

# Trabajo final

## Sistemas Operativos



Integrantes: Yamila Sanchez,  
Valentina olmos, Giuliana Manzo

Profesor: Lucas Treser

**Desarrollo en software**

**Primer año**

**Comisión: B**

**Instituto Superior IDRA**

# Contenido

<b>Contenidos.....</b>	<b>2</b>
<b>Introduccion.....</b>	<b>3</b>
1.1. Descripción del proyecto.....	3
1.2. Objetivos generales del proyecto.....	4
1.3. Objetivos específicos de cada script.....	4
<b>Ejecución del Script en Linux.....</b>	<b>5</b>
2.1. Requisitos para ejecutar los scripts .....	5
2.2. Pasos para ejecutar los scripts en Linux.....	6
2.3. Guardar los scripts.....	6
2.4. Abrir la terminal(Shell).....	6
2.5. Accede a la Ruta del Script.....	7
2.6. Verificar que los Scripts estén Instalados correctamente.....	8
2.7. Asignar Permisos a los Scripts.....	8
2.8. Ejecución del script.....	9
<b>Pruebas de las funcionalidades del script.....</b>	<b>10</b>
3.1. Generar un informe y guardarlo como archivo log.....	10
3.2. Crear usuarios nuevos y gestionar permisos.....	11
3.3. Eliminar archivos para liberar espacio en el disco duro.....	12
<b>Desarrollo y explicación detallada del código.....</b>	<b>13</b>
4.1. Script menu.....	13
4.2. Script informe.....	15
4.3. Script usuarios.....	17
4.4. Script limpieza.....	20
<b>Reflexiones finales y bibliografía.....</b>	<b>22</b>
5.1. Dificultades y mejoras.....	22
5.2. Referencias bibliográficas.....	23

## Introducción

Este proyecto consiste en el desarrollo de un conjunto de scripts en Bash diseñados para facilitar la administración y mantenimiento básico de un sistema operativo basado en Linux. A menudo, los administradores de sistemas y usuarios avanzados necesitan realizar tareas complejas como monitorear el estado del sistema, gestionar usuarios y mantener limpio el entorno operativo de manera recurrente, así que automatizar estos procesos no solo mejora la eficiencia, sino que también reduce el riesgo de error humano y asegura una gestión más organizada de los recursos del sistema.

## Descripción del Proyecto

El proyecto proporciona un script principal que consta de un menú de tareas para simplificar la interacción del usuario con algunas de las operaciones de administración. Las tareas incluidas son:

1. **Generar un Informe del Sistema:** Este script le ofrece al usuario un archivo .log que se crea automáticamente con información detallada sobre el uso actual de CPU, memoria y disco. Este informe se puede usar para monitorear el estado general del sistema y diagnosticar posibles problemas de rendimiento.
2. **Crear y Gestionar Usuarios:** Este script permite la creación de nuevos usuarios en el sistema y ofrece opciones para gestionar sus permisos y su directorio personal. Al utilizar este script, el administrador puede establecer contraseñas, definir permisos para el directorio personal y asignar al usuario a grupos adicionales, facilitando un control y administración eficientes de los múltiples usuarios en el sistema.
3. **Eliminar Archivos Temporales y la Caché del sistema y navegadores:** Este script limpia los archivos temporales y la caché generados por el sistema y los navegadores. Esta limpieza regular ayuda a liberar espacio y puede mejorar el rendimiento general del sistema.

## Objetivos generales del proyecto

Los objetivos generales son automatizar la gestión y el mantenimiento básico del sistema mediante la creación de usuarios, generación de informes de uso del sistema y limpieza de archivos temporales y caché, permitiendo optimizar el rendimiento y el control del sistema.

