

Script de Automatización

Instituto Superior IDRA

Tecnicatura de Desarrollo de Software

Santarsiero Mateo <mateo.santarsiero@gmail.com>

Pedro Joaquín Mendoza <pedrojoaquinmendoza@gmail.com>

Thiago Tarricone <thiagotarricone2003@gmail.com>

1. Descripción del proyecto.....	3
2. Requisitos.....	3
3. Explicación del código.....	4
Menú:.....	4
Respaldo de un directorio:.....	5
Generación de informe (CPU, Memoria, Disco):.....	6
Creación de nuevo usuario:.....	7
Actualización del sistema:.....	8
Eliminación de archivos temporales y caché:.....	9
4. Reflexión.....	9

1. Descripción del proyecto

El proyecto consistirá de un programa realizado en bash en un sistema operativo Linux basado en Debian, *Raspberry PI OS*. Donde mediante un menú interactivo se podrán realizar las siguientes acciones en el sistema:

- Respaldo de un directorio a un formato comprimido zip.
- Generación de un informe legible del uso del CPU, memoria y disco.
- Creación de un nuevo usuario.
- Actualización del sistema y los paquetes instalados en el sistema.
- Eliminación de archivos caché e innecesarios

2. Requisitos

- Poseer un entorno en el cual se pueda ejecutar un script en bash.
- Poseer el paquete zip instalado. (sudo apt install zip unzip).
- Permisos de ejecución del script (sudo chmod +x script.sh).

3.Explicación del código

Colores:

Para la función de los colores se introduce un código haciendo referencia al color que se necesite.

ENDCOLOR: Esto realiza que sea incoloro todo lo que siga después de él .

```
RED='\e[1;31m'  
GREEN='\e[1;32m'  
CYAN='\e[1;36m'  
ENDCOLOR='\e[0m'
```

Imprime las opciones con colores para luego, mediante un ciclo *while true*, imprima en pantalla las opciones para ingresar a las distintas funciones del script:

[illegible]

Respaldo de un directorio:

Se imprimen en pantalla los archivos y directorios donde se está ejecutando el script. Al momento de ingresar el nombre del directorio (**sin barra invertida '/'**), se comprueba si la ruta ingresada es un directorio, en tal caso de serlo se comprueba la existencia de una carpeta de respaldos "respaldos/", de no existir automáticamente se genera una. Se elimina un respaldo antiguo del mismo directorio en caso de que exista alguno, se comprime el directorio seleccionado y se mueve a la carpeta "respaldos/".

```
opcion1(){
    echo $(ls)
    read -p "Que directorio desea respaldar? " direct
    ruta=$PWD/$direct
    if [ -d "$ruta" ]; then
        if [ ! -d "respaldos/" ];then
            mkdir respaldos
        fi

        if [ -e "respaldos/$direct.zip" ];then
            rm respaldos/$direct.zip
            echo -e "${RED}Respaldo antiguo encontrado y eliminado!${ENDCOLOR}"
        fi

        zip -r $direct.zip $direct/
        mv $direct.zip respaldos/
        echo -e "Respaldo creado en ${GREEN}'respaldos'${ENDCOLOR} con el nombre de ${CYAN}$direct.zip${ENDCOLOR}"

    elif [ -e "$ruta" ]; then
        echo -e "${RED}$ruta es un archivo.${ENDCOLOR}"
    else
        echo -e "${RED}No existe $direct${ENDCOLOR}"
    fi
}
```

```
user@raspberrypi:~/Desktop/PruebaLinux/PruebaLinux$ ls
DirectorioPrueba README.md respaldos script.sh
user@raspberrypi:~/Desktop/PruebaLinux/PruebaLinux$ ls respaldos/
DirectorioPrueba.zip
user@raspberrypi:~/Desktop/PruebaLinux/PruebaLinux$
```

Generación de informe (CPU, Memoria, Disco):

Se genera un archivo “reporte.log” con los siguientes datos y mediante los comandos:

