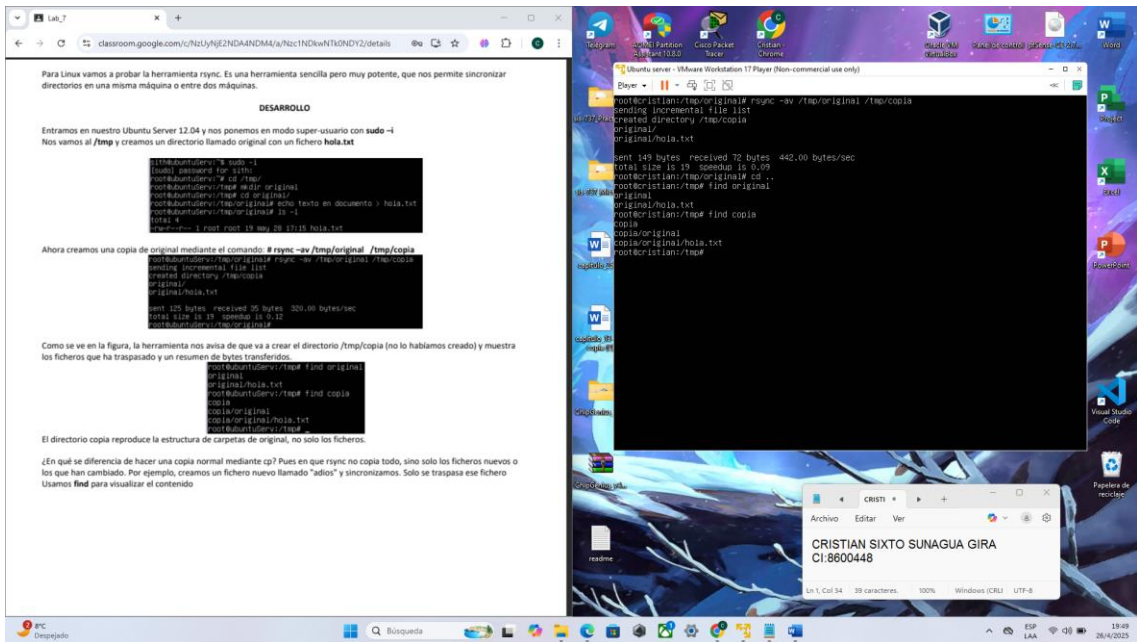
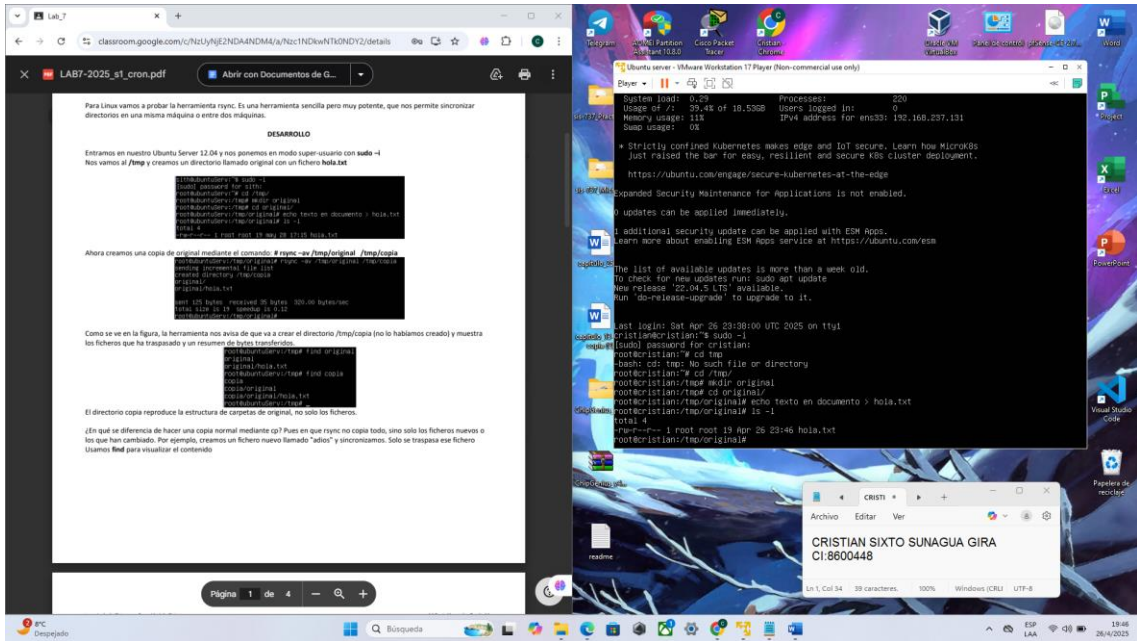
[illegible][illegible]