### Objetivos específicos de cada script

- **Generar un informe del sistema:**
  - Mostrar el uso actual de la CPU.
  - Visualizar el uso de la memoria RAM.
  - Mostrar el uso de los discos duros.
  - Guardar el informe en un archivo llamado `informe_sistema.log`.
- **Gestionar usuarios del sistema:**
  - Permitir la creación de nuevos usuarios solicitando el nombre y la contraseña.
  - Crear el directorio personal del usuario y asignar permisos específicos.
- **Realizar limpieza del sistema:**
  - Eliminar archivos temporales y caché del sistema para liberar espacio en disco.
  - Eliminar la caché de navegadores como Google Chrome y Firefox si están presentes.

El menú que es el script principal `menutareas.sh` facilita la selección y ejecución de cada una de estas tareas.

# Ejecución de los Scripts en Linux

## Requisitos técnicos, herramientas y dependencias necesarias para ejecutar los scripts

### 1. Visual Studio Code

Es un editor de código ligero y versátil, ideal para trabajar con múltiples lenguajes de programación. En este caso, lo utilizamos para escribir, editar y gestionar los scripts en Bash.

### 2. BashDebug

Es una extensión de VS Code que te permite depurar scripts de Bash. La depuración consiste en ejecutar el código paso a paso para encontrar errores y comprender mejor cómo se comporta en tiempo de ejecución.

Nos facilitó encontrar y corregir errores lógicos o de ejecución. Permite inspeccionar variables en tiempo real y ver cómo evolucionan durante la ejecución del script. También pueden establecer puntos de interrupción (breakpoints) para pausar el script y analizar su comportamiento línea por línea.

### 3. ShellCheck

Es una herramienta estática para analizar scripts en Bash y detectar errores comunes o malas prácticas. ShellCheck proporciona sugerencias para mejorar la calidad y la seguridad del código.

Nos ayudó a detectar errores sintácticos antes de ejecutar el script. Alerta sobre riesgos de seguridad (como uso incorrecto de variables sin comillas). Ofrece recomendaciones de buenas prácticas, mejorando la portabilidad y la eficiencia del script.

## 5. Ubuntu

Ubuntu es un sistema operativo basado en Linux que es ampliamente utilizado por su estabilidad, seguridad y facilidad de uso. En nuestro trabajo, Ubuntu sirve como el entorno en el que ejecutamos los scripts en Bash y otras herramientas. Nos proporcionó un entorno compatible con Bash, necesario para el desarrollo y prueba de scripts. Este sistema ofrece una terminal para ejecutar comandos, depurar scripts y realizar tareas administrativas. Facilita la instalación de herramientas como Visual Studio Code, ShellCheck y BashDebug mediante sus repositorios. Ubuntu es ideal tanto para desarrollo como para administración de sistemas.

## Pasos para ejecutar los scripts en Linux

Para la correcta ejecución de los scripts de y dentro de `menutareas.sh` se deben seguir los pasos que se detallan a continuación.

### 1. Guardar los Scripts

Primero guarda el código del script principal y sus scripts secundarios en una carpeta con una ruta conocida y con extensión `.sh`. Por ejemplo:

- Script principal: `menutareas.sh`
- Script informe del sistema: `informe.sh`
- Script crear y gestionar usuarios: `usuarios.sh`
- Script limpieza de archivos: `limpieza.sh`

