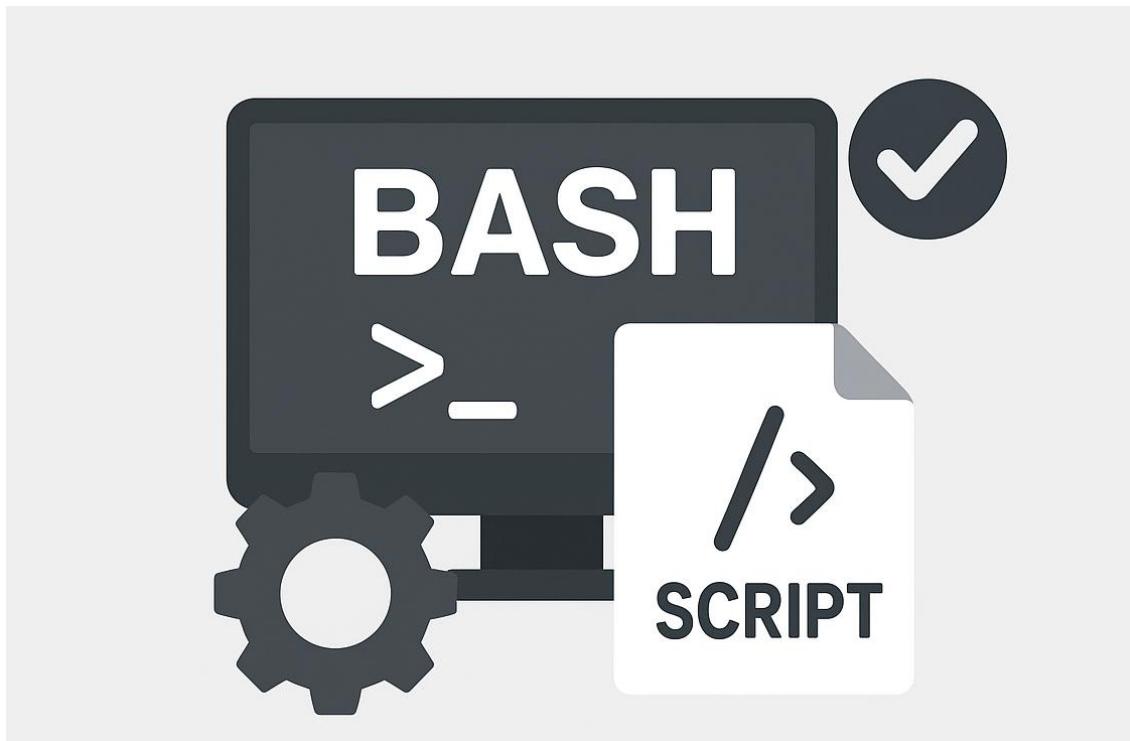


# **TRABAJO FINAL INTEGRADOR MEMORIA TECNICA**

**Automatización de tareas del Sistema Operativo**



Materia: **Sistemas Operativos – Tec. Desarrollo de Software**

Alumnos: **Larsen Camila, Nerli Matias, Rosa Matias, Natiello Vicente**

Año: **2025**

# 1. Introducción

Este trabajo tiene como objetivo desarrollar un sistema de automatización de tareas utilizando **scripts en Bash** dentro del entorno Linux. A través de este proyecto se busca aplicar los conceptos vistos en clase sobre líneas de comandos, permisos, procesos, y gestión del sistema operativo, automatizando acciones que comúnmente se realizan de forma manual.

El proyecto se compone de un conjunto de tres scripts principales, ejecutados desde un menú interactivo que permiten realizar tareas de mantenimiento, monitoreo y respaldo del sistema. Esta automatización no solo mejora la eficiencia del trabajo del usuario, sino que también contribuye al aprendizaje práctico del funcionamiento interno del sistema operativo.

## 2. Objetivos

- Automatizar tareas básicas del sistema Linux utilizando scripts en Bash.
- Aplicar estructuras de control, comandos y permisos del sistema.
- Mejorar la comprensión del funcionamiento de los procesos del sistema operativo.
- Implementar un menú que centralice todas las funciones del proyecto.

