



SERVICIOS MONTAR/DESMONTAR UDS (LOCAL-REMOTE)



OTROS DOCUMENTOS DE LINUX

[SERVICIOS-SYSTEMD/SHUTDOWN](#)

[BASH_COMANDS](#)

[BASH COMANDS - Extended](#)



INDICE DOCUMENTO



CONCEPTOS UTILES

[Montaje Google-Drive](#)

[Montaje One-Drive](#)

[Montaje Datos-D](#)

[GPT](#)



CREAR UN SERVICIOS SYSTEMD



MONTAJE UNIDAD REMOTA GOOGLE DRIVE



CREAR SCRIPT DE MONTAJE

Ruta: `/usr/local/bin/file-name.sh`

```
#!/bin/bash
#/usr/local/bin/mount_drive.sh

rclone mount --allow-non-empty gdrive: /home/gerar_kde/Drive &
exit
```

& exit

& hace que el proceso se ejecute en segundo plano, y luego haces exit, lo que mata el proceso del script junto con el montaje, porque Type=simple espera que el proceso continúe.

--allow-non-empty

no es recomendable, puede causar errores si la carpeta ya tiene contenido.

--vfs-cache-mode

es obligatorio para muchas operaciones con rclone mount.





CORREGIDO:

```
#!/bin/bash
#/usr/local/bin/mount_drive.sh

/usr/bin/rclone mount \
gdrive: /home/gerar_kde/Drive \
--vfs-cache-mode writes \
--allow-other
```



VERSION MEJORADA DEL SCRIPT

- Verificación del punto de montaje
- Registro en log
- Control de errores
- Ejecución en segundo plano si lo usas como servicio systemd

```
#!/bin/bash
# /usr/local/bin/mount_drive.sh

MOUNT_POINT="/home/gerar_kde/Drive"
LOG_FILE="/var/log/mount_drive.log"
TIMESTAMP="$(date '+%Y-%m-%d %H:%M:%S')"
```

Redirige salida estándar y de errores al log (sobrescribe)

```
exec > "$LOG_FILE" 2>&1
echo "[${TIMESTAMP}] Inicio del montaje de Google Drive"
```

Verifica si ya está montado

```
if mountpoint -q "$MOUNT_POINT"; then
    echo "Google Drive ya está montado en $MOUNT_POINT"
    exit 0
fi
```

Verifica que el punto de montaje exista

```
if [ ! -d "$MOUNT_POINT" ]; then
    echo "Error: el directorio $MOUNT_POINT no existe."
    exit 1
fi
```

Ejecuta rclone mount

```
/usr/bin/rclone mount \
gdrive: "$MOUNT_POINT" \
--vfs-cache-mode writes \
--allow-other \
--daemon
```

```
STATUS=$?
```

```
if [ $STATUS -eq 0 ]; then
```



```
    echo "Montaje exitoso de Google Drive en $MOUNT_POINT"
else
    echo "Error al montar Google Drive (código $STATUS)"
    exit $STATUS
fi
```



ASEGURATE DE:

- Crear el archivo de log y dar permisos adecuados:

```
sudo touch /var/log/montar_drive.log
sudo chown gerar_kde:gerar_kde /var/log/montar_drive.log
```

- Hacer el script ejecutable:

```
sudo chmod +x /usr/local/bin/mount_drive.sh
```



CREAR ARCHIVO DE SERVICIO SYSTEMD

Ruta: /etc/systemd/system/file-name.service

```
[Unit]
Description=Montar Google Drive al inicio
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=gerar_kde
ExecStart=/usr/local/bin/montar_drive.sh
Restart=on-failure

[Install]
WantedBy=default.target
```



HABILITAR EL SERVICIO AL INICIO:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable montar_drive.service
```



PRUEBA EL SCRIPT SIN REINICIAR:

```
sudo systemctl start montar_drive.service
```



REVISAR LOGS DEL SERVICIO:

```
journalctl -u montar_drive.service --no-pager
```

Ahí verás si falla algo como permisos, rutas, o errores de rclone.

RECARGA Y REINICIA EL SERVICIO

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable montar_drive.service
```



MONTAJE UNIDAD REMOTA MS ONE-DRIVE

CREAR SCRIPT DE MONTAJE

Ruta: /usr/local/bin/montar_onedrive.sh

```
#!/bin/bash
# /usr/local/bin/montar_onedrive.sh

/usr/bin/rclone mount \
onedrive: /home/gerar_kde/One_Drive \
--allow-non-empty
```