### 2. Abrir la Terminal (Shell)

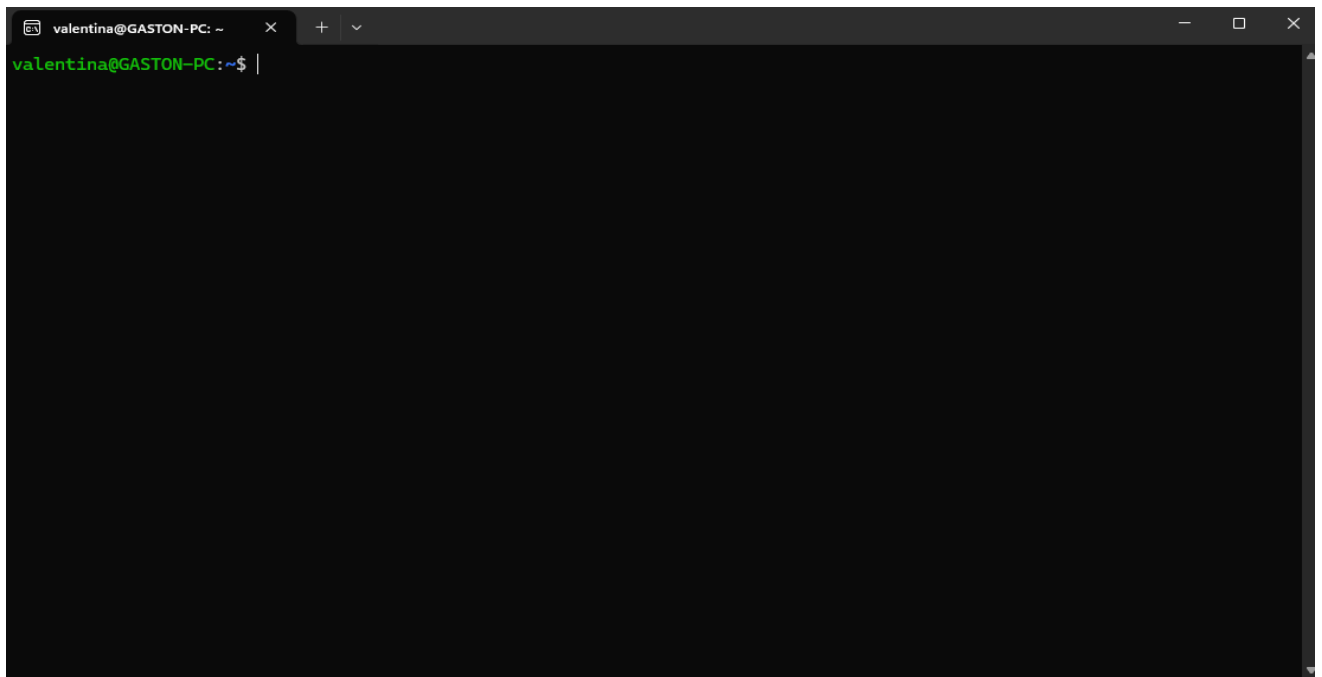
Como segundo paso, abre la terminal de tu sistema Linux para acceder a la línea de comandos (shell). Esto puede hacerse desde el menú de aplicaciones, buscando "Terminal" o "Shell". O para sistemas Windows descargar un WSL (Windows Subsystem for Linux) como puede ser "Ubuntu" para poder ingresar a la terminal de Linux.

