el disco de respaldos.

## EVALUACIÓN

1. Primeramente, detenga el servicio de `cron`. Ahora cree un nuevo archivo de texto en la carpeta `original` con el nombre `nuevo25_seguridad.txt`, y ayudado por el comando `find` que utilizó en la primera parte, observe el comportamiento de las 3 carpetas.



Comando `find` que utilizó: **find original copia backup1**

## ¿Qué archivos están inicialmente en original, copia y backup1

### Original:

nuevo25\_seguridad.txt

hola.txt

### Copia:

hola.txt

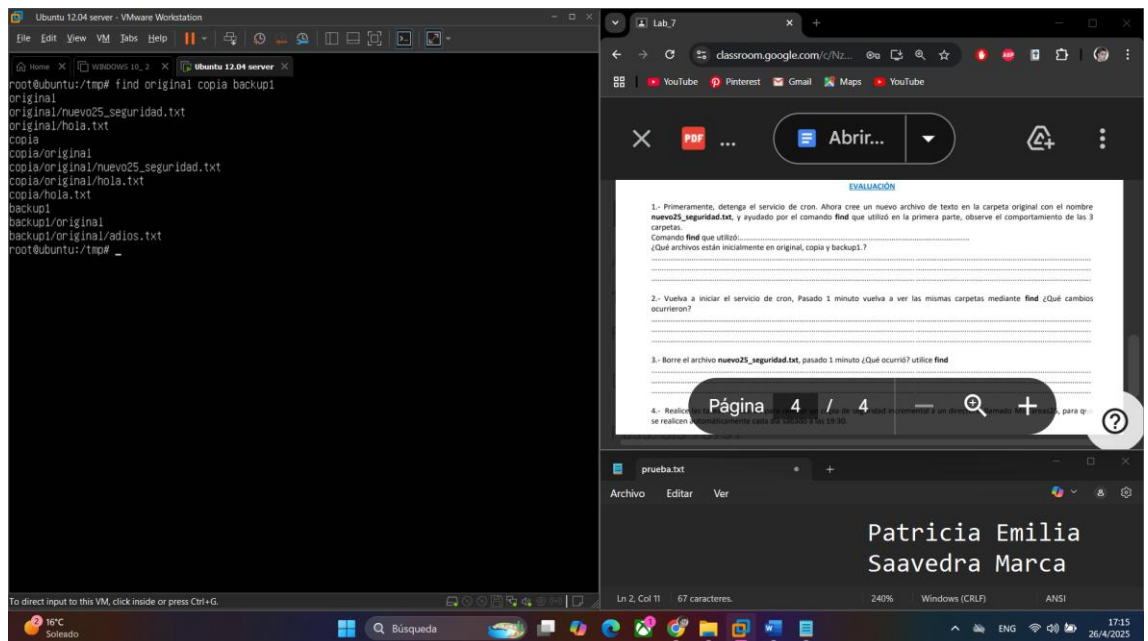
### Backup1:

adios.txt

## 2. Vuelva a iniciar el servicio de cron.

Pasado 1 minuto, vuelva a ver las mismas carpetas mediante `find`.

The screenshot shows a dual-screen setup. On the left is a terminal window titled 'Ubuntu 12.04 server - VMware Workstation'. It displays the output of a `find` command, showing files in the 'original', 'copia', and 'backup1' directories. The files listed are 'nuevo25\_seguridad.txt', 'hola.txt', and 'adios.txt'. Below the terminal output, there is a message about the script being converted to an Upstart job and instructions to start the cron service. On the right is a web browser window showing a Google Classroom page titled 'Aux.Sis-737-01\_2025'. The page contains an evaluation section with four questions related to the cron service and file management. The browser window also shows a sidebar with a list of files and a search bar.



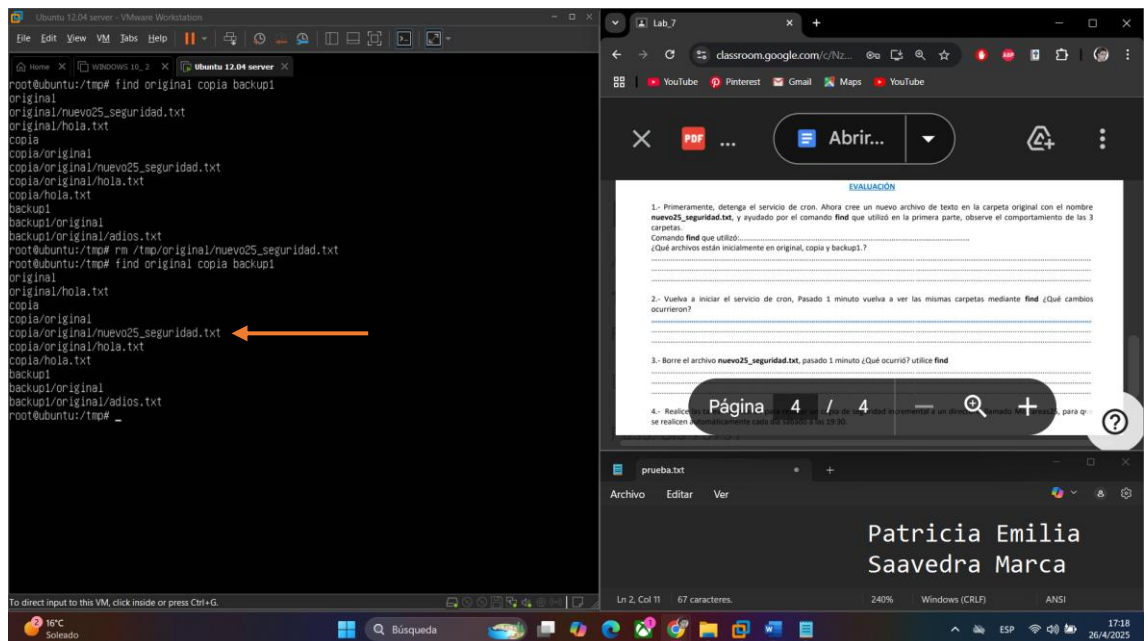
## ¿Qué cambios ocurrieron?