--allow-non-empty

no es recomendable, puede causar errores si la carpeta ya tiene contenido, pero One Drive parece que lo use así

DARLE PERMISOS DE EJECUCION:

```
sudo chmod +x /usr/local/bin/montar_onedrive.sh
```

CREAR ARCHIVO DE SERVICIO SYSTEMD

Ruta: /etc/systemd/system/montar_onedrive.service

```
[Unit]
Description=Montar MS One Drive al inicio
After=network-online.target
Wants=network-online.target

[Service]
```





```
Type=simple
User=gerar_kde
ExecStart=/usr/local/bin/montar_onedrive.sh
Restart=on-failure

[Install]
WantedBy=default.target
```

HABILITAR EL SERVICIO AL INICIO:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable montar_onedrive.service
```

PRUEBA EL SCRIPT SIN REINICIAR:

```
sudo systemctl start montar_onedrive.service
```

REVISAR LOGS DEL SERVICIO:

```
journalctl -u montar_onedrive.service --no-pager
```

Ahí verás si falla algo como permisos, rutas, o errores de rclone.

RECARGA Y REINICIA EL SERVICIO

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable montar_onedrive.service
```



MONTAJE UNIDAD DE PARTICIÓN /DATOS

CREAR DIRECTORIO DE MONTAJE

```
sudo mkdir -p /media/DATOS
sudo chown gerar_kde:gerar_kde /media/DATOS
```

chown

controlador moderno para NTFS (más rápido que ntfs-3g en kernels nuevos).





gerar_kde:gerar_kde: asigna como propietario al usuario gerar_kde y también al grupo gerar_kde

CREAR SCRIPT DE MONTAJE DE UNIDAD



SCRIPT MEJORADO PARA:

- Evitar montar si ya está montada
- Verificar permisos de escritura del punto de montaje
- Informar claramente en caso de error

Ruta: `/usr/local/bin/montar_datos.sh`

```
#!/bin/bash
#/usr/local/bin/montar_datos.sh

MOUNT_POINT="/media/DATOS"
UUID="524F567904D39ECB"
FS_TYPE="ntfs3"

# Comprobamos si ya está montado
if mountpoint -q "$MOUNT_POINT"; then
    echo "La partición ya está montada en $MOUNT_POINT."
    exit 0
fi

# Comprobamos si el punto de montaje existe y tiene permisos
if [ ! -d "$MOUNT_POINT" ]; then
    echo "Error: El directorio $MOUNT_POINT no existe."
    exit 1
fi

if [ ! -w "$MOUNT_POINT" ]; then
    echo "Error: No tienes permisos de escritura en $MOUNT_POINT."
    exit 2
fi

# Intentamos montar
mount -t $FS_TYPE -o rw,uid=1000,gid=1000,umask=022 UUID=$UUID
"$MOUNT_POINT"
STATUS=$?

if [ $STATUS -eq 0 ]; then
    echo "Partición montada correctamente en $MOUNT_POINT."
else
    echo "Error al montar la partición (código $STATUS)."
    exit $STATUS
fi
```

ntfs3 controlador moderno para NTFS (más rápido que ntfs-3g en kernels nuevos).

uid=1000,gid=1000 asegura que el usuario (gerar_kde) tenga acceso. Tu usuario gerar_kde tiene UID 1000 (casi





<code>id -u gerar_kde</code>	siempre es así para el primer usuario). Puedes verificarlo con: <code>id -u gerar_kde</code>
<code>umask=022</code>	da permisos de lectura para todos y escritura sólo para el dueño.

DARLE PERMISOS DE EJECUCION:

```
sudo chmod +x /usr/local/bin/montar_datos.sh
```

CREAR ARCHIVO DE SERVICIO SYSTEMD

Ruta: `/etc/systemd/system/montar_datos.service`

```
[Unit]
Description=Montar partición Datos (NTFS)
After=local-fs.target

[Service]
Type=oneshot
ExecStart=/usr/local/bin/montar_datos.sh
RemainAfterExit=true

[Install]
WantedBy=multi-user.target
```

<code>Type=oneshot</code>	porque solo monta una vez.
<code>RemainAfterExit=true</code>	para que systemd lo considere "activo" después de ejecutar.

