

Tarea 1

Sistemas operativos

Brayan Poloche- Karen Garcia

- Paso a paso de como se instaló las máquinas virtuales en qemu.

- CentOS:

1. Se descarga la ISO de CentOS desde <https://www.centos.org/download/>.

2. Crear la imagen de disco es necesario porque QEMU no proporciona automáticamente un disco duro virtual. Es el disco en el que se instalará el sistema operativo.

```
brayan@bayan- laptop:~$ qemu-img create -f qcow2  
/home/brayan/Escritorio/centos.qcow2 20G
```

- `qemu-img create` → Crea un disco virtual en formato qcow2.
- `-f qcow2` → Especifica el formato `qcow2` en lugar de `raw`.
- `/home/brayan/Escritorio/centos.qcow2` → Ruta y nombre del archivo de la imagen de disco.
- `20G` → Capacidad del disco (en GB).

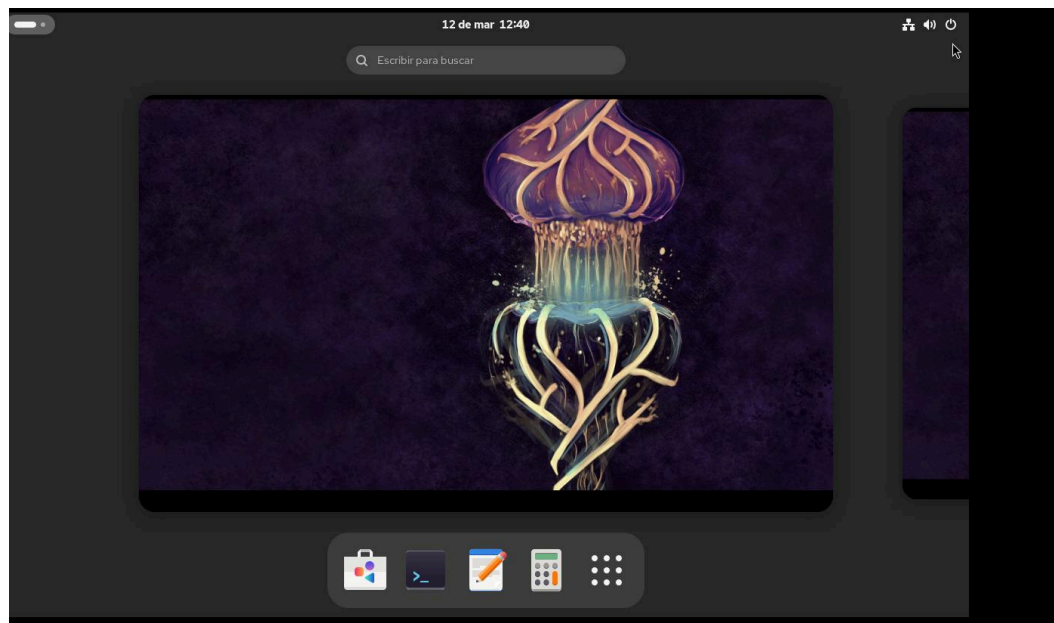
3. Instalar CentOS

```
brayan@bayan- laptop:~$ qemu-system-x86_64 \ # Ejecuta QEMU para una  
arquitectura x86_64.  
-enable-kvm \ #Habilita la aceleración KVM para un mejor rendimiento.  
-m 4G \ #Asigna 4GB de RAM a la VM.  
-cpu host \ # Usa la CPU física del host para un mejor rendimiento.  
-smp 4 \ # Asigna 4 núcleos de CPU.  
-hda /home/brayan/Escritorio/centos.qcow2 \ #Especifica el disco virtual donde  
se instalará CentOS.  
-cdrom /home/brayan/Descargas/CentOS-Stream-9-latest-x86_64-dvd1.iso \  
#Monta la ISO de instalación.  
-boot d \ #Arranca desde el CD-ROM (ISO) en lugar del disco duro.  
-vga qxl \ # Mejora la compatibilidad gráfica.  
-netdev bridge,id=net0,br=virbr0 -device virtio-net-pci,netdev=net0 #Configura  
que la máquina virtual tenga acceso a la red desde el inicio.
```