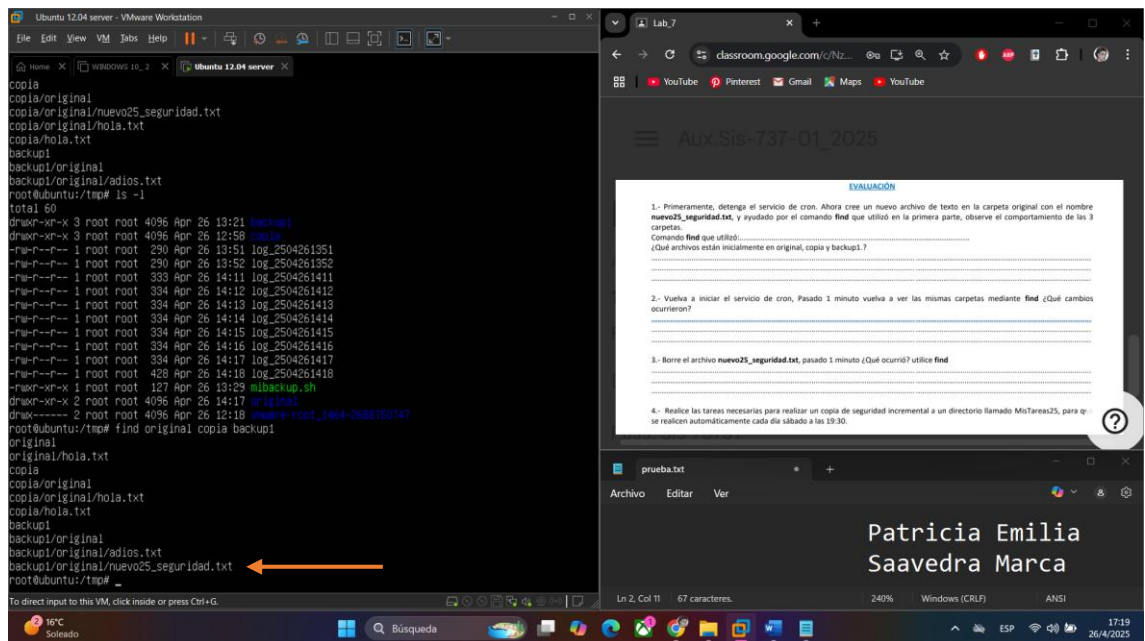
Pues el archivo `nuevo25_seguridad.txt` se copió en `/tmp/copia` automáticamente.

### 3. Borre el archivo `nuevo25_seguridad.txt`.

Pasado 1 minuto, ¿qué ocurrió? (utilice `find`).  
Pues el archivo `nuevo25_seguridad.txt` se eliminó de `original`, se manteni6 en `/tmp/copia` un momento, y luego pasó a `backup1`.

Utilizamos `find original copia backup1`.





4. Realice las tareas necesarias para realizar una copia de seguridad incremental a un directorio llamado MisTareas25, para que se realicen automáticamente cada día sábado a las 19:30.

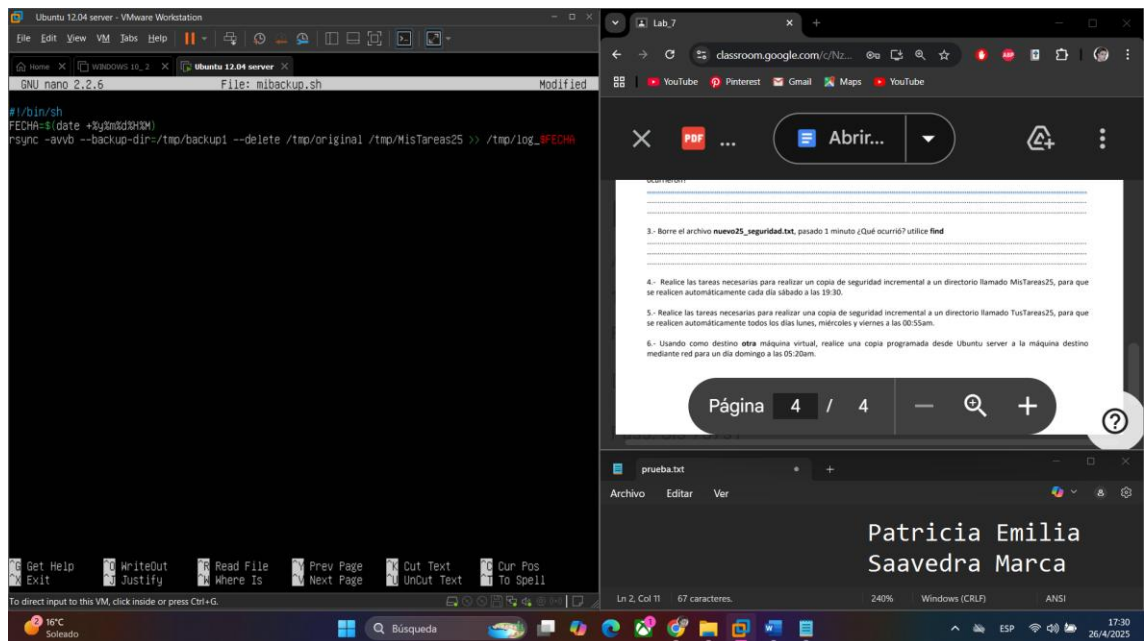
Después de crear con `mkdir /tmp/MisTareas25`, modificamos en nuestro script: `mibackup.sh`, para sincronizar el **original** a **MisTareas25**, con:

```
#!/bin/bash
```

```
FECHA=$(date +%Y%m%d%H%M)
```

```
rsync -av --delete --backup-dir=/tmp/backup1 /tmp/original
/tmp/MisTareas25 >> /tmp/log_backup_$FECHA
```