HABILITAR EL SERVICIO AL INICIO Y PROBAR :

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable montar_datos.service
sudo systemctl start montar_datos.service
```

PRUEBA EL SCRIPT SIN REINICIAR:

```
sudo systemctl start montar_drive.service
```

VERIFICA SI ESTA MONTADO:

```
mount | grep Datos-D
```





SCRIPT MEJORADO DE MONTAJE DE UNIDAD

- incluye registro en un archivo de log: /var/log/montar_datos.log.

```
#!/bin/bash
#/usr/local/bin/montar_datos.sh

MOUNT_POINT="/media/DATOS"
UUID="524F567904D39ECB"
FS_TYPE="ntfs3"
LOG_FILE="/var/log/montar_datos.log"
TIMESTAMP="$(date '+%Y-%m-%d %H:%M:%S')"
```

```
log() {
    echo "[${TIMESTAMP}] $1" >> "$LOG_FILE"
}

# Comprobamos si ya está montado
if mountpoint -q "$MOUNT_POINT"; then
    log "La partición ya está montada en $MOUNT_POINT."
    exit 0
fi

# Comprobamos si el punto de montaje existe y tiene permisos
if [ ! -d "$MOUNT_POINT" ]; then
    log "Error: El directorio $MOUNT_POINT no existe."
    exit 1
fi

if [ ! -w "$MOUNT_POINT" ]; then
    log "Error: No tienes permisos de escritura en $MOUNT_POINT."
    exit 2
fi

# Intentamos montar
mount -t $FS_TYPE -o rw,uid=1000,gid=1000,umask=022 UUID=$UUID "$MOUNT_POINT"
STATUS=$?

if [ $STATUS -eq 0 ]; then
    log "Partición montada correctamente en $MOUNT_POINT."
else
    log "Error al montar la partición (código $STATUS)."
```

```
    exit $STATUS
fi
```

Asegúrate de que El archivo de log se pueda escribir:

```
sudo touch /var/log/montar_datos.log
sudo chown gerar_kde:gerar_kde /var/log/montar_datos.log
```

El script es ejecutable:



```
sudo chmod +x /usr/local/bin/montar_datos.sh
```



ERROR

```
cat /var/log/montar_onedrive.log
[2025-05-11 22:00:17] Iniciando montaje de OneDrive...
2025/05/11 22:00:19 ERROR : Daemon timed out. Failed to terminate daemon pid
3239: os: process already finished
2025/05/11 22:00:19 CRITICAL: Fatal error: daemon exited with error code 1
Error al montar OneDrive (código 1).
```



ERROR AL INTENTAR MONTAR GOOGLE DRIVE CON RCLONE

El error que estás viendo al intentar montar Google Drive con Rclone indica que el daemon (el proceso en segundo plano) no pudo iniciarse correctamente o se cerró inesperadamente. Aquí está el análisis del mensaje de error:

ERROR : Daemon timed out. Failed to terminate daemon pid 5554: os: process already finished
CRITICAL: Fatal error: daemon exited with error code 1

Esto sugiere que:

- El proceso del daemon de Rclone (probablemente rclone mount) se inició pero terminó inesperadamente.
- Rclone intentó terminar el proceso (quizás porque detectó un error), pero ya había terminado por su cuenta.
- El código de error 1 generalmente indica un fallo genérico en Rclone.

Posibles causas:

- Fallo en la autenticación con Google Drive
- Si el token expiró o hay un error en la configuración remota (rclone config), el montaje fallará.
- Permisos insuficientes
- Si estás ejecutando el script con un usuario que no tiene los permisos necesarios para montar FUSE o crear directorios en el destino.
- FUSE no está correctamente instalado
- Rclone usa FUSE para montar sistemas de archivos. Si fuse o fusemount no están disponibles o bien configurados, puede fallar.
- **Conflicto con un montaje previo**
- **Puede que el punto de montaje ya esté ocupado o haya quedado "colgado" de un intento anterior.**





MONTAR UNIDADES CON SYSTEMD

Montar las unidades directamente sólo con un script servicio, no con script de montaje + script de servicio



REQUISITOS PREVIOS

ASEGURATE DE TENER:

- Rclone funcionando y configurado (con remotos como gdrive: y onedrive:).
- Directorios de montaje vacíos y existentes (ej: ~/mnt/drive, ~/mnt/onedrive).
- FUSE instalado.



CREAR LOS SERVICIOS SYSTEMD



Google Drive

Ruta: /etc/systemd/system/rclone-gdrive.service