4. Iniciar CentOS en Qemu

```
brayan@brayan-laptop:~$ qemu-system-x86_64 \
-enable-kvm \
-m 4G \
-cpu host \
-smp 4 \
-hda /home/brayan/Escritorio/centos.qcow2 \
-vga qxl
-netdev bridge,id=net0,br=virbr0 -device virtio-net-pci,netdev=net0
```

- `qemu-system-x86_64` → Ejecuta QEMU para una arquitectura x86_64.
- `enable-kvm` → Habilita la aceleración KVM para mejor rendimiento.
- `m 4G` → Asigna 4GB de RAM a la VM.
- `cpu host` → Usa la CPU física del host para mejor rendimiento.
- `smp 4` → Asigna 4 núcleos de CPU.
- `hda /home/brayan/Escritorio/centos.qcow2` → Usa el archivo centos.qcow2 como disco duro virtual.
- `vga qxl` → Mejora la compatibilidad gráfica.
- `netdev bridge,id=net0,br=virbr0 -device virtio-net-pci,netdev=net0` → Configura que la máquina virtual tenga acceso a la red desde el inicio.



- **Windows 10:**

1. Se descarga la ISO de Windows 10 desde <https://www.microsoft.com/es-es/software-download/windows10>.

2. Crear la imagen de disco es necesario porque QEMU no proporciona automáticamente un disco duro virtual. Es el disco en el que se instalará el sistema operativo.

```
brayan@brayan-laptop:~$ qemu-img create -f qcow2  
/home/brayan/Escritorio/windows/windows10.qcow2 30G
```

- `qemu-img create` → Crea un disco virtual en formato qcow2.
- `-f qcow2` → Especifica el formato `qcow2` en lugar de `raw`.
- `/home/brayan/Escritorio/windows/windows10.qcow2` → Ruta y nombre del archivo de la imagen de disco.
- `30G` → Capacidad del disco (en GB).