## 3. Desarrollo del proyecto

El trabajo consta de cuatro archivos principales:

- *menu.sh*
- *backup.sh*
- *informe.sh*
- *limpiar.sh*

A continuación, se describe la función de cada uno:

## menu.sh – Menú principal

Este script funciona como **interfaz central del programa**, permitiendo al usuario seleccionar qué tarea desea ejecutar. Mediante una estructura **case**, el menú ofrece las siguientes opciones:



```
GNU nano 8.3                               menu.sh
#!/bin/bash
while true; do
    clear
    echo -e "\e[1;34m _____ \e[0m"
    echo -e "\e[1;34m     MENÚ PRINCIPAL      \e[0m"
    echo -e "\e[1;34m _____ \e[0m"
    echo -e " 1. Realizar Backup"
    echo -e " 2. Generar informe del sistema"
    echo -e " 3. Limpiar archivos temporales"
    echo -e " 4. Salir"
    echo -e "\e[1;34m _____ \e[0m"
    read -p " Seleccione una opcion: " opcion

    case $opcion in
        1) bash backup.sh
            echo ""
            echo -e "\e[1;32m Backup completado correctamente.\e[0m"
            echo ...
            read -p "Presione Enter para volver al menu..."
            ;;
        2) bash informe.sh
            echo ""
            echo -e "\e[1;32m Informe generado exitosamente.\e[0m"
            echo ...
            read -p "Presione Enter para volver al menu..."
            ;;
        3) bash Limpiar.sh
            echo ""
            echo -e "\e[1;32m Limpieza completada con exito.\e[0m"
            read -p "Presione Enter para volver al menu..."
            ;;
        4) echo -e "\e[1;33m Saliendo...\e[0m"; sleep 2; break;
        *) echo -e "\e[1;31m Opcion no valida. Intente nuevamente.\e[0m"; sleep 2;;
    esac
done
```

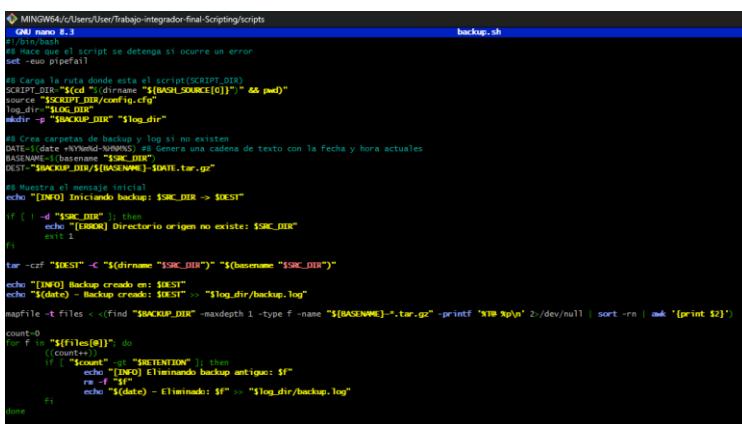
El **menú** se ejecuta dentro de un bucle **while**, permitiendo al usuario realizar varias acciones consecutivas sin reiniciar el programa.

## backup.sh – Copia de seguridad automática

Este script crea un **backup comprimido** de la carpeta “Documentos” del usuario.

El archivo generado se almacena en una carpeta “backups” dentro del directorio personal (\$HOME/backups), con un nombre que incluye la fecha y hora del respaldo.

Además, se incluye un comando **find** que elimina automáticamente los respaldos con más de 7 días de antigüedad, optimizando el espacio en disco.



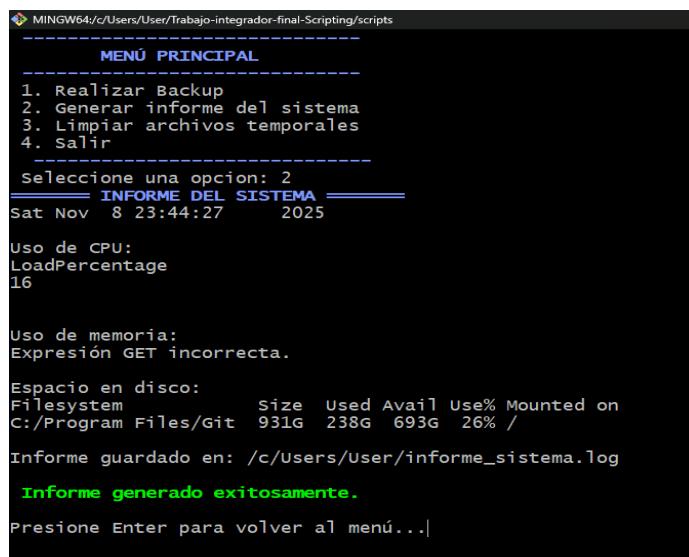
```
GNU nano 8.3
#!/bin/bash
# El script se detenga si ocurre un error
set -euo pipefail
## Carga la ruta donde esta el script(SCRIPT_DIR)
SCRIPT_DIR=$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)
source "$SCRIPT_DIR/conf/q.cfg"
log_dir="$HOME/log_dir" "log_dir"
## Crea carpetas de backup y log si no existen
DATE=$(date +%Y%m%d-%H%M%S) # Genera una cadena de texto con la Fecha y hora actuales
BASENAME=$(basename "$SRC_DIR")
DEST="$BACKUP_DIR/$BASENAME-$DATE.tar.gz"
## Muestra el mensaje inicial
echo "[INFO] Iniciando backup: $SRC_DIR -> $DEST"
if [ ! -d "$SRC_DIR" ]; then
    echo "[ERROR] Directorio origen no existe: $SRC_DIR"
    exit 1
fi
tar -czf "$DEST" < ${!((dirname "$SRC_DIR"))} ${!((basename "$SRC_DIR"))}
echo "[INFO] Backup creado en: $DEST"
echo "$((date)) - Backup creado: $DEST" >> "$log_dir/backup.log"
rmfile=tar -t files < find "$BACKUP_DIR" -maxdepth 1 -type f -name "${BASENAME}-*.tar.gz" | sort -rn | awk '{print $2}'
count=0
for f in ${!rmfile[@]}; do
    if [ ${rmfile[$count]} = "INCREMENT" ]; then
        echo "[INFO] Eliminando backup antiguo: $f"
        rm -rf "$f"
        echo "$((date)) - Eliminado: $f" >> "$log_dir/backup.log"
    fi
done
```