Fecha y Hora	date '+%F %T'
Uso del CPU	top -bn1 grep “
Uso de Memoria	free -h
Uso de Disco	df -h

```
✓ opcion2(){
    {
        echo " --- INFORME --- "
        echo -e "Fecha y Hora actual:$(date '+%F %T')"        echo ""
        echo "Uso de CPU"
        echo ""
        top -bn1 | grep "Cpu(s)"
        echo ""
        echo "Uso de Memoria:"
        free -h
        echo ""

        echo "Uso de Disco:"
        df -h
    } > reporte.log
    echo -e "Reporte creado en ${GREEN}'reporte.log'${ENDCOLOR}"
}
```

```
user@raspberrypi:~/Desktop/PruebaLinux/PruebaLinux$ cat reporte.log
--- INFORME ---
Fecha y Hora actual: 2024-10-27 15:43:07

Uso de CPU:
%Cpu(s):  7,1 us,   0,0 sy,   0,0 ni, 92,9 id,   0,0 wa,   0,0 hi,   0,0 si,   0,0 st

Uso de Memoria:
              total        used        free      shared  buff/cache   available
Mem:          1,9Gi         276Mi         539Mi          27Mi          1,1Gi          1,5Gi
Swap:          1,0Gi           0,0Ki          1,0Gi

Uso de Disco:
S.ficheros  Tamaño Usados  Disp Uso% Montado en
udev         947M      0  947M   0% /dev
tmpfs        199M    948K  198M   1% /run
/dev/sda1     19G    7,0G   11G  40% /
tmpfs        992M      0  992M   0% /dev/shm
tmpfs         5,0M    4,0K   5,0M   1% /run/lock
tmpfs        199M    28K  199M   1% /run/user/1000
```

Creación de nuevo usuario:

Se le pide al usuario que ingrese el nombre del nuevo usuario, si el usuario ya está creado, mostrará un cartel de advertencia. En caso de que no exista un usuario con el mismo nombre, se añade el usuario (*sudo adduser*). Luego se le pregunta al usuario si quiere añadir el usuario dentro de un grupo.

Si la respuesta es afirmativa, el código pide que se ingrese el nombre del grupo separado por comas, verifica si el nombre de ese grupo ya está creado (*grupos array <<< "\$grupos"*), si no lo está, lo crea (*sudo groupadd*), y si es grupo ya se había creado previamente, lo añade a ese grupo (*sudo usermod -aG*).

Si la respuesta es negativa, simplemente crea un usuario sin grupo.

```
opcion3(){
    echo -e "Ingrese el nombre del ${GREEN}nuevo usuario${ENDCOLOR}:"
    read nombre_usuario

    if id "$nombre_usuario" &>/dev/null; then
        echo -e "${RED}El usuario '$nombre_usuario' ya existe.${ENDCOLOR}"
        return 1
    fi

    sudo adduser "$nombre_usuario"

    if id "$nombre_usuario" &>/dev/null; then
        echo -e "${GREEN}El usuario '$nombre_usuario' se creó exitosamente.${ENDCOLOR}"

        echo -e "¿Desea añadir grupos al usuario? (${CYAN}s/n${ENDCOLOR}):"
        read agregar_grupos

        if [[ "$agregar_grupos" == "s" || "$agregar_grupos" == "S" ]]; then
            echo -e "Ingrese los nombres de los grupos separados por comas:"
            read grupos

            IFS=',' read -r -a grupos_array <<< "$grupos"

            for grupo in "${grupos_array[@]"; do
                if ! getent group "$grupo" &>/dev/null; then
                    echo -e "${CYAN}El grupo '$grupo' no existe. Creándolo...${ENDCOLOR}"
                    sudo groupadd "$grupo"
                fi

                sudo usermod -aG "$grupo" "$nombre_usuario"
                echo -e "${GREEN}Añadido al grupo '$grupo'.${ENDCOLOR}"
            done
        fi
    else
        echo -e "${RED}Hubo un error al crear el usuario '$nombre_usuario'.${ENDCOLOR}"
    fi
}
```



```

Ingrese el nombre del nuevo usuario:
gaspaii
Añadiendo el usuario `gaspaii' ...
Añadiendo el nuevo grupo `gaspaii' (1007) ...
Añadiendo el nuevo usuario `gaspaii' (1005) con grupo `gaspaii' ...
Creando el directorio personal `/home/gaspaii' ...
Copiando los ficheros desde `/etc/skel' ...
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para gaspaii
Introduzca el nuevo valor, o pulse INTRO para usar el valor predeterminado
    Nombre completo []: Gaspi
    Número de habitación []:
    Teléfono del trabajo []:
    Teléfono de casa []:
    Otro []:
¿Es correcta la información? [S/n] S
El usuario 'gaspaii' se creó exitosamente.
¿Desea añadir grupos al usuario? (s/n):
s
Ingrese los nombres de los grupos separados por comas:
youtuber,grupoSinNombre
Añadido al grupo 'youtuber'.
El grupo 'grupoSinNombre' no existe. Creándolo...
Añadido al grupo 'grupoSinNombre'.