```
[Unit]
Description=Montar Google Drive con Rclone
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=gerar_kde
ExecStart=/usr/bin/rclone mount gdrive: /home/gerar_kde/Drive \
--vfs-cache-mode writes \
--dir-cache-time 1h \
--poll-interval 1m \
--log-level INFO \
--log-file /var/log/rclone-gdrive.log
ExecStop=/bin/fusermount -u /home/gerar_kde/Drive
Restart=on-failure

[Install]
WantedBy=default.target
```



MS ONE DRIVE

Ruta: /etc/systemd/system/rclone-onedrive.service

```
[Unit]
Description=Montar OneDrive con Rclone
After=network-online.target
Wants=network-online.target

[Service]
```



```
Type=simple
User=gerar_kde
ExecStart=/usr/bin/rclone mount onedrive: /home/gerar_kde/One_Drive \
  --vfs-cache-mode writes \
  --dir-cache-time 1h \
  --poll-interval 1m \
  --log-level INFO \
  --log-file /var/log/rclone-onedrive.log
ExecStop=/bin/fusermount -u /home/gerar_kde/One_Drive
Restart=on-failure

[Install]
WantedBy=default.target
```



RECARGAR SYSTEMD Y HABILITAR LOS SERVICIOS

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload

sudo systemctl enable rclone-gdrive.service
sudo systemctl enable rclone-onedrive.service
```



OPCIONAL: PARA MONTARLOS AHORA SIN REINICIAR:

```
sudo systemctl start rclone-gdrive.service
sudo systemctl start rclone-onedrive.service
```



LOGS Y MONITOREO

Puedes ver si funcionó con:

```
systemctl status rclone-gdrive.service
cat /var/log/rclone-gdrive.log
```



DESMONTAR DRIVE Y ONE DRIVE AL APAGAR/REINICIAR



CREAR SCRIPT DE DESMONTAJE DE UNIDAD

Script completo que desmonta las unidades de Google Drive y OneDrive solo si están montadas, y que se ejecuta automáticamente al apagar o reiniciar el sistema:

Ruta `/lib/systemd/system-shutdown/desmontar_drives.sh`

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



```
# Rutas a tus puntos de montaje
DRIVE_MOUNT="/home/gerar_kde/Drive"
ONEDRIVE_MOUNT="/home/gerar_kde/One_Drive"

echo "[Shutdown] Intentando desmontar Google Drive y OneDrive..." >>
/var/log/desmontar_drives.log

# Función para desmontar si está montado
desmontar_si_montado() {
    MOUNT_PATH="$1"
    if mount | grep -q "$MOUNT_PATH"; then
        fusermount -u "$MOUNT_PATH" && echo "Desmontado: $MOUNT_PATH"
    >> /var/log/desmontar_drives.log || echo "Fallo al desmontar:
    $MOUNT_PATH" >> /var/log/desmontar_drives.log
    else
        echo "No montado: $MOUNT_PATH" >> /var/log/desmontar_drives.log
    fi
}

desmontar_si_montado "$DRIVE_MOUNT"
desmontar_si_montado "$ONEDRIVE_MOUNT"

exit 0
```



DAR PERMISOS DE EJECUCION

```
sudo chmod +x /lib/systemd/system-shutdown/desmontar_drives.sh
```



PRUEBA EL SCRIPT SIN APAGAR:

```
sudo /lib/systemd/system-shutdown/desmontar_drives.sh
```



REVISAR EL LOG:

```
cat /var/log/desmontar_drives.log
```

En este caso, no es necesario que el script tenga la extensión .sh. En Linux, los scripts que se ejecutan automáticamente por el sistema (como los de /lib/systemd/system-shutdown/) son identificados por su nombre y permisos, no por la extensión.

Lo importante es que:

- El archivo sea ejecutable (chmod +x).
- Tenga una shebang válida (#!/bin/bash al principio).





- Esté en el directorio correcto (como `/lib/systemd/system-shutdown/`).

Aunque puede ser [desmontar_drives.sh](#) , Y También funcionará perfectamente