Lab\_7

classroom.google.com/c/NzUYqjZND4iNDM4/a/NzI1NDwNTk0NDY2/details

Ingeniería de Sistemas - Seguridad de Sistemas

M. Sc. J. Alexander Durán M.

```
root@ubuntu:/tmp# rm original/adios.txt
root@ubuntu:/tmp# rsync -av /tmp/original /tmp/copia
rsync: incremental file list
original/
original/adios.txt
sent 204 bytes received 35 bytes 388.00 bytes/sec
total size is 40 speedup is 0.25
root@ubuntu:/tmp# find /tmp/copia
/tmp/copia
/tmp/copia/original
/tmp/copia/original/hola.txt
/tmp/copia/original/adios.txt
root@ubuntu:/tmp#
```

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

### 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 --avb --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`

```
root@ubuntu:/tmp# rm original/adios.txt
root@ubuntu:/tmp# rsync --avb --delete --backup-dir=/tmp/backup1 --delete /tmp/original /tmp/copia
rsync: backup-dir is /tmp/backup1/
rsync: incremental file list
original/
original/adios.txt
sent 90 bytes received 241 bytes 662.00 bytes/sec
total size is 19 speedup is 0.06
root@ubuntu:/tmp# find /tmp/backup1
/tmp/backup1
/tmp/backup1/original
/tmp/backup1/original/hola.txt
/tmp/backup1/original/adios.txt
root@ubuntu:/tmp#
```

Ubuntu server - VMware Workstation 17 Player (Non-commercial use only)

Player

rsync --avb --delete --backup-dir=/tmp/backup1 --delete /tmp/original /tmp/copia  
rsync: incremental file list  
backup-dir is /tmp/backup1/  
data-transmission disabled for local transfer or --whole-file  
backed up original/adios.txt to /tmp/backup1/original/adios.txt  
deleting original/adios.txt  
original/hola.txt is up-to-date  
original/  
total: matches=0 hash\_hits=0 false\_alarms=0 data=0  
sent 90 bytes received 241 bytes 662.00 bytes/sec  
total size is 19 speedup is 0.06  
root@ubuntu:/tmp# find /tmp/backup1  
original  
original/hola.txt  
copia  
copia/original  
copia/original/hola.txt  
backup1  
backup1/copia  
backup1/copia/original  
backup1/copia/original/hola.txt  
backup1/copia/original/adios.txt  
backup1/original  
backup1/original/hola.txt  
backup1/original/adios.txt  
copy out /tmp/backup1/tmp/

CRISTIAN SIXTO SUNAGUA GIRA  
CI:8600448

Lab\_7

classroom.google.com/c/NzUYqjZND4iNDM4/a/NzI1NDwNTk0NDY2/details

Ingeniería de Sistemas - Seguridad de Sistemas

M. Sc. J. Alexander Durán M.

```
root@ubuntu:/tmp# rm original/adios.txt
root@ubuntu:/tmp# rsync --avb --delete --backup-dir=/tmp/backup1 --delete /tmp/original /tmp/copia
rsync: backup-dir is /tmp/backup1/
rsync: incremental file list
original/
original/adios.txt
sent 90 bytes received 241 bytes 662.00 bytes/sec
total size is 19 speedup is 0.21
root@ubuntu:/tmp# find /tmp/backup1
/tmp/backup1
/tmp/backup1/original
/tmp/backup1/original/hola.txt
/tmp/backup1/original/adios.txt
root@ubuntu:/tmp#
```

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) 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.

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

```
#!/bin/bash
# Nombre del directorio de origen
ORIG="/tmp/original"
# Nombre del directorio de destino
DEST="/tmp/backup1"
# Ejecutar el script de sincronización
rsync --avb --delete --backup-dir=$DEST --delete $ORIG $DEST
```

Procedemos a editar el cron ingresando el comando: `crontab -e`

Ubuntu server - VMware Workstation 17 Player (Non-commercial use only)

Player

#!/bin/bash  
# Nombre del directorio de origen  
ORIG="/tmp/original"  
# Nombre del directorio de destino  
DEST="/tmp/backup1"  
# Ejecutar el script de sincronización  
rsync --avb --delete --backup-dir=\$DEST --delete \$ORIG \$DEST  
/tmp/log\_mibackup

CRISTIAN SIXTO SUNAGUA GIRA  
CI:8600448

Lab\_7

Documentos de Google

classroom.google.com/c/NzUyNjZlNDI4NDM4/a/NzUyNjZlNDI4NDI2/details

Choose 1-3 [2]: 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.

```
# * * * * * /tmp/hibackup.sh
```

Si queremos ver el contenido de crontab utilizamos: **crontab -l**

A continuación, inicializamos el servicio:

```
root@ubuntu:/tmp# sudo systemctl start cron
root@ubuntu:/tmp# systemctl status cron
● cron.service
  Loaded: loaded (/etc/systemd/system/cron.service; vendor preset: enabled)
  Active: active (running) since Mon 2023-05-29 18:49:10 UTC; 1min 1s ago
  Main PID: 300
  CGroup: /systemd/systemd/cron.service
          └─ 300 /usr/sbin/cron --logtostdout

May 29 18:49:10 ubuntu systemd[1]: Starting the cron daemon...
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
```

