

Copias de Seguridad

Ismael Macareno Chouikh

2024-11-10

Índice

| | |
|--|-----------|
| 1. Instrucciones | 2 |
| 1.1. Mi nube | 2 |
| 2. Copia de Seguridad Local | 2 |
| 2.1. GNU/Linux | 2 |
| 2.1.1. Realizar la copia | 2 |
| 2.1.2. Restaurar el <i>backup</i> | 3 |
| 2.1.3. Programar el <i>backup</i> | 4 |
| 2.2. <i>Microsoft Windows</i> | 5 |
| 2.2.1. Realizar la copia | 5 |
| 2.2.2. Restaurar el <i>backup</i> | 6 |
| 2.2.3. Programar el <i>backup</i> | 7 |
| 3. Copia de Seguridad en una nube externa | 8 |
| 3.1. GNU/Linux | 8 |
| 3.1.1. Instalación | 8 |
| 3.1.2. Realizar el <i>backup</i> | 9 |
| 3.1.3. Programar el <i>backup</i> | 10 |
| 3.2. <i>Microsoft Windows</i> | 10 |
| 3.2.1. Instalación | 10 |
| 3.2.2. Realización del <i>backup</i> | 10 |
| 3.2.3. Programación del <i>backup</i> | 10 |
| 4. Valoración Personal | 10 |

1. Instrucciones

Realiza una copia de seguridad de una carpeta con todos sus archivos. Para ello planifica una política de copias de seguridad y automatiza la realización de ellas.

Prueba a realizarlo en ambos sistemas operativos (LINUX y *WINDOWS*)

Una vez realizada la copia y la automatización de la misma a nivel local, automatiza dicha copia en un sistema en la nube

1.1. Mi nube

En mi caso he decidido usar la nube [Pcloud](#)

2. Copia de Seguridad Local

2.1. GNU/Linux

En este caso usaré **rsync** que es una herramienta la cuál lo que hace es sincronizar de manera inteligente datos de manera local o remota. Compara el origen y el destino y solo transfiere las porciones de datos que han cambiado. Es una herramienta de *backup* incremental.

2.1.1. Realizar la copia

Para realizar este simulacro lo que haré será usar una máquina virtual Ubuntu 24.04 LTS con dos discos, uno el cuál contiene el SO y otro el cuál usaré para realizar el *backup*.

Lo primero que haremos será insertar el disco del *backup* y comprobar el ID o como está etiquetado.

```
maka@makaSAD:~$ ls -la /dev/ | grep -i sd
brw-rw---- 1 root disk      8,  0 Nov  8 18:32 sda
brw-rw---- 1 root disk      8,  1 Nov  8 18:32 sda1
brw-rw---- 1 root disk      8,  2 Nov  8 18:32 sda2
brw-rw---- 1 root disk      8, 16 Nov  8 18:32 sdb
```

Una vez sabemos que el nuevo disco es `/dev/sdb/` lo que haremos será averiguar dónde está montado

```
maka@makaSAD:~$ lsblk | grep sd
sda      8:0    0    65G  0 disk
sda1     8:1    0     1M  0 part
sda2     8:2    0    65G  0 part /
sdb      8:16   0    70G  0 disk
```

Como se puede apreciar arriba no está montado.

Después de las comprobaciones que hemos realizado lo que haremos será formatear el nuevo disco **sdb** en **ext4**.

¡OJO!, para esto el disco deberá estar desmontado en caso de que en las comprobaciones haya aparecido montado

```
maka@makaSAD:~$ sudo mkfs.ext4 /dev/sdb
[sudo] password for maka:
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 18350080 4k blocks and 4587520 inodes
```

```
Filesystem UUID: a2a2e49f-9585-4057-81da-9b07094d2634
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
```

Una vez formateado el disco `sdb` lo que haremos será volver a montarlo pero esta vez en un punto de montaje llamado *backup* en `/`

```
maka@makaSAD:~$ sudo mkdir /backup
maka@makaSAD:~$ sudo mount /dev/sdb /backup/
```

Después de haber montado el disco `sdb` lo que haremos será comprobar si se ha montado correctamente

```
maka@makaSAD:~$ df -Th | grep sd
/dev/sda2      ext4      64G      11G      50G      18% /
/dev/sdb       ext4      69G       24K      65G       1% /backup
```

Una vez este montado correctamente lo que haremos será hacer el *backup* de manera incremental.