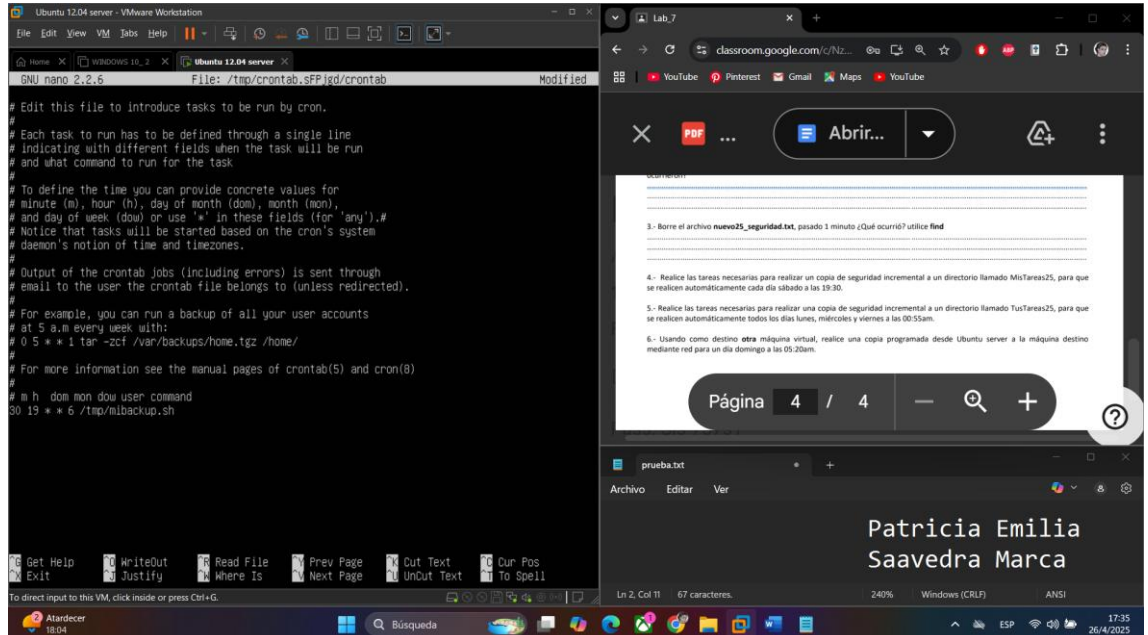




Agregamos el script al cron, con crontab -e.

Agregamos la siguiente línea al final: 30 19 \* \* 6 /tmp/mibackup.sh

Esto ejecutará el script **mibackup.sh** todos los sábados a las 19:30 (el 6 en la posición de los días de la semana representa el sábado).



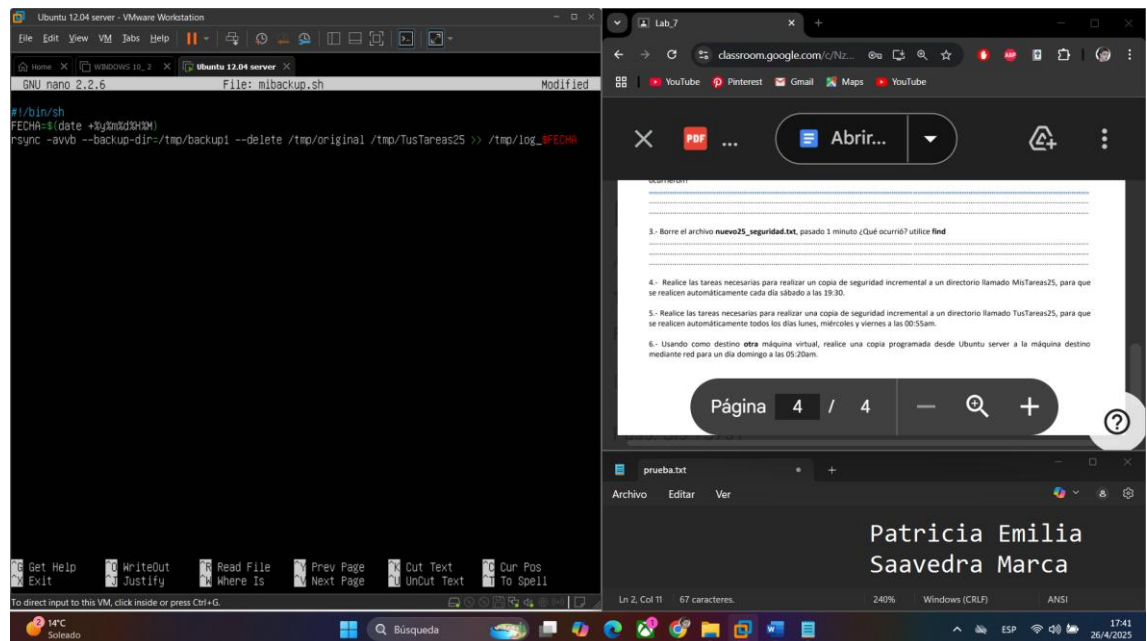
5. Realice las tareas necesarias para realizar una copia de seguridad incremental a un directorio llamado TusTareas25, para que se realicen automáticamente todos los días lunes, miércoles y viernes a las 00:55 am.