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

```
root@ubuntu:/tmp# tail -f /var/log/syslog
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
```

Para detener el servicio utilizamos **cron stop**, caso contrario se llenará el disco de respaldos.

```
root@ubuntu:/tmp# sudo systemctl stop cron
root@ubuntu:/tmp# systemctl status cron
● cron.service
  Loaded: loaded (/etc/systemd/system/cron.service; vendor preset: enabled)
  Active: inactive (dead)
  Main PID: 300
  CGroup: /systemd/systemd/cron.service
          └─ 300 /usr/sbin/cron --logtostdout
```

Player

Ubuntu Server - VMware Workstation 17 Player (Non-commercial use only)

GNU nano 4.8 /tmp/crontab.crontab

CRISTIAN SIXTO SUNAGUA GIRA  
CI:8600448

Lab\_7

Documentos de Google

classroom.google.com/c/NzUyNjZlNDI4NDM4/a/NzUyNjZlNDI4NDI2/details

Choose 1-3 [2]: 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.

```
# * * * * * /tmp/hibackup.sh
```

Si queremos ver el contenido de crontab utilizamos: **crontab -l**

A continuación, inicializamos el servicio:

```
root@ubuntu:/tmp# sudo systemctl start cron
root@ubuntu:/tmp# systemctl status cron
● cron.service
  Loaded: loaded (/etc/systemd/system/cron.service; vendor preset: enabled)
  Active: active (running) since Mon 2023-05-29 18:49:10 UTC; 1min 1s ago
  Main PID: 300
  CGroup: /systemd/systemd/cron.service
          └─ 300 /usr/sbin/cron --logtostdout

May 29 18:49:10 ubuntu systemd[1]: Starting the cron daemon...
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
```

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

```
root@ubuntu:/tmp# tail -f /var/log/syslog
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
May 29 18:49:10 ubuntu cron[300]: (root) * * * * * /tmp/hibackup.sh
```

Para detener el servicio utilizamos **cron stop**, caso contrario se llenará el disco de respaldos.

```
root@ubuntu:/tmp# sudo systemctl stop cron
root@ubuntu:/tmp# systemctl status cron
● cron.service
  Loaded: loaded (/etc/systemd/system/cron.service; vendor preset: enabled)
  Active: inactive (dead)
  Main PID: 300
  CGroup: /systemd/systemd/cron.service
          └─ 300 /usr/sbin/cron --logtostdout
```

Player

Ubuntu Server - VMware Workstation 17 Player (Non-commercial use only)

GNU nano 4.8 /tmp/crontab.crontab

CRISTIAN SIXTO SUNAGUA GIRA  
CI:8600448





Lab\_7

Documentos de Google

classroom.google.com/c/NzUyNjZlNDI4NDM4aWNoNkN0dWNTk0NDY2/details

3. ./usr/bin/via.tiny

Choose 1-3 [2]: 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.

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

Si queremos ver el contenido de crontab utilizamos: **crontab -l**

A continuación, inicializamos el servicio:

```
root@ubuntu:/tmp# sudo systemctl daemon-reload
root@ubuntu:/tmp# systemctl enable cron.service
root@ubuntu:/tmp# systemctl start cron.service
```

Since the script you are attempting to invoke has been converted to an systemd job, you may also use the start(8) utility, e.g. start cron

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

```
root@ubuntu:/tmp# tail -f /var/log/syslog
tail: log file is empty
cron[496]: 3 root root 4096 May 26 18:49 mibackup.sh
cron[496]: 3 root root 4096 May 26 17:45 mibackup.sh
cron[496]: 1 root root 11931 May 26 23:47 log_123456789
cron[496]: 1 root root 300 May 29 00:22 log_1665290022
cron[496]: 1 root root 300 May 29 00:23 log_1665290023
cron[496]: 1 root root 300 May 29 00:24 log_1665290024
cron[496]: 1 root root 300 May 29 00:25 log_1665290025
```

Para detener el servicio utilizamos **cron stop**, caso contrario se llenará el disco de respaldos.

```
root@ubuntu:/tmp# sudo systemctl stop cron.service
root@ubuntu:/tmp# systemctl status cron.service
root@ubuntu:/tmp#
```

Since the script you are attempting to invoke has been converted to an systemd job, you may also use the stop(8) utility, e.g. stop cron