Para esto, lo primero que haremos será ejecutar el siguiente comando en nuestra terminal (previamente tendremos que haber navegado a nuestro directorio raíz)

```
$ sudo rsync -aAXv / -dry-run --delete --exclude={
    /dev/*,/proc/*,/sys/*,/tmp/*,/run/*,/mnt/*,/media/*,/cdrom/*,/lost+found} /backup
```

Las opciones del comando que se pueden ver tienen las siguientes funcionalidades:

- `-a`: modo archivo
- `-A`: mantiene el *access control list*
- `-X`: mantiene los atributos de los ficheros
- `-v`: verbose
- `--delete`: está es la opción que hace que la copia sea incremental
- `--dry-run`: simula el proceso de *backup*
- `--exclude`: excluye los directorios dados de la copia de seguridad

2.1.2. Restaurar el *backup*

Lo primero que haremos para restaurar el *backup* es crear dos directorios

- `/mnt/drive/`
- `/mnt/system/`

```
maka@makaSAD:/$ sudo mkdir /mnt/drive
[sudo] password for maka:
maka@makaSAD:/$ sudo mkdir /mnt/system
```

Luego de haber creado los dos directorios lo que haremos será montar el disco **sdb** que se supone que está simulando un dispositivo de almacenamiento externo

```
maka@makaSAD:/$ sudo mount /dev/sdb /mnt/drive/
```

Por último, ejecutaremos el siguiente comando para restaurar

```
maka@makaSAD:/$ sudo rsync -aXv -delete /mnt/drive /mnt/system
```

2.1.3. Programar el *backup*

Para programar el *backup* CLI en GNU/Linux se suele usar el **cron**.

Para programar el *backup* lo que haremos será, primero acceder al crontab con el comando **crontab -e** y elegiremos el editor que más nos guste. En mi caso usaré **vi**.

Una vez en el crontab añadiremos la siguiente línea

```
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 8,16,23 * * * sudo rsync -aXv / -dry-run --delete -exclude={
    /dev/*,/proc/*,/sys/*,/tmp/*,/run/*,/mnt/*,/media/*,/cdrom/*,/lost+found} /backup
```

Para entender la línea añadida en el crontab necesitaremos saber la estructura de crontab

```
# |----- minute (0 - 59)
# |----- hour (0 - 23)
# |----- day of the month (1 - 31)
# |----- month (1 - 12)
# |----- day of the week (0 - 6) (Sunday to Saturday;
# |                               7 is also Sunday on some systems)
#
# * * * * * command to execute
```

Figura 1: Macareno, Ismael. (2024). Estructura Crontab [JPG]. Hostinger

2.2. Microsoft Windows

2.2.1. Realizar la copia

Para realizar el *backup* en *Microsoft Windows* lo primero que tendremos que hacer será añadir un dispositivo de almacenamiento a nuestra máquina virtual de Windows 10 con un poco más de tamaño que el dispositivo el cuál contiene el sistema operativo. En mi caso he añadido un dispositivo de 60 GiBi.

Luego de haber añadido el dispositivo de almacenamiento lo que tendremos que hacer será arrancar la máquina virtual y acceder al administrador de dispositivos para poder inicializar el disco como GPT.