Después de crear con `mkdir /tmp/MisTareas25`, modificamos en nuestro script: `mibackup.sh`, para sincronizar el **original** a **MisTareas25** con:

```
#!/bin/bash
```

```
FECHA=$(date +%Y%m%d%H%M)
```

```
rsync -av --delete --backup-dir=/tmp/backup1 /tmp/original  
/tmp/TusTareas25 >> /tmp/log_backup_$(FECHA)
```

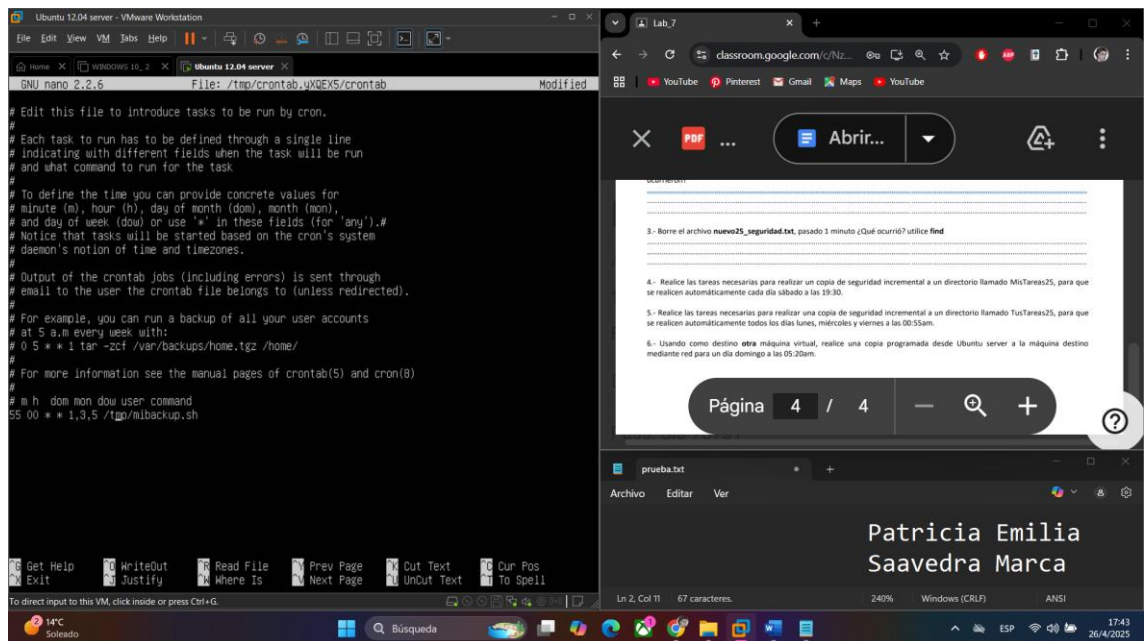


Agregamos el script al cron, con `crontab -e`.

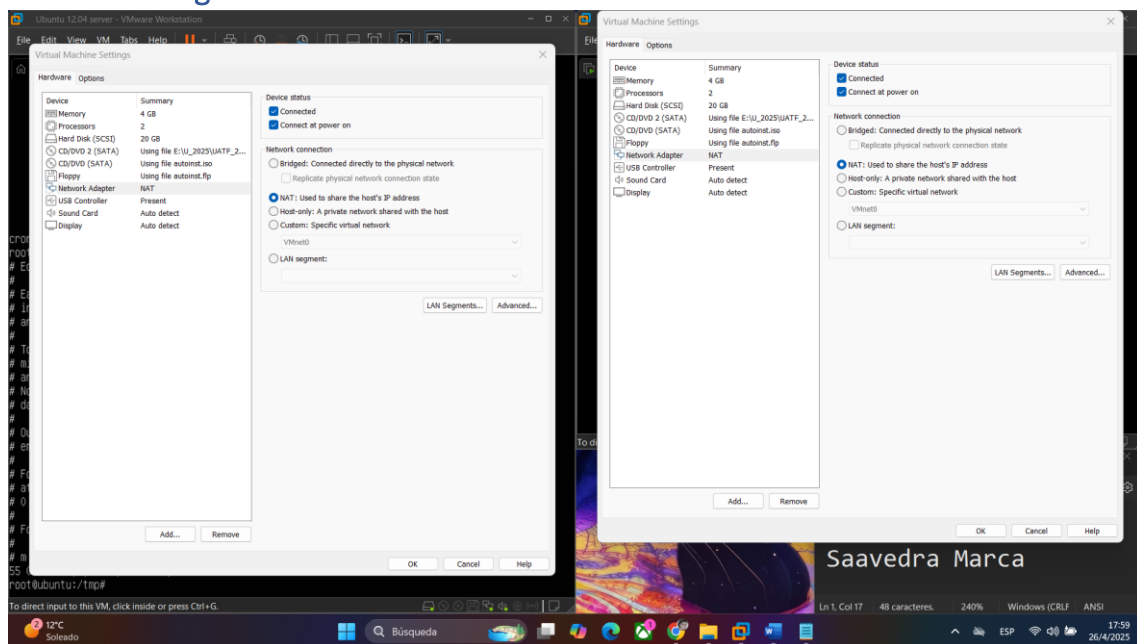
Agregamos la siguiente línea al final: `55 00 * * 1,3,5`  
`/tmp/mibackup.sh`

Esto ejecutará el script **mibackup.sh** a las **00:55 am** todos los **lunes (1)**, **miércoles (3)** y **viernes (5)**.