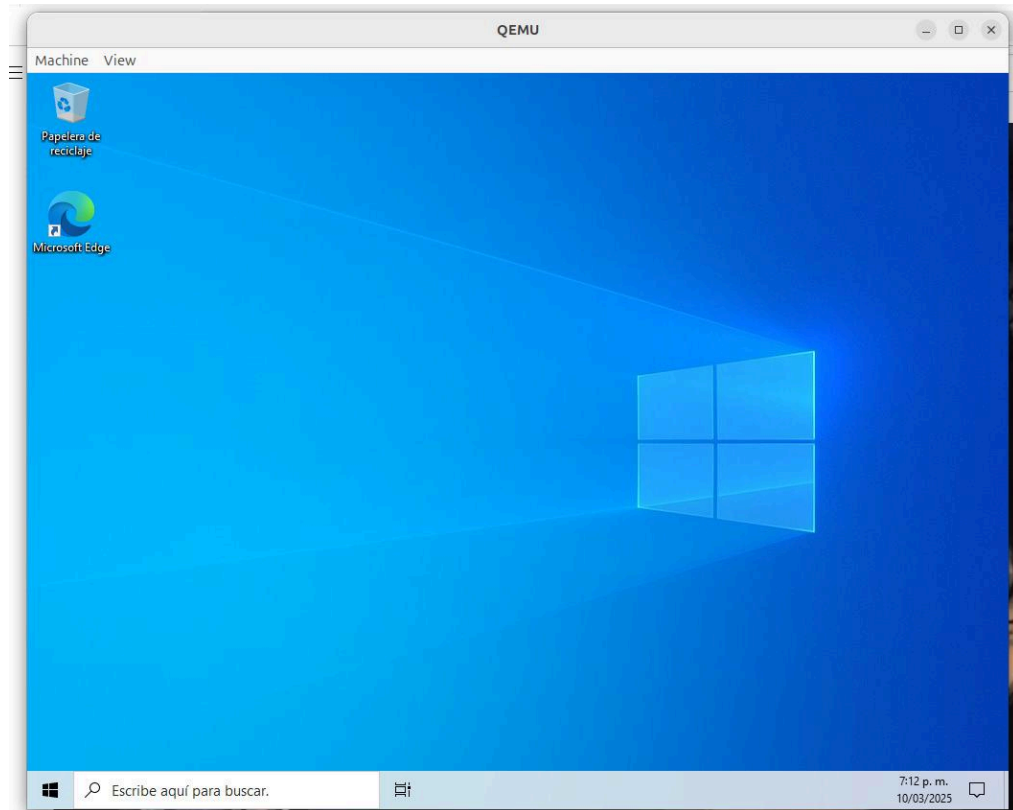
3. Instalar Windows 10

```
brayan@brayan-laptop:~$ qemu-system-x86_64 \ # Ejecuta QEMU para una  
arquitectura x86_64.  
-enable-kvm \ #Habilita la aceleración KVM para un mejor rendimiento.  
-m 4G \ #Asigna 4GB de RAM a la VM.  
-cpu host \ # Usa la CPU física del host para un mejor rendimiento.  
-smp 4 \ # Asigna 4 núcleos de CPU.  
-hda /home/brayan/Escritorio/windows/windows10.qcow2 \#Especifica el disco  
virtual donde se instalará Windows.  
-cdrom /home/brayan/Escritorio/windows/Win10_22H2_Spanish_x64v1.iso \  
#Monta la ISO de instalación.  
-boot d \ #Arranca desde el CD-ROM (ISO) en lugar del disco duro.  
-vga qxl \ # Mejora la compatibilidad gráfica.  
-netdev bridge,id=net0,br=virbr0 -device virtio-net-pci,netdev=net0 #Configura  
que la máquina virtual tenga acceso a la red desde el inicio.
```

4. Iniciar Windows 10 en Qemu

```
brayan@brayan-laptop:~$ qemu-system-x86_64 \  
-enable-kvm \  
-m 4G \  
-cpu host \  
-smp 4 \  
-hda /home/brayan/Escritorio/windows/windows10.qcow2 \  
-vga qxl \  
-netdev bridge,id=net0,br=virbr0 -device virtio-net-pci,netdev=net0
```

- `qemu-system-x86_64` → Ejecuta QEMU para una arquitectura x86_64.
- `enable-kvm` → Habilita la aceleración KVM para mejor rendimiento.
- `m 4G` → Asigna 4GB de RAM a la VM.
- `cpu host` → Usa la CPU física del host para mejor rendimiento.
- `smp 4` → Asigna 4 núcleos de CPU.
- `hda /home/brayan/Escritorio/windows/windows10.qcow2` → Usa el archivo windows.qcow2 como disco duro virtual.
- `vga qxl` → Mejora la compatibilidad gráfica.
- `netdev bridge,id=net0,br=virbr0 -device virtio-net-pci,netdev=net0` → Configura que la máquina virtual tenga acceso a la red desde el inicio.



- Scientific Linux:

1. Se descarga la ISO de Scientific Linux desde

<https://scientificlinux.org/downloads/>

2. Crear la imagen de disco es necesario porque QEMU no proporciona automáticamente un disco duro virtual. Es el disco en el que se instalará el sistema operativo.

brayan@bayan- laptop:~\$ qemu-img create -f qcow2
/home/brayan/Escritorio/scientific-linux.qcow2 20G

- **qemu-img create** → Crea un disco virtual en formato qcow2.
- **-f qcow2** → Especifica el formato **qcow2** en lugar de **raw**.
- **/home/brayan/Escritorio/scientific-linux.qcow2** → Ruta y nombre del archivo de la imagen de disco.
- **20G** → Capacidad del disco (en GB).