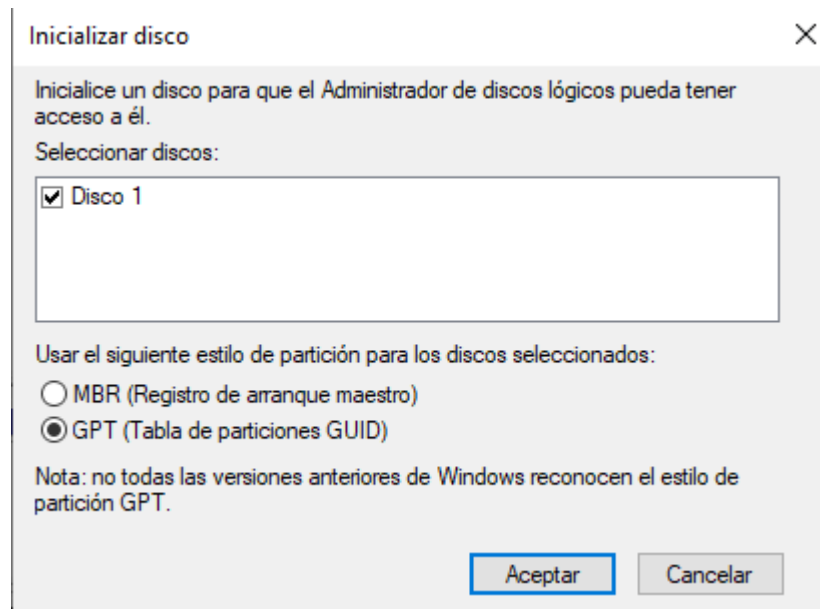


Figura 2: Macareno, Ismael. (2024). Inicializar disco como GPT [PNG]. Propia

Luego de haber inicializado el disco lo que tendremos que hacer será crear un nuevo volumen simple, para ello lo que haremos será hacer clic derecho sobre el disco inicializado y seguidamente hacer clic en nuevo volumen simple.

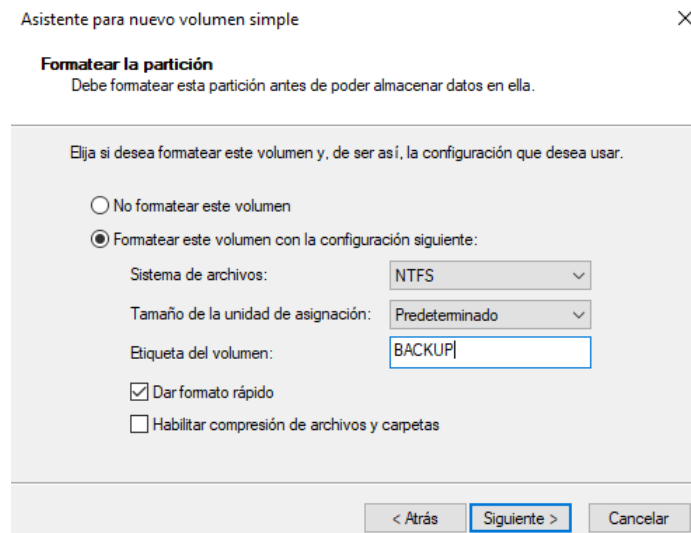


Figura 3: Macareno, Ismael. (2024). Creación nuevo volumen simple NTFS [PNG]. Propia

Luego de haber creado el volumen simple lo que haremos será realizar la copia de seguridad.

Para realizar la copia de seguridad lo que tendremos que hacer será ir a:

1. Configuración
2. Actualizaciones y seguridad
3. Copia de seguridad
4. Hacer clic en añadir una unidad

Copia de seguridad

Copia de seguridad con Historial de archivos

Realiza una copia de seguridad de tus archivos en otra unidad y restáuralos si los originales se han perdido, están dañados o se han eliminado.

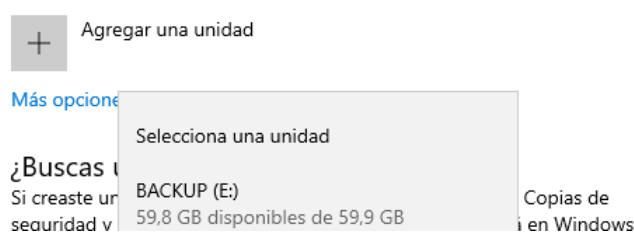


Figura 4: Macareno, Ismael. (2024). Añadir unidad para *backup* [PNG]. Propia

Luego de haber añadido la unidad para que se realicen los *backups* podremos hacer clic en **más opciones** para poder configurar a más profundidad las copias de seguridad

Opciones de copia de seguridad

Información general

Tamaño de la copia de seguridad: 0 bytes
Espacio total en BACKUP (E:) (E:): 59,9 GB
Haciendo copia de seguridad de los datos...