## informe.sh – Informe del sistema

Este script recopila información actual del sistema, incluyendo:

- Porcentaje de uso del CPU.
- Memoria total y disponible.
- Espacio libre y ocupado en disco.

Toda la información se guarda en un archivo de texto llamado **informe\_sistema.log**, con la fecha y hora de generación. El propósito es que el usuario pueda monitorear el rendimiento y estado general del sistema de manera rápida y ordenada.



```
MENÚ PRINCIPAL
1. Realizar Backup
2. Generar informe del sistema
3. Limpiar archivos temporales
4. Salir
Seleccione una opcion: 2
INFORME DEL SISTEMA
Sat Nov 8 23:44:27 2025

Uso de CPU:
LoadPercentage
16

Uso de memoria:
Expresión GET incorrecta.

Espacio en disco:
Filesystem      Size  Used Avail Use% Mounted on
C:/Program Files/Git  931G  238G  693G  26% /

Informe guardado en: /c/Users/User/informe_sistema.log
Informe generado exitosamente.

Presione Enter para volver al menú...|
```

```

MINGW64/c/Users/User/Trabajo-integrador-final-Scripting/scripts
GNU nano 8.3                                         informe.sh

#!/bin/bash
# informe.sh
# Genera un informe de CPU, memoria y disco (muestra y guarda en log)
log="SHOME/informe_sistema.log"
echo -e "\e[1;34m----- INFORME DEL SISTEMA ----- \e[0m" | tee "$log"
date | tee -a "$log"
echo "" | tee -a "$log"
echo "Uso de CPU:" | tee -a "$log"
if command -v top >/dev/null; then
    top -bn1 | grep "CPU(s)" | tee -a "$log"
elif command -v wmic >/dev/null; then
    wmic cpu get loadpercentage | tee -a "$log"
else
    echo "No se pudo obtener la información de CPU" | tee -a "$log"
fi
echo "" | tee -a "$log"
echo "Uso de memoria:" | tee -a "$log"
if command -v free >/dev/null; then
    free -h | tee -a "$log"
elif command -v wmic >/dev/null; then
    wmic os get FreePhysicalMemory,TotalVisibleMemorySize /value | tee -a "$log"
else
    echo "No se pudo obtener la información de memoria" | tee -a "$log"
fi
echo "" | tee -a "$log"
echo "Espacio en disco:" | tee -a "$log"
df -h | tee -a "$log"
echo "" | tee -a "$log"
echo "Informe guardado en: $log" | tee -a "$log"
sleep 2

```

El script utiliza comandos como **top**, **free**, **df** y **wmic**, combinados con **tee** para visualizar y registrar simultáneamente la salida. También aplica colores en la salida del terminal, incorpora *control de errores* ante la falta de comandos disponibles y fecha/hora de ejecución, dando así, un resultado legible y completo del estado actual del sistema.

## limpiar.sh – Mantenimiento del sistema

El script **automatiza tareas de limpieza del sistema** para liberar espacio en disco y mantener el entorno ordenado.

Todo el proceso queda registrado en un archivo de *log* (*limpieza\_sistema.log*), donde se guarda la fecha, las carpetas limpiadas y los archivos eliminados.