3. Instalar Scientific Linux

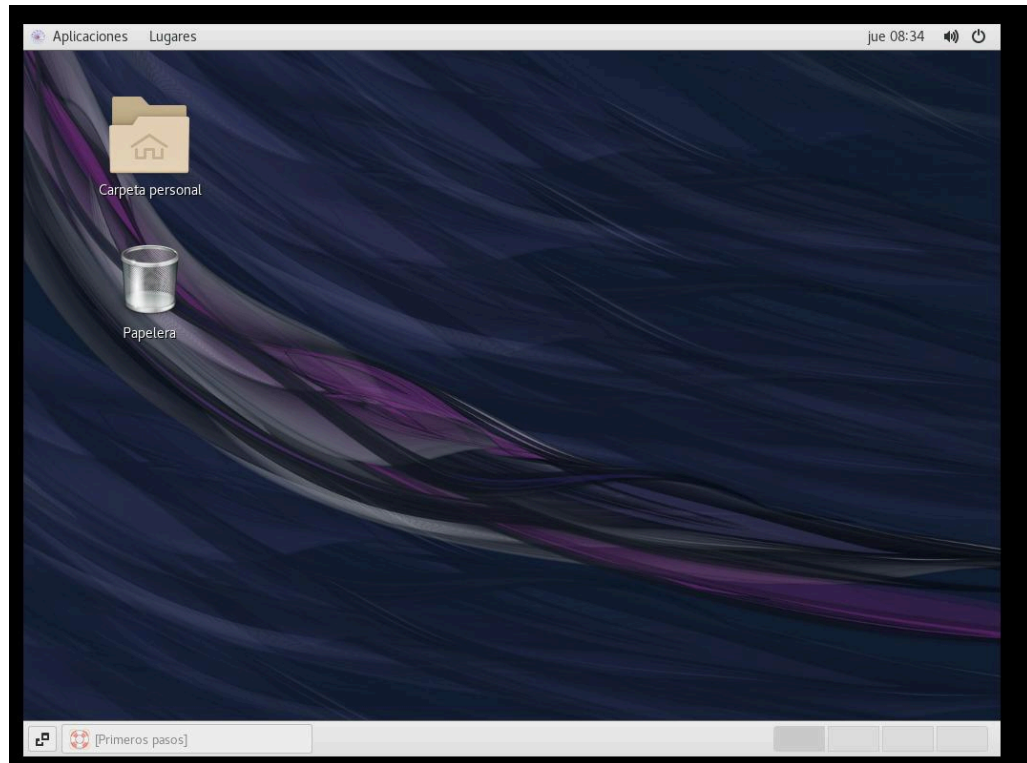
brayan@bayan- laptop:~\$ qemu-system-x86_64 \ # Ejecuta QEMU para una arquitectura x86_64.
-enable-kvm \ #Habilita la aceleración KVM para un mejor rendimiento.
-m 4G \ #Asigna 4GB de RAM a la VM.
-cpu host \ # Usa la CPU física del host para un mejor rendimiento.
-smp 4 \ # Asigna 4 núcleos de CPU.
-hda /home/brayan/Escritorio/scientific-linux.qcow2 \ #Especifica el disco virtual donde se instalará Scientific-linux.
-cdrom /home/brayan/Escritorio/ScientificLinux-7.9-x86_64-DVD.iso \ #Monta la ISO de instalación.
-boot d \ #Arranca desde el CD-ROM (ISO) en lugar del disco duro.
-vga qxl \ # Mejora la compatibilidad gráfica.
-netdev bridge,id=net0,br=virbr0 -device virtio-net-pci,netdev=net0 #Configura que la máquina virtual tenga acceso a la red desde el inicio.