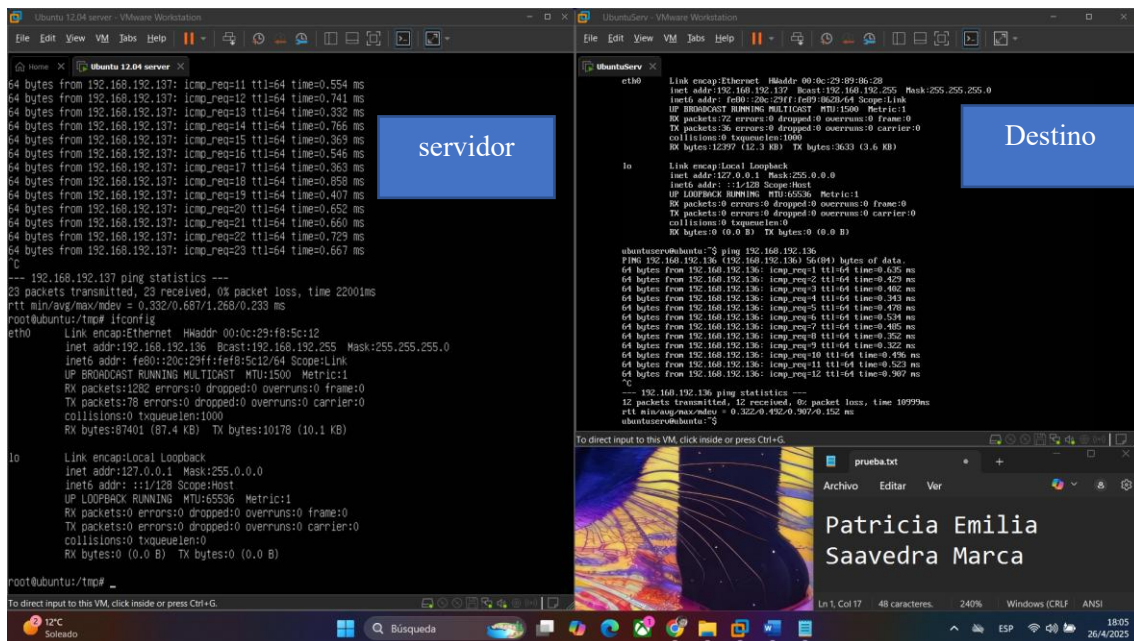




- Usando como destino otra máquina virtual, realice una copia programada desde Ubuntu server a la máquina destino mediante red para un día domingo a las 05:20 am.