Hacer ahora una copia de seguridad

Realizar una copia de seguridad de mis archivos

Cada hora (predeterminado)

Mantener las copias de seguridad

Para siempre (predeterminado)

Hacer una copia de seguridad de estas carpetas

Agregar una carpeta

Juegos guardados
C:\Users\alumno

Vínculos
C:\Users\alumno

Descargas
C:\Users\alumno

Favoritos
C:\Users\alumno

Contactos

Figura 5: Macareno, Ismael. (2024). Más opciones copia de seguridad Windows [PNG]. Propia

2.2.2. Restaurar el *backup*

Para restaurar el *backup* simplemente lo que tendremos que hacer será ir a

1. Configuración
2. Actualizaciones y seguridad
3. Copias de seguridad
4. Más opciones
 - Restaurar archivos desde una copia de seguridad actual

Opciones de configuración relacionadas

[Ver la configuración avanzada](#)

[Restaurar archivos desde una copia de seguridad actual](#)

Figura 6: Macareno, Ismael. (2024). Recuperar archivos desde una copia de seguridad actual [PNG]. Propia

Luego de haber hecho clic en la opción mencionada anteriormente lo que nos aparecerá será algo parecido a lo siguiente

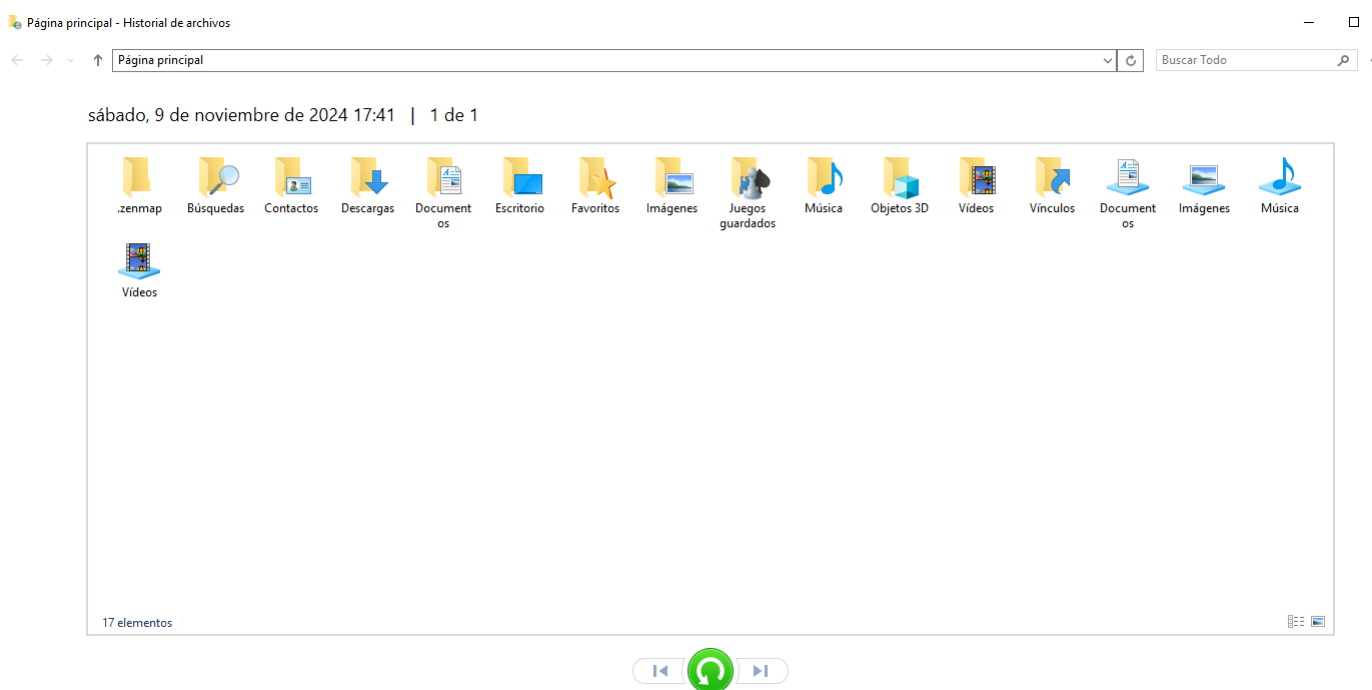


Figura 7: Macareno, Ismael. (2024). Ejemplo ventana restauración de una copia de seguridad [PNG]. Propia

Simplemente tendremos que hacer clic en el botón verde que se puede apreciar en la imagen

2.2.3. Programar el *backup*

Para programar el *backup* tendremos que volver al mismo sitio en el cuál hemos estado realizando los dos apartados anteriores.