4. Iniciar Scientific Linux en Qemu

```
brayan@brayan-laptop:~$ qemu-system-x86_64 \
-enable-kvm \
-m 4G \
-cpu host \
-smp 4 \
-hda /home/brayan/Escritorio/scientific-linux.qcow2 \
-vga qxl \
-netdev bridge,id=net0,br=virbr0 -device virtio-net-pci,netdev=net0
```

- **qemu-system-x86_64** → Ejecuta QEMU para una arquitectura x86_64.
- **enable-kvm** → Habilita la aceleración KVM para mejor rendimiento.
- **m 4G** → Asigna 4GB de RAM a la VM.
- **cpu host** → Usa la CPU física del host para mejor rendimiento.
- **smp 4** → Asigna 4 núcleos de CPU.

- `hda /home/brayan/Escritorio/scientific-linux` → Usa el archivo `scientific-linux.qcow2` como disco duro virtual.
- `vga qxl` → Mejora la compatibilidad gráfica.
- `netdev bridge,id=net0,br=virbr0 -device virtio-net-pci,netdev=net0` → Configura que la máquina virtual tenga acceso a la red desde el inicio.



- **Nmap para escanear la red interna y externa.**

1. Se instaló Nmap con el comando:

`brayan@bayan- laptop:~$ sudo apt update && sudo apt install nmap -y`

- `sudo apt update` → Actualiza la lista de paquetes disponibles.
 - `sudo apt install nmap -y` → Instala Nmap automáticamente sin pedir confirmación.
2. Se realiza el escaneo de manera local cuando las máquinas virtuales están encendidas:


```
brayan@brayan-laptop:~$ sudo nmap -sn 192.168.122.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-13 11:49 -05
Nmap scan report for 192.168.122.127
Host is up (0.00025s latency).
MAC Address: 52:54:00:36:CB:72 (QEMU virtual NIC)
Nmap scan report for brayan-laptop (192.168.122.1)
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 4.09 seconds
```

3. Se realiza el escaneo de manera global para visualizar que ve el pc de manera global:

```
brayan@brayan-laptop:~$ sudo nmap -A scanme.nmap.org
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-13 12:01 -05
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.14s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
| 2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
| 256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_ 256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
|_ http-favicon: Nmap Project
|_ http-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: Go ahead and ScanMe!
646/tcp   filtered ldap
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.4
Network Distance: 17 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 1723/tcp)
HOP RTT ADDRESS
1 0.27 ms _gateway (10.254.121.1)
2 1.12 ms 201-234-142-193.static.ipsat.net.ar (201.234.142.193)
3 1.41 ms 201-234-142-176.static.ipsat.net.ar (201.234.142.176)
4 ...
5 11.44 ms 4.68.59.153
6 ...
7 64.56 ms 4.15.158.206
8 64.65 ms ae11.r01.mia01.icn.netarch.akamai.com (23.207.239.34)
9 103.13 ms ae24.r22.dfw01.icn.netarch.akamai.com (23.32.62.96)
10 134.70 ms ae20.r01.lax01.icn.netarch.akamai.com (23.32.62.90)
11 141.51 ms ae18.r02.sjc01.icn.netarch.akamai.com (23.193.113.13)
12 140.95 ms ae2.r11.sjc01.iem.netarch.akamai.com (23.207.232.39)
13 141.78 ms ae22.gw4.scz1.netarch.akamai.com (23.203.158.53)
```

```

Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|_ 2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|_ 256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_ 256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp    open  http      Apache httpd 2.4.7 ((Ubuntu))
|_ http-favicon: Nmap Project
|_ http-server-header: Apache/2.4.7 (Ubuntu)
|_ http-title: Go ahead and ScanMe!
646/tcp   filtered ldp
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.4
Network Distance: 17 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 1723/tcp)
HOP RTT ADDRESS
1 0.27 ms _gateway (10.254.121.1)
2 1.12 ms 201-234-142-193.static.imsat.net.ar (201.234.142.193)
3 1.41 ms 201-234-142-176.static.imsat.net.ar (201.234.142.176)
4 ...
5 11.44 ms 4.68.59.153
6 ...
7 64.56 ms 4.15.158.206
8 64.65 ms ae11.r01.mia01.icn.netarch.akamai.com (23.207.239.34)
9 103.13 ms ae24.r22.dfw01.icn.netarch.akamai.com (23.32.62.96)
10 134.70 ms ae20.r01.lax01.icn.netarch.akamai.com (23.32.62.90)
11 141.51 ms ae18.r02.sjc01.icn.netarch.akamai.com (23.193.113.13)
12 140.95 ms ae2.r11.sjc01.iem.netarch.akamai.com (23.207.232.39)
13 141.78 ms ae22.gw4.scz1.netarch.akamai.com (23.203.158.53)
14 ... 16
17 141.43 ms scanme.nmap.org (45.33.32.156)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit
Nmap done: 1 IP address (1 host up) scanned in 19.60 seconds