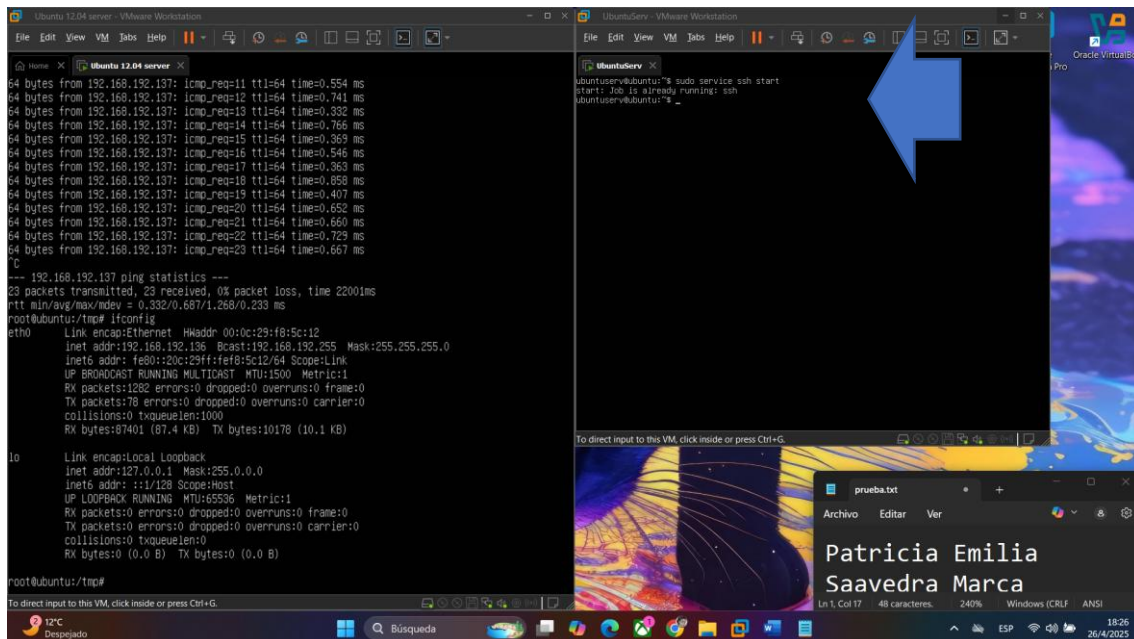


Verifica la conexión

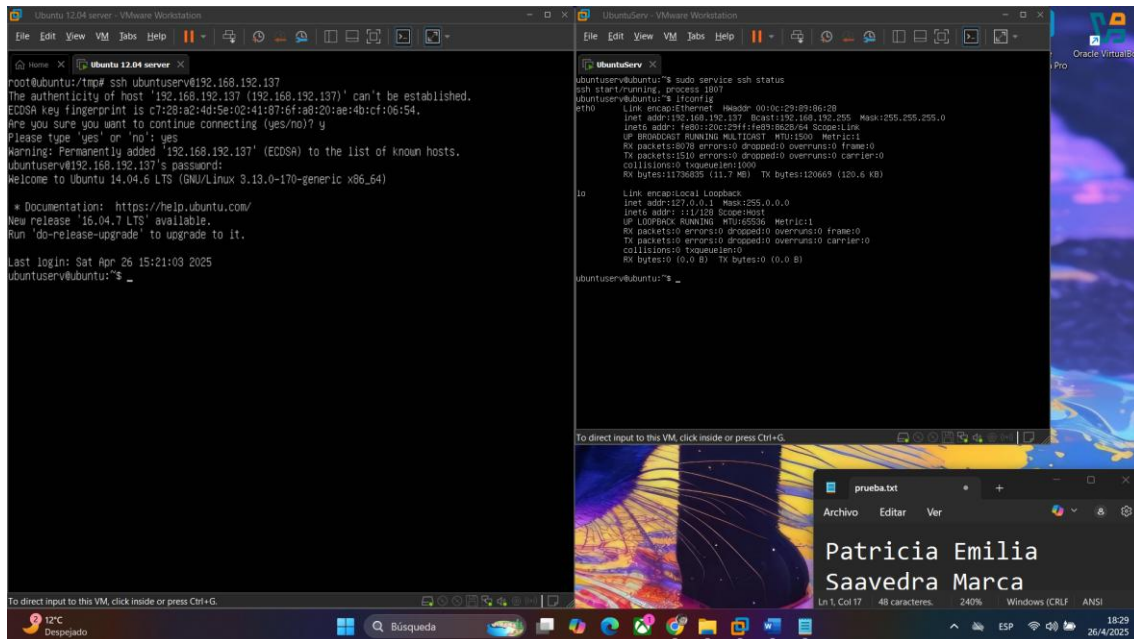


Cómo configurar SSH sin contraseña

En la máquina virtual destino, instalamos ssh, e iniciamos el servicio.



Verificamos que haya conexión ssh desde el servidor al destino, con ssh `ubuntuser@192.168.192.137`



**OJO: MEJOR SI ACÁ EN LA MAQUINA SERVIDOR, LLEGAN A CREAR UN PERMISO PARA QUE SE PUEDA INGRESAR AL SSH SIN CONTRASEÑA.**

1. Crea una clave SSH (si no tienes)

En la máquina origen:

```
ssh-keygen
```

Te saldrá algo como:

Enter file in which to save the key (/root/.ssh/id\_rsa):

Simplemente presiona **Enter** (para aceptar la ruta por defecto).

Después:

Enter passphrase (empty for no passphrase):

¡IMPORTANTE! Solo presiona **Enter** (dejar vacío).

Así no te pedirá clave nunca.

Te debe salir algo como:

```
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
```