Player

Notice that tasks will be started based on the cron's system  
daemon's notion of time and timezones.  
Output of the crontab jobs (including errors) is sent through  
email to the user the crontab file belongs to (unless redirected).  
For example, you can run a backup of all user accounts  
at 5 a.m every week with:  
0 5 \* \* 1 tar -czf /var/backups/home.tar.gz /home/  
For more information see the manual pages of crontab(5) and cron(8)  
# \* \* \* \* \* /tmp/mibackup.sh  
root@ubuntu:/tmp# sudo systemctl daemon-reload  
root@ubuntu:/tmp# systemctl enable cron.service  
root@ubuntu:/tmp# systemctl start cron.service  
root@ubuntu:/tmp# tail -f /var/log/syslog  
tail: log file is empty  
cron[496]: 4 root root 4096 Apr 26 23:52 mibackup.sh  
cron[496]: 3 root root 4096 Apr 26 23:48 mibackup.sh  
cron[496]: 2 root root 127 Apr 26 23:59 mibackup.sh  
cron[496]: 2 root root 4096 Apr 26 23:51 mibackup.sh  
cron[496]: 3 root root 4096 Apr 26 23:57 mibackup.sh  
cron[496]: 3 root root 4096 Apr 26 23:57 mibackup.sh  
cron[496]: 3 root root 4096 Apr 26 23:57 mibackup.sh  
cron[496]: 3 root root 4096 Apr 26 23:57 mibackup.sh  
cron[496]: 3 root root 4096 Apr 26 23:57 mibackup.sh  
cron[496]: 2 root root 4096 Apr 26 23:57 mibackup.sh  
root@ubuntu:/tmp#

CRISTIAN SIXTO SUNAGUA GIRA  
CI:8600448  
Ln 1, Col 34 39 caracteres 100% Windows (CRLF) UTF-8

Lab\_7

Documentos de Google

classroom.google.com/c/NzUyNjZlNDI4NDM4aWNoNkN0dWNTk0NDY2/details

3. ./usr/bin/via.tiny

Choose 1-3 [2]: 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.

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

Si queremos ver el contenido de crontab utilizamos: **crontab -l**

A continuación, inicializamos el servicio:

```
root@ubuntu:/tmp# sudo systemctl daemon-reload
root@ubuntu:/tmp# systemctl enable cron.service
root@ubuntu:/tmp# systemctl start cron.service
```

Since the script you are attempting to invoke has been converted to an systemd job, you may also use the start(8) utility, e.g. start cron

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

```
root@ubuntu:/tmp# tail -f /var/log/syslog
tail: log file is empty
cron[496]: 3 root root 4096 May 26 18:49 mibackup.sh
cron[496]: 3 root root 4096 May 26 17:45 mibackup.sh
cron[496]: 1 root root 11931 May 26 23:47 log_123456789
cron[496]: 1 root root 300 May 29 00:22 log_1665290022
cron[496]: 1 root root 300 May 29 00:23 log_1665290023
cron[496]: 1 root root 300 May 29 00:24 log_1665290024
cron[496]: 1 root root 300 May 29 00:25 log_1665290025
```

Para detener el servicio utilizamos **cron stop**, caso contrario se llenará el disco de respaldos.

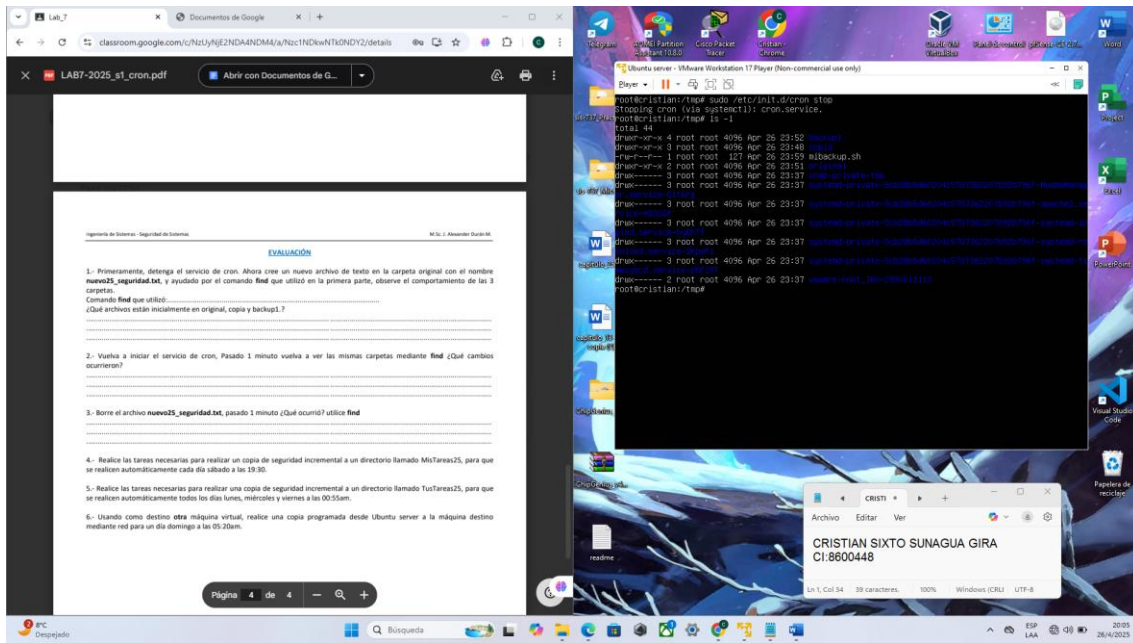
```
root@ubuntu:/tmp# sudo systemctl stop cron.service
root@ubuntu:/tmp# systemctl status cron.service
root@ubuntu:/tmp#
```