- Define una variable **LOG\_FILE** con la ruta del archivo de registro:
- Elimina archivos temporales del sistema que se encuentran en la carpeta **/tmp**:
- Busca y elimina carpetas de caché en el directorio del usuario (por ejemplo, **~/.cache**):
- Borra archivos con extensiones típicas de temporales en el directorio personal:
- Todos los mensajes y resultados se envían al archivo log mediante **tee -a "\$LOG\_FILE"**, para mantener un registro histórico de cada ejecución.

```

MINGW64/c/Users/User/Trabajo-integrador-final-Scripting/scripts
GNU nano 8.3                                         limpiar.sh

#!/bin/bash
# limpiar.sh
# Limpia archivos temporales del sistema y del usuario en Linux

# CONFIGURACION
LOGFILE="SHOME/limpieza_sistema.log"
DATE=$(date "+%Y-%m-%d %H:%M:%S")
echo -e "\e[1;34m----- LIMPIEZA DE ARCHIVOS TEMPORALES ----- \e[0m" | tee -a "$LOGFILE"
echo "Fecha: $DATE" | tee -a "$LOGFILE"
echo "" | tee -a "$LOGFILE"

# LIMPIEZA DE /tmp Y /var/tmp
echo -e "\e[1;34m----- Limpando carpetas temporales del sistema... \e[0m" | tee -a "$LOGFILE"
rm -rf /tmp/* /var/tmp/* 2>/dev/null
echo "[OK] Carpetas /tmp y /var/tmp limpiadas." | tee -a "$LOGFILE"
echo "" | tee -a "$LOGFILE"

# LIMPIEZA DE CACHE DE USUARIO
echo -e "\e[1;34m----- Limpando cachés del usuario... \e[0m" | tee -a "$LOGFILE"
find "/home/" -name ".cache" -type d | rm -rf --one-file-system 2>/dev/null
echo "[OK] Cachés del usuario eliminados." | tee -a "$LOGFILE"
else
    echo "[INFO] No se encontró carpeta de caché en el usuario actual." | tee -a "$LOGFILE"
fi
echo "" | tee -a "$LOGFILE"

# LIMPIEZA DE ARCHIVOS TEMPORALES GENERALES
echo -e "\e[1;34m----- Eliminando archivos temporales de usuario (*.tmp, *.bak, *.log)... \e[0m" | tee -a "$LOGFILE"
find "SHOME" -type f -name "*.tmp" -o -name "*.bak" -o -name "*.log" -delete 2>/dev/null
echo "[OK] Archivos temporales eliminados del directorio personal." | tee -a "$LOGFILE"
echo "" | tee -a "$LOGFILE"

```

**MENÚ PRINCIPAL**

1. Realizar Backup
2. Generar informe del sistema
3. Limpieza de archivos temporales
4. Salir

Seleccione una opción: 3 **LIMPIEZA DE ARCHIVOS TEMPORALES**

Fecha: 2025-11-08 23:45:20

- Limpieza de archivos temporales del sistema...

[OK] Carpetas /tmp y /var/tmp limpiadas.

- Limpieza de cachés del usuario...

[INFO] No se encontró carpeta de caché en el usuario actual.

- Eliminación de archivos temporales de usuario (\*.tmp, \*.bak, \*.log)...

[OK] Archivos temporales eliminados del directorio personal.

■ Limpieza completa correctamente.

Informe guardado en: /c/Users/User/limpieza\_sistema.log

Limpieza completa con éxito.

## 4. Requisitos técnicos

- Sistema operativo: Linux (Ubuntu, Debian o derivados).
  - Intérprete de comandos Bash.
  - Permisos de ejecución en los scripts (*chmod +x nombre.sh*).
  - Acceso básico de administrador para la limpieza del sistema.
- 

## 5. Pruebas y resultados

Durante la ejecución del menú, el usuario puede visualizar en la terminal las distintas acciones realizadas por los scripts. Se realizaron pruebas individuales de cada módulo, confirmando que:

- Los backups se generan correctamente.
- El informe del sistema se guarda con los datos actualizados.
- Los archivos temporales son eliminados sin afectar el funcionamiento del sistema.

Cada opción ejecuta correctamente la tarea correspondiente y muestra un mensaje de confirmación en pantalla.

---

## 6. Conclusión

El proyecto permitió integrar los conceptos teóricos de Sistemas Operativos con la práctica real del uso de la línea de comandos en **Linux**. La automatización mediante scripts demostró ser una herramienta útil para optimizar tareas repetitivas, reducir errores humanos y mejorar la organización del sistema.

Además, se reforzaron conocimientos sobre permisos, estructuras de control, comandos del sistema, redirección de archivos y gestión de procesos. Como mejora futura, se podría incluir un registro histórico de ejecuciones y una interfaz visual.

---

## 7. Referencias

- Manual de Bash GNU – <https://www.gnu.org/software/bash/>
- Documentación oficial de Ubuntu – <https://help.ubuntu.com/>
- Manuales y apuntes de la materia Sistemas Operativos