2. Copia la clave pública a la máquina destino

Ahora ejecuta:

```
ssh-copy-id ubuntu:192.168.192.137
```

Te preguntará la primera vez:

```
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Escribe **yes** y presiona **Enter**.

Luego **te pedirá la contraseña del usuario `ubuntuserv`** (esta vez sí tienes que escribirla manualmente).

Después de eso, **nunca más** te la volverá a pedir.

---

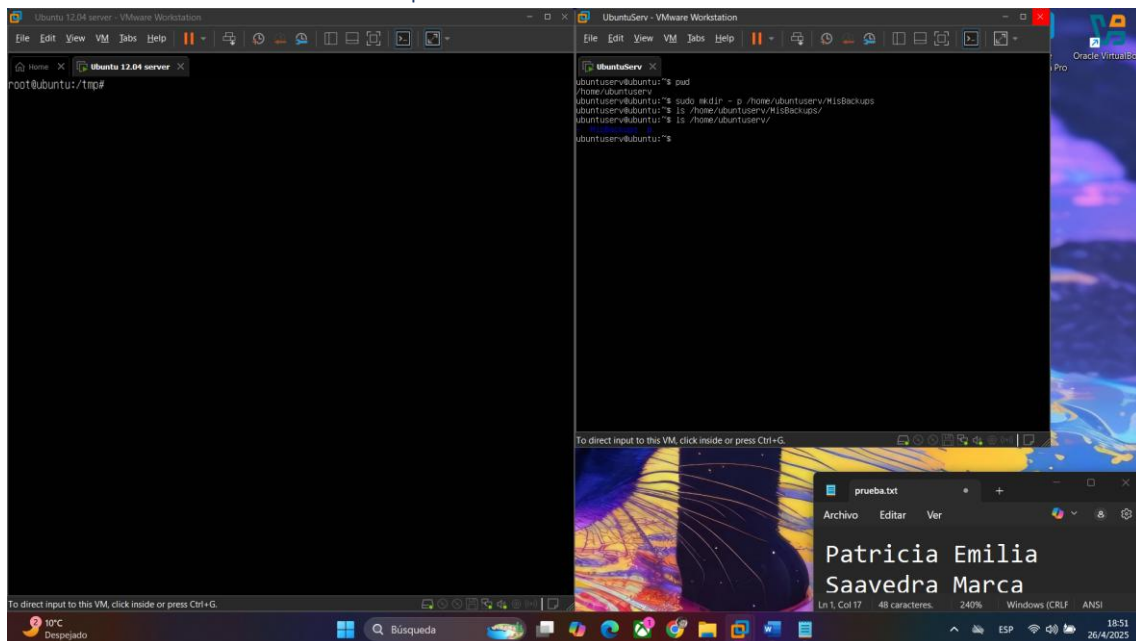
### 3. Prueba si funciona

Desde la máquina origen, prueba:

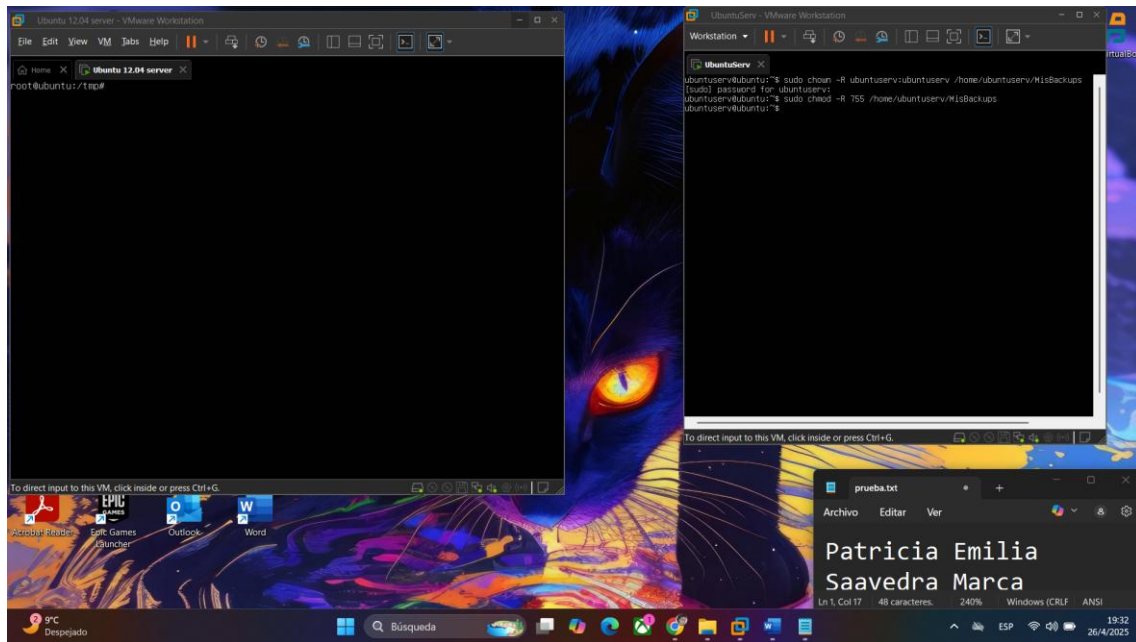
```
ssh ubuntuserv@192.168.192.137
```

Si **entra directamente sin pedir contraseña**, ¡listo!