```

- Tareas programadas con Cron y Crontab.

1. Se abre el archivo de configuración de cron del usuario.

```

brayan@brayan-laptop:~$ crontab -e
crontab: installing new crontab

```

2. Dentro de él se ingresan las tareas.

```

30 12 * * * virsh shutdown scientific-linux && virsh shutdown centos && virsh shutdown windows 10
30 12 * * * nmap -sn 192.168.122.0/24 > ~/nmap_scan.log
30 12 * * * cd ~/Tarea1 && git add . && git commit -m "Backup automático" && git push

```

- La primera apaga tres máquinas virtuales llamadas **scientific-linux**, **centos** y **windows 10**, usando el comando **virsh shutdown** el cual es el que apaga.
- El segundo ejecuta un escaneo de red en la subred **192.168.122.0/24** usando **nmap**, detectando dispositivos activos, y guarda los resultados en el archivo **nmap_scan.log**.
- El tercero: Cambia al directorio **Tarea1**, agrega todos los archivos modificados a Git, crea un commit con el mensaje "Backup automático" && git push

automático" y envía los cambios al repositorio remoto en la rama **main**.

3. Confirmación de que las tareas programadas en crontab se ejecutaron.

```
brayan@brayan-laptop:~$ grep CRON /var/log/syslog | grep "12:30"
2025-03-13T12:30:01.154708-05:00 brayan-Laptop CRON[33513]: (root) CMD ([ -x /etc/init.d/anacron ] && if [ ! -d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron start >/dev/null; fi)
2025-03-13T12:30:01.180592-05:00 brayan-Laptop CRON[33515]: (brayan) CMD (sudo /home/SUSER/Brayan_y_Karen/backup_manager.sh)
2025-03-13T12:30:01.180926-05:00 brayan-Laptop CRON[33516]: (brayan) CMD (nmap -sn 192.168.122.0/24 > ~/nmap_scan.log)
2025-03-13T12:30:01.181154-05:00 brayan-Laptop CRON[33517]: (brayan) CMD (cd ~/Tarea1 && git add . && git commit -m "Backup autom\303\241tico" && git push origin main)
2025-03-13T12:30:01.181367-05:00 brayan-Laptop CRON[33518]: (brayan) CMD (virsh shutdown scientific-linux && virsh shutdown centos && virsh shutdown windows10)
2025-03-13T12:30:01.181613-05:00 brayan-Laptop CRON[33519]: (brayan) CMD (sudo /home/SUSER/Brayan_y_Karen/backup_manager.sh)
2025-03-13T12:30:01.183670-05:00 brayan-Laptop CRON[33508]: (CRON) info (No MTA installed, discarding output)
2025-03-13T12:30:01.190464-05:00 brayan-Laptop CRON[33511]: (CRON) info (No MTA installed, discarding output)
2025-03-13T12:30:01.190616-05:00 brayan-Laptop CRON[33512]: (CRON) info (No MTA installed, discarding output)
2025-03-13T12:30:01.230860-05:00 brayan-Laptop CRON[33510]: (CRON) info (No MTA installed, discarding output)
```