1. Configuración
2. Actualizaciones y seguridad

3. Copias de seguridad
4. Más opciones
 - Configurar cada cuanto tiempo se realizará el *backup*

🏠 Opciones de copia de seguridad

Información general

Tamaño de la copia de seguridad: 119 MB

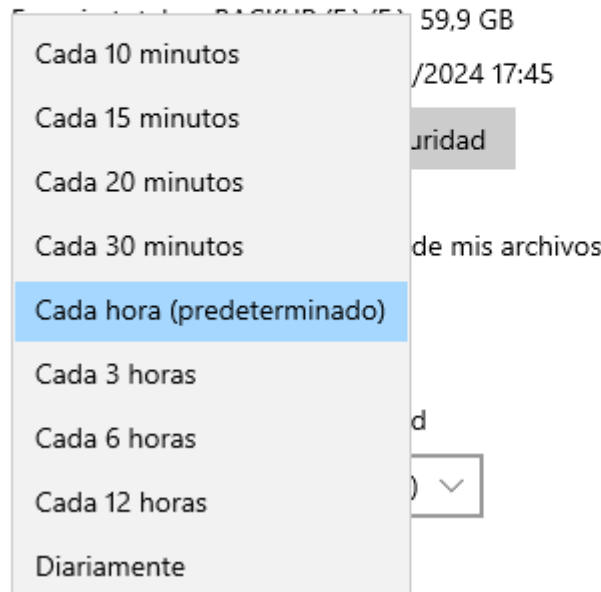


Figura 8: Macareno, Ismael. (2024). Programación de cada cuanto tiempo se realiza el *backup* [PNG]. Propia

3. Copia de Seguridad en una nube externa

3.1. GNU/Linux

3.1.1. Instalación

Para poder crear un *backup* en pcloud lo primero que tendremos que hacer será hacernos una cuenta en esta nube. Está cuenta nos dará un almacenamiento de 7 GiBi, obviamente no es suficiente almacenamiento para un dispositivo de *backups* pero lo usaremos para la prueba.

Después de habernos creado una cuenta en pcloud lo que haremos será abrir nuestra MV de Ubuntu 24.04 LTS e introducir los siguientes comandos:

- `sudo add-apt-repository universe`
- `sudo apt install libfuse2`

Después de haber ejecutado esos dos comandos lo que tendremos que hacer será descargarnos el paquete de pcloud para poder instalarlo. En mi caso he ejecutado el siguiente comando

- `wget -S https://linuxstans.com/goto/pcloud-download-64bit/`

Cuando tengamos el paquete instalado tendremos que darle permisos de ejecución mediante el comando `chmod`

- `chmod u+x pcloud`

Luego de haberle dado permisos de ejecución lo que tendremos que hacer será ejecutar `pcloud`

- `./pcloud`

Cuando lo ejecutemos se nos abrirá una ventana la cuál nos pedirá que iniciemos sesión con nuestra cuenta de `pcloud`