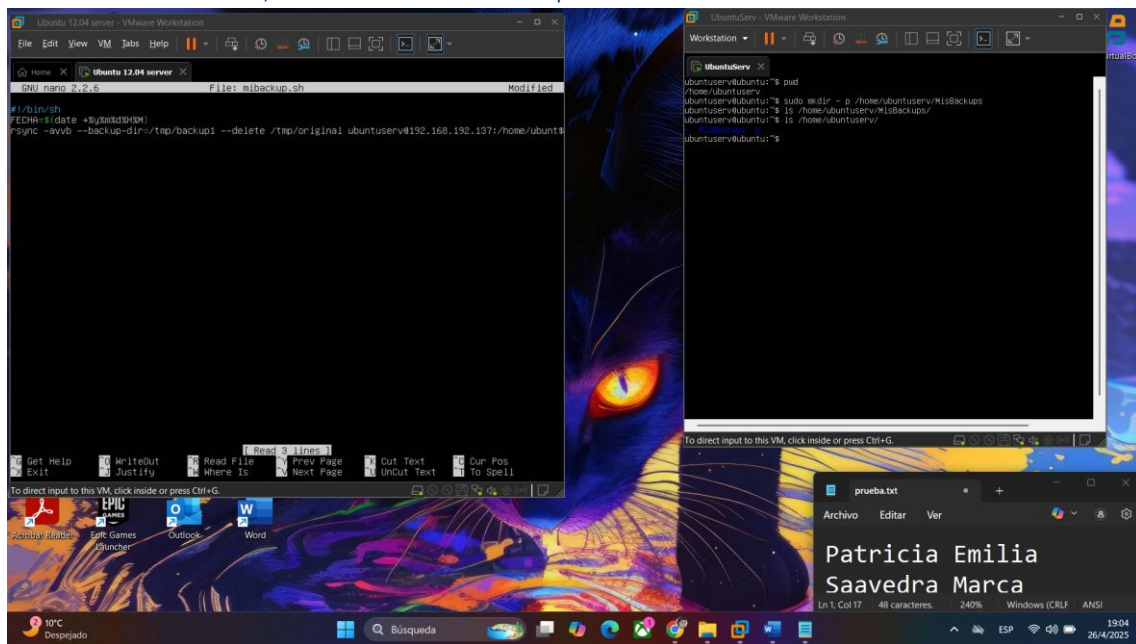
Creamos el directorio en la máquina destino



Y damos permisos:

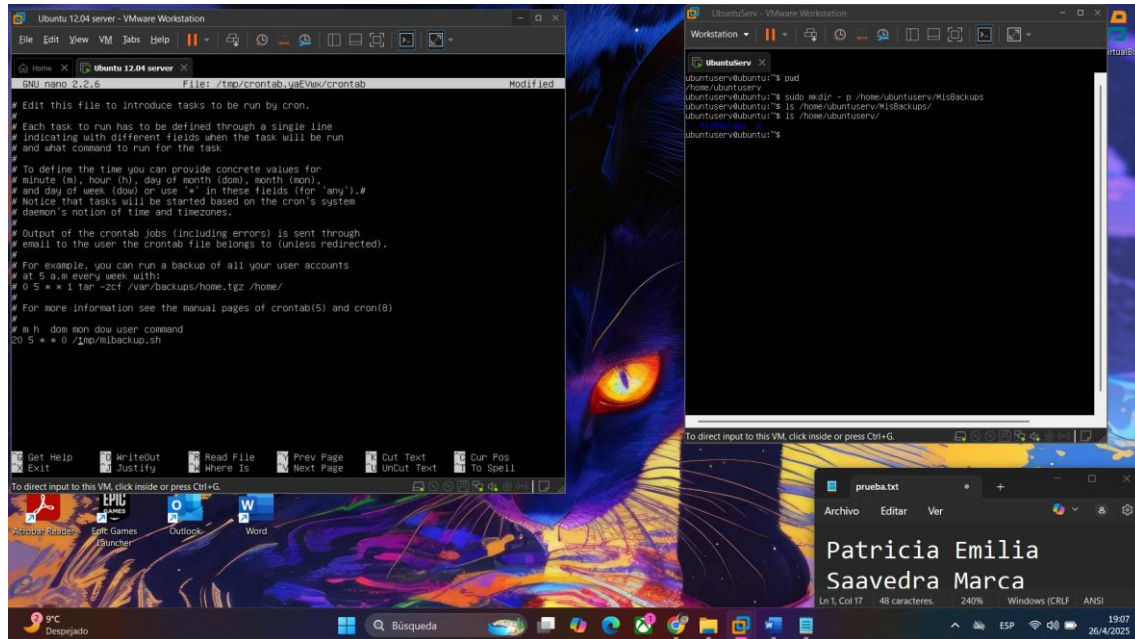


Ahora en el servidor, editamos nuestro script.





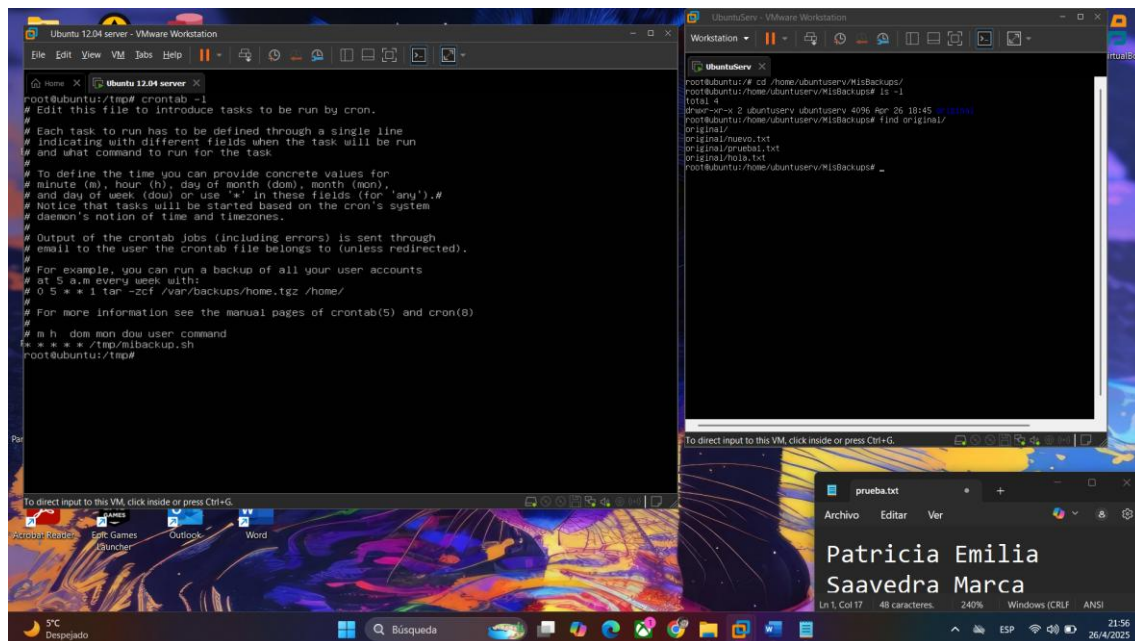
Luego configuramos el cron



Para que se ejecute cron, pues ponemos en el servidor, `sudo /etc/init.d/cron start`.

## VERIFICACIÓN

CAMBIAMOS EN EL CRON LAS ESPECIFICACIONES, PARA VER SI REALMENTE SE GUARDA EN LA MAQUINA VIRTUAL DESTINO.



Ejecutamos cron, creamos un archivo nuevo .txt y verificamos que tanto en el original (servidor) y en MisBackups (destino) se guardaron las modificaciones.