Since the script you are attempting to invoke has been converted to an systemd job, you may also use the stop(8) utility, e.g. stop cron

Player

root@ubuntu:/tmp# sudo systemctl stop cron.service  
Stopping cron (via systemctl): cron.service.  
root@ubuntu:/tmp#

CRISTIAN SIXTO SUNAGUA GIRA  
CI:8600448  
Ln 1, Col 34 39 caracteres 100% Windows (CRLF) UTF-8



## EVALUACION

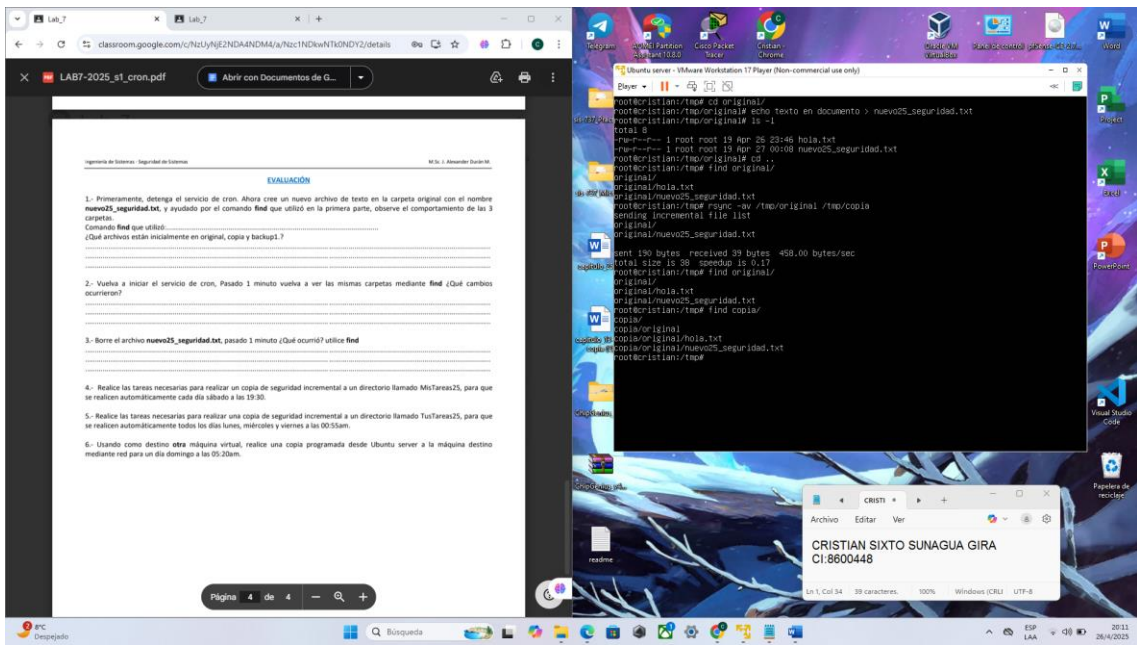
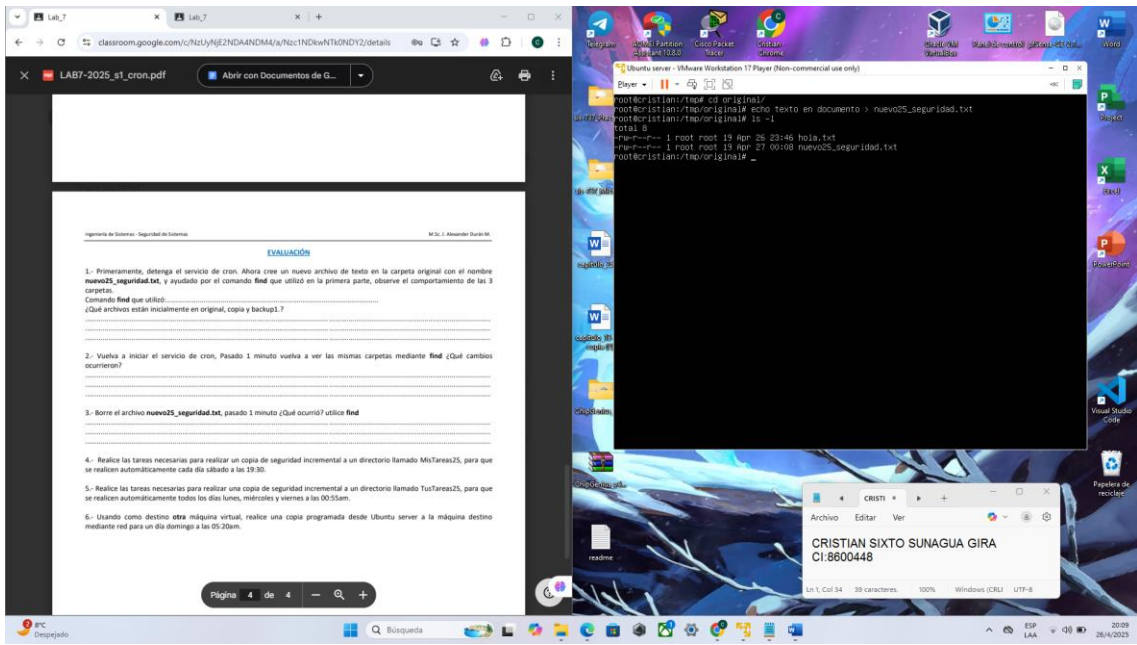
1.