```

Actualización del sistema:

Se actualiza la base de datos de los repositorios mediante *apt update*, posteriormente se descargan e instalan todas las actualizaciones de los paquetes instalados mediante *apt upgrade -y*.

```

opcion4(){
echo -e "${CYAN}Actualizando el Sistema${ENDCOLOR}..."
sudo apt update && sudo apt upgrade -y
echo -e "${CYAN}Sistema actualizado correstamente.${ENDCOLOR}"
}

```

```

Actualizando el Sistema...
Obj:1 http://deb.debian.org/debian bullseye InRelease
Obj:2 http://deb.debian.org/debian-security bullseye-security InRelease
Err:3 https://cli.github.com/packages stable InRelease
  Certificate verification failed: The certificate is NOT trusted. The certificate issuer is unknown. Could not handshake: Error in the certificate verification. [IP: 185.199.110.153 443]
Obj:4 http://deb.debian.org/debian bullseye-updates InRelease
Obj:5 http://archive.raspberrypi.org/debian bullseye InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se puede actualizar 1 paquete. Ejecute «apt list --upgradable» para verlo.
W: https://cli.github.com/packages/dists/stable/InRelease: No system certificates available. Try installing ca-certificates.
W: Fallo al obtener https://cli.github.com/packages/dists/stable/InRelease Certificate verification failed: The certificate is NOT trusted. The certificate issuer is unknown. Could not handshake: Error in the certificate verification. [IP: 185.199.110.153 443]
W: No se han podido descargar algunos archivos de índice, se han omitido, o se han utilizado unos antiguos en su lugar
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los siguientes paquetes se han retenido:
  linux-image-amd64:amd64
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Sistema actualizado correstamente.

```

Eliminación de archivos temporales y caché:

Mediante *apt autoremove -y* se eliminan las dependencias de paquetes que ya han sido eliminados. De igual manera, mediante *apt clean* y *rm -rf /tmp/**, se eliminan los archivos de paquete caché (**.deb**) y los archivos y directorios encontrados en el directorio **tmp**.

```
❏ opcion5(){  
  echo -e "${CYAN}Eliminando archivos innecesarios y caché${ENDCOLOR}..."  
  sudo apt autoremove -y  
  sudo apt clean  
  sudo rm -rf /tmp/*  
  echo -e "${CYAN}Archivos y caché eliminados correctamente.${ENDCOLOR}"  
}
```

```
Eliminando archivos innecesarios y caché...  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 1 no actualizados.  
Archivos y caché eliminados correctamente.
```

4. Reflexión

Mientras estábamos realizando este proyecto, nos encontramos con varias dificultades, las más destacables son las relacionadas con las máquinas virtuales, sin embargo las pudimos solucionar en un tiempo prolongado. Además, no poseíamos práctica con el lenguaje Bash, por lo que nos costó acostumbrarnos a la sintaxis. Pero al final se logró terminar el proyecto satisfactoriamente.