

Guía paso a paso para implementar y ejecutar el script en Ubuntu

1. Crear el script

1. Accede a tu terminal en Ubuntu.

Crea un archivo de script en el directorio donde prefieras guardar el script. Por ejemplo, en el directorio **backups**:

```
mkdir ~/backups  
cd ~/backups  
touch script_backup.sh
```

Edita el archivo con un editor de texto como **nano**:

```
nano script_backup.sh
```

Copia y pega el contenido del script que proporcionaste:

```
#!/bin/bash  
  
# Configuración  
BACKUP_DIR="/home/andres/backups"  
SOURCE_DIR="/home/andres/Documentos/UNAB"  
LOG_FILE="/var/log/script_backup.log"  
DAYS_TO_KEEP=7  
  
# Función de log  
log_message() {  
    echo "$(date +%Y-%m-%d %H:%M:%S) - $1" | tee -a "$LOG_FILE"  
}  
  
# Crear backup  
log_message "Iniciando copia de seguridad..."  
tar -czf "$BACKUP_DIR/backup_$(date +%Y%m%d').tar.gz" "$SOURCE_DIR"  
log_message "Copia de seguridad completada."  
  
# Limpiar logs antiguos (más de 7 días)  
log_message "Eliminando logs antiguos..."  
find /var/log -name "*.log" -type f -mtime +$DAYS_TO_KEEP -exec rm -f {} \;  
log_message "Limpieza de logs finalizada."  
  
# Actualizar paquetes del sistema  
log_message "Iniciando actualización de paquetes..."  
sudo apt-get update -y  
# si se quiere instalar automáticamente los paquetes podría utilizar  
# sudo apt-get upgrade -y  
log_message "Actualización de paquetes completada."
```

```
log_message "Script finalizado."
```

Guarda y cierra el editor presionando **CTRL + X**, luego **Y** y **Enter**.

2. Dar permisos de ejecución al script

Otorga permisos de ejecución para que el script pueda ser ejecutado:

```
chmod +x script_backup.sh
```

Verifica los permisos con el siguiente comando:

```
ls -l script_backup.sh
```

Verás algo similar a:

```
-rwxr-xr-x 1 andres andres 1234 mar 26 14:00 script_backup.sh
```

1. El **x** indica que el archivo es ejecutable.

3. Probar el script manualmente

Ejecuta el script manualmente para asegurarte de que todo funcione correctamente:

```
./script_backup.sh
```

1. Verifica la salida:

- Se creará un archivo **.tar.gz** en el directorio de backups.
- Se limpiarán los logs antiguos (más de 7 días).
- Se actualizarán los paquetes del sistema.
- El log del script será enviado al correo especificado.

4. Configurar el script para que se ejecute automáticamente (cron)

Abre el crontab para editar las tareas programadas:

```
crontab -e
```

Añade una línea al final del archivo para que el script se ejecute automáticamente cada semana (por ejemplo, los domingos a las 2 a.m.):

```
0 2 * * 0 /home/andres/backups/script_backup.sh
```

1. Esta línea indica que el script se ejecutará a las 2:00 AM todos los domingos.

2. **Guarda y cierra el crontab.** En **nano**, presiona **CTRL + X**, luego **Y** y **Enter**.

Verifica que el cronjob está configurado correctamente con el siguiente comando:

```
crontab -l
```

Deberías ver algo como:

```
0 2 * * 0 /home/andres/backups/script_backup.sh
```

5. Verificar el funcionamiento del cronjob

Verifica que el cronjob se ejecutó correctamente revisando el archivo de log:

```
cat /var/log/script_backup.log
```

1. También puedes revisar si el archivo de backup se ha creado en **/home/andres/backups**.

6. Usar sudo en el Script

Agregar sudo a comandos que necesiten permisos elevados:

```
sudo find /var/log -name "*.log" -type f -mtime +7 -exec rm -f {} \;
```

1. **Alternativa (dar permisos a los archivos de log):**

```
sudo chmod -R u+rw /var/log
```

7. Restaurar Permisos de Seguridad

Restaurar permisos de directorio /var/log:

```
sudo chmod 755 /var/log
```

Restaurar permisos de archivos .log:

```
sudo chmod 644 /var/log/*.log
```