Comando find original

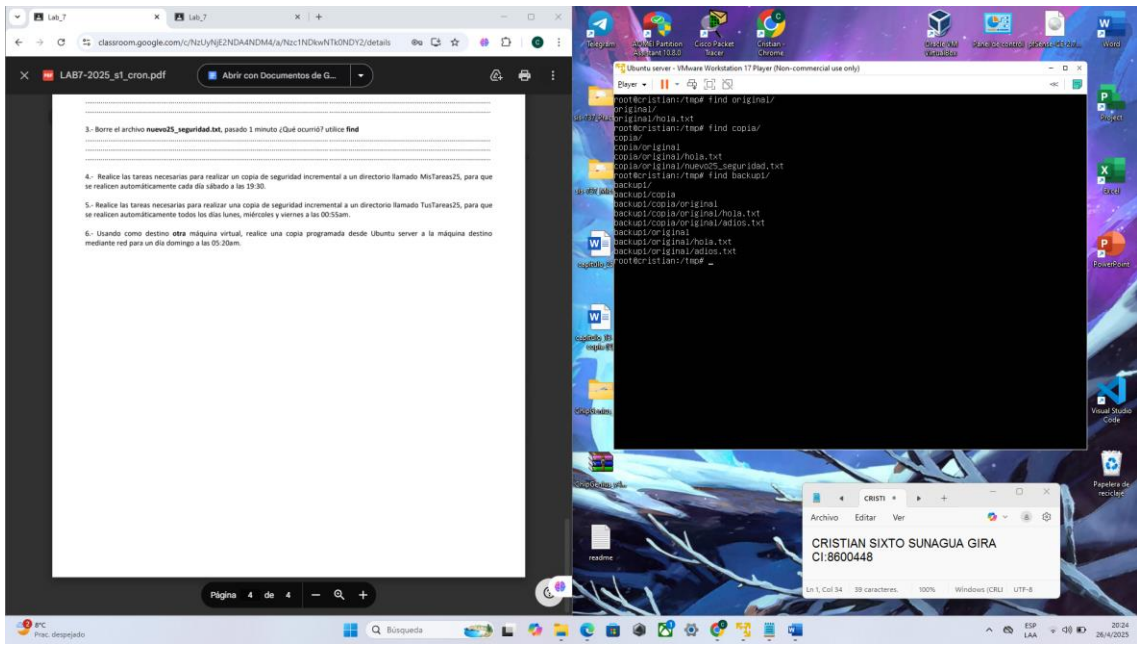
¿Qué archivos están inicialmente en original, copia y backup1?:

- original:  
hola.txt  
nuevo25\_seguridad.txt
- copia:  
original/hola.txt  
original/nuevo25\_seguridad.txt
- backup1
- copa/orignal/hola.txt/adios.txt