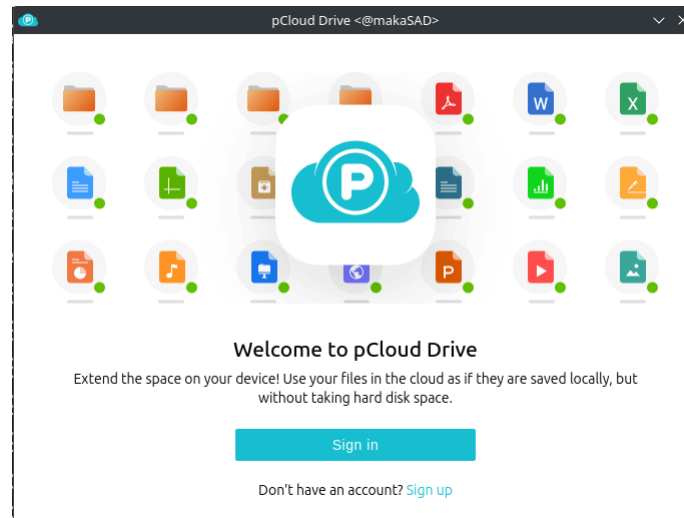


Figura 9: Macareno, Ismael. (2024). *Login* de pcloud Ubuntu 24.04 LTS [PNG]. Propia

Cuando iniciemos sesión lo único que ocurrirá es que se nos abrirá nautilus con el *drive* de pcloud

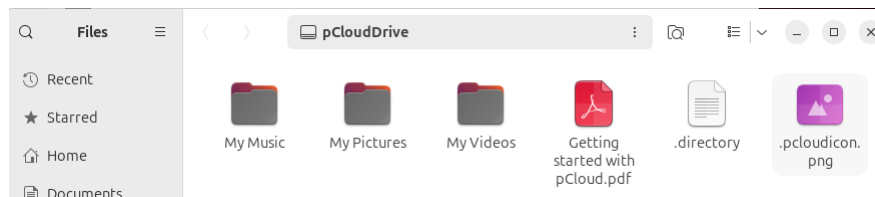


Figura 10: Macareno, Ismael. (2024). Pcloud *drive* Ubuntu 24.04 LTS [PNG]. Propia

3.1.2. Realizar el *backup*

Para realizar el *backup* usaremos el comando `tar`.

Lo primero que haremos será crear un directorio en nuestro *drive* de pcloud llamado *backupUbuntu*

```
maka@makaSAD:~/pCloudDrive$ mkdir backupUbuntu
maka@makaSAD:~/pCloudDrive$ ls -la
total 12
drwxr-xr-x  3 maka maka 4096 Nov  9 19:02 .
drwxr-x--- 23 maka maka 4096 Nov  9 18:46 ..
drwxrwxr-x  2 maka maka 4096 Nov  9 19:02 backupUbuntu
```

Luego de haber creado el directorio anterior lo único que tendremos que hacer es ejecutar el siguiente comando

- `tar -cf /home/maka/pCloudDrive/backupUbuntu/backup.tgz --directory=/ .`

El resultado será algo parecido a lo siguiente

```
maka@makaSAD:~/pCloudDrive$ cd backupUbuntu/
maka@makaSAD:~/pCloudDrive/backupUbuntu$ ls -la
total 46203732
drwxrwxr-x 2 maka maka      4096 Nov 10 14:04 .
drwxr-xr-x 3 maka maka      4096 Nov 10 14:04 ..
-rw-r--r-- 1 root root 47312515072 Nov 10 14:02 backupUbu.tgz
```

3.1.3. Programar el *backup*

Al igual que lo hicimos en el apartado 2 esto se hace mediante el `cron`.

1. Editar el `cron`

- `crontab -e`

2. Añadir lo siguiente

```
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 8,16,23 * * * tar -cf /home/maka/pCloudDrive/backupUbuntu/backup.tgz --directory=/ .
```

3.2. Microsoft Windows

3.2.1. Instalación

Para instalar `pcloud` en *Microsoft Windows* lo único que tendremos que hacer será acceder a la siguiente URL

- <https://www.pcloud.com/how-to-install-pcloud-drive-windows.html?download=windows-10-64bit>

Una vez descargado el `.exe` lo único que tendremos que hacer será ejecutarlo e iniciar sesión en nuestra cuenta.

3.2.2. Realización del *backup*

Seguiremos los mismos pasos que en el apartado 2.2.1 de copias de seguridad en local pero en vez de elegir como unidad donde se almacenaran las copias de seguridad otro disco, elegiremos el *pCloudDrive*.

3.2.3. Programación del *backup*

De la misma manera que en el apartado 2.2.3 de esta práctica.

4. Valoración Personal

Práctica bastante interesante y poco problemática (en mi caso).

Siempre está muy bien saber de copias de seguridad tanto en local como en remoto.