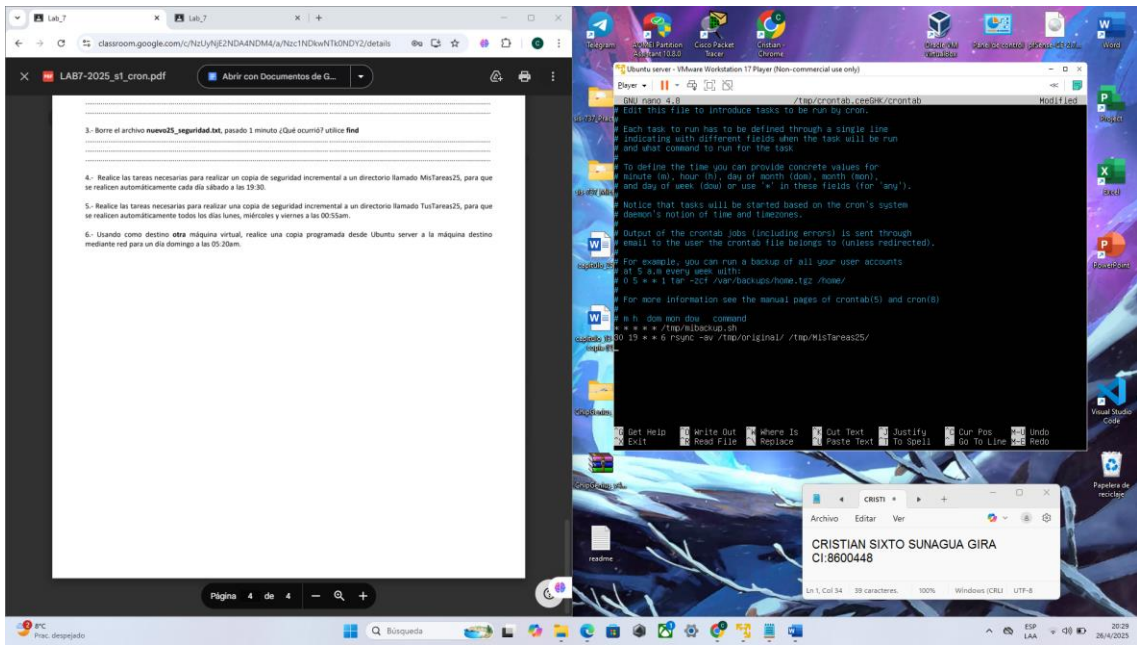




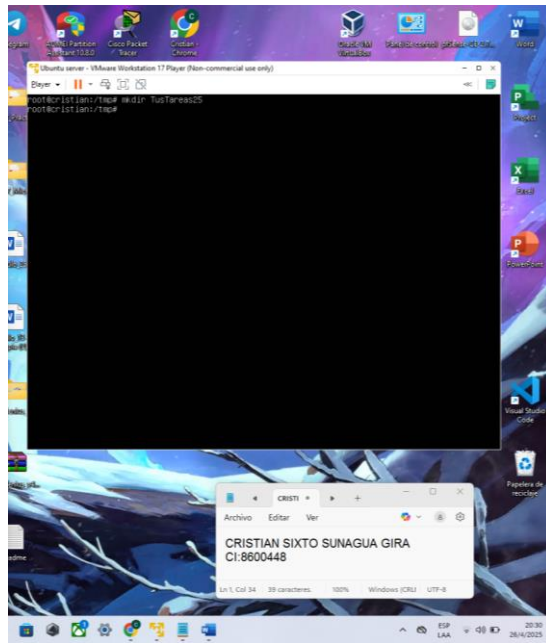
[illegible]



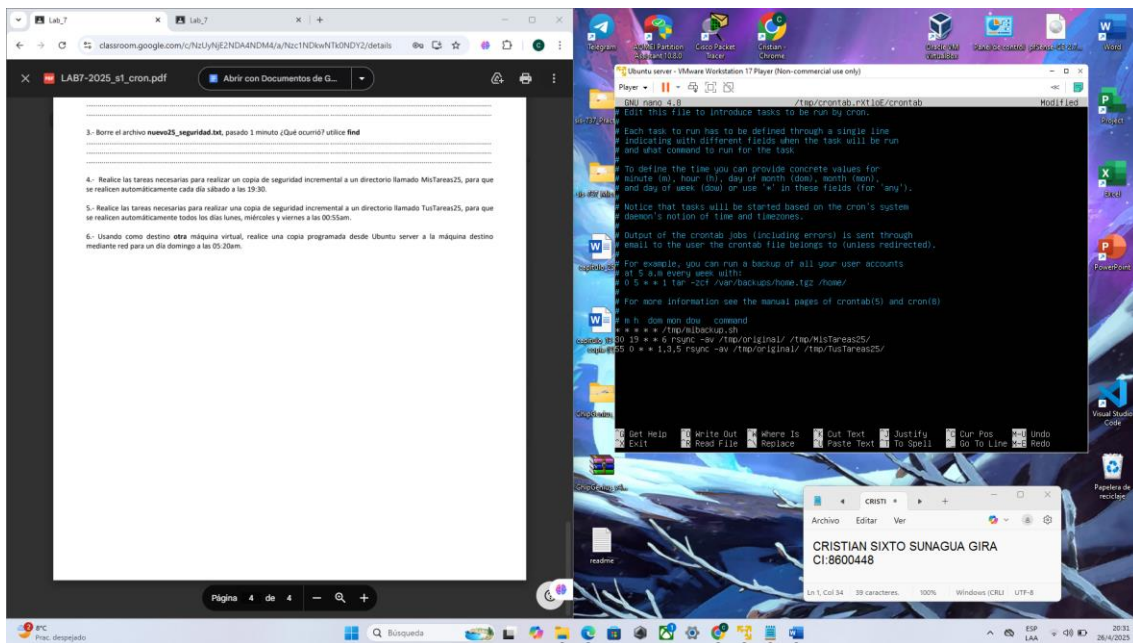
4.

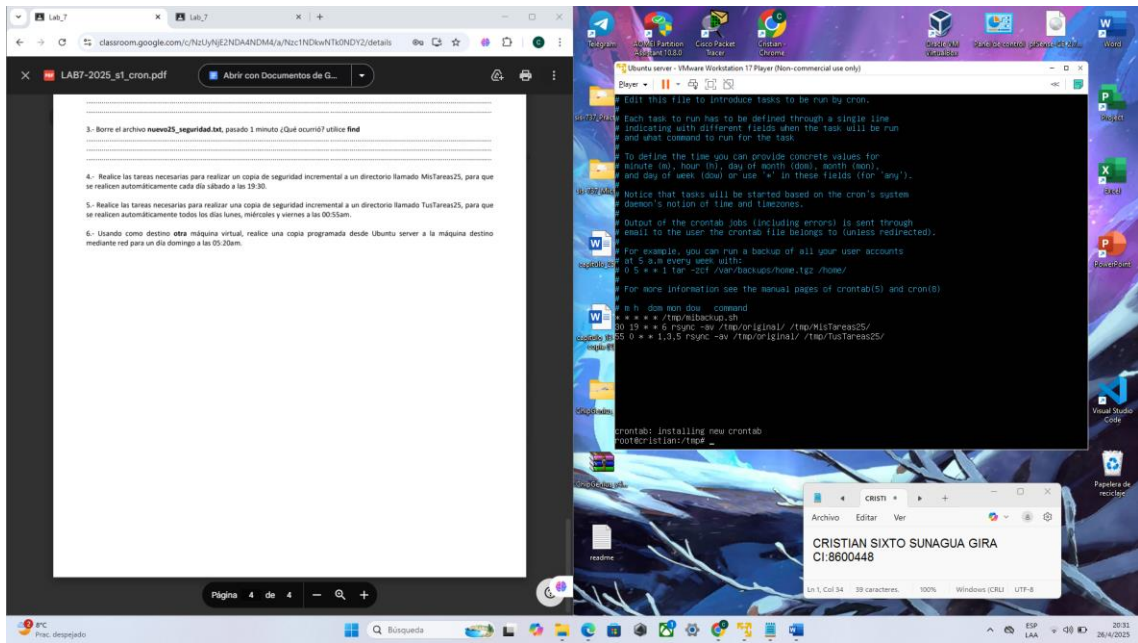




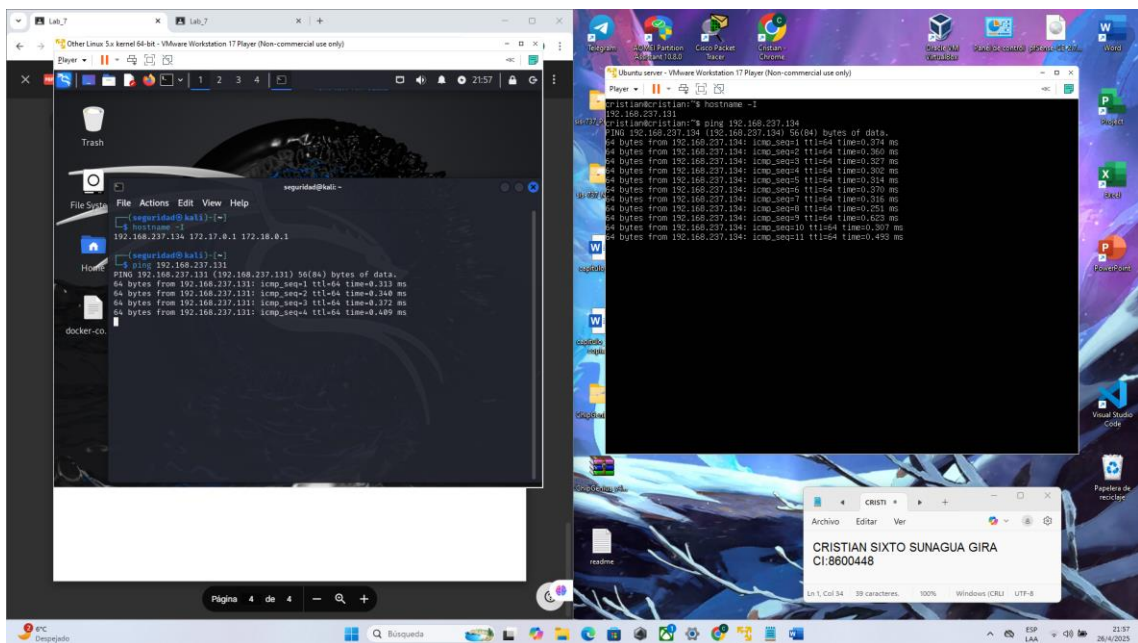


5.





## 6. Verificamos si tenemos ping entre ambas maquinas



Instalamos ssh en la maquina destino(kali) y entramos con ssh desde la maquina cliente (Ubuntu\_server)