Sistemas Operativos

Lucas Martin Treser

Comisión B

Yamila Sanchez, Valentina olmos, Giuliana Manzo

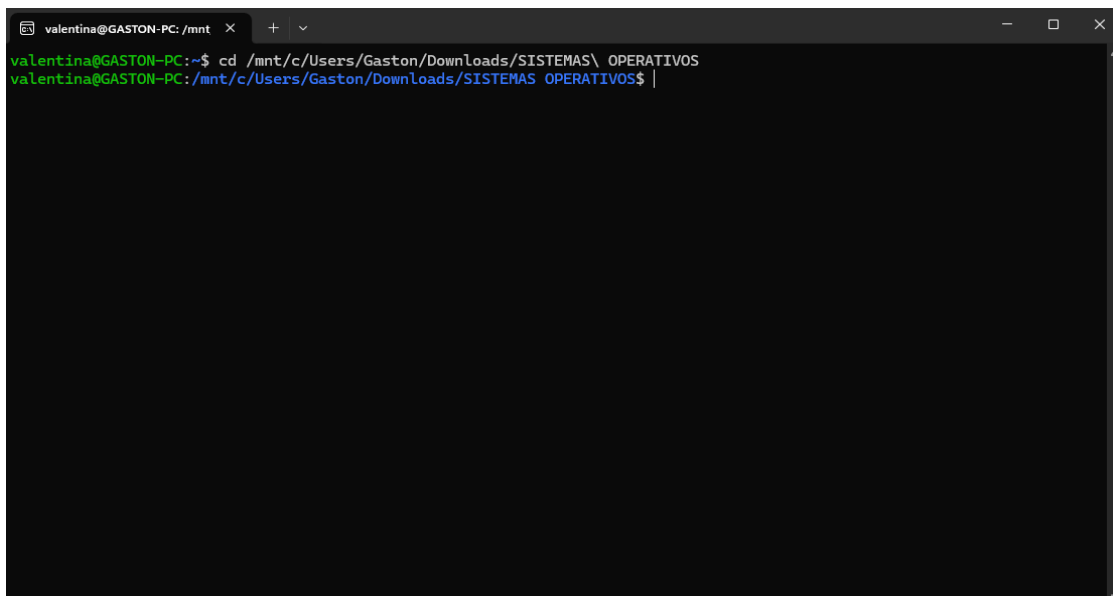


```
valentina@GASTON-PC: ~$
```

### 3. Accede a la Ruta del Script

Dirígete a la ruta donde guardaste los archivos del script. Utilizando el comando

```
cd /ruta/del/script
```

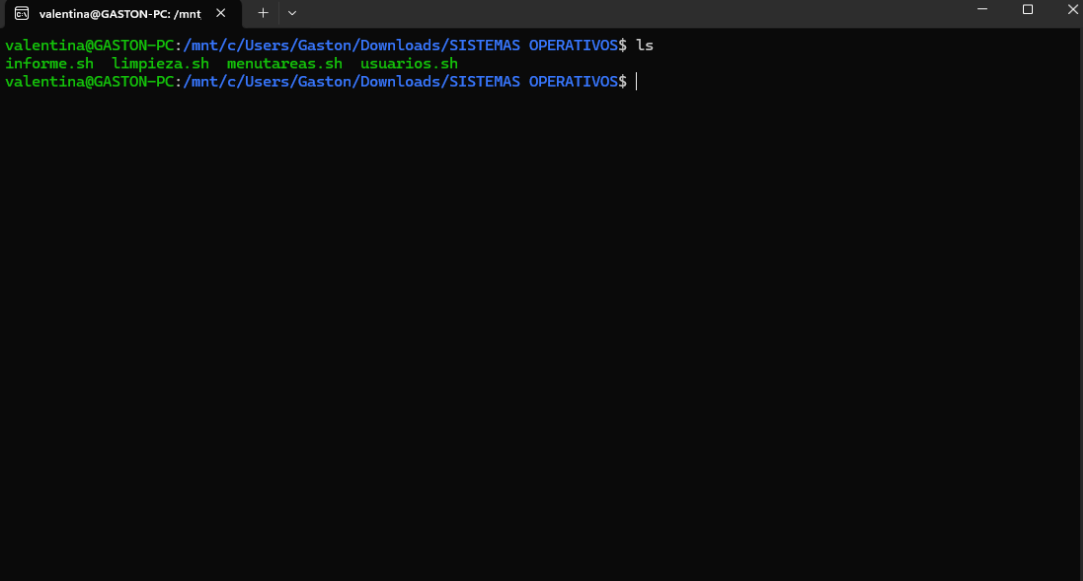


```
valentina@GASTON-PC: ~$ cd /mnt/c/Users/Gaston/Downloads/SISTEMAS\ OPERATIVOS
valentina@GASTON-PC: /mnt/c/Users/Gaston/Downloads/SISTEMAS OPERATIVOS$
```

#### 4. Verificar que los Scripts estén descargados correctamente (opcional):

Como tarea opcional puedes verificar que estén todos correctamente en la carpeta, ejecutando el siguiente comando:

`ls`

A screenshot of a terminal window titled 'valentina@GASTON-PC: /mnt'. The prompt is 'valentina@GASTON-PC:/mnt/c/Users/Gaston/Downloads/SISTEMAS OPERATIVOS\$'. The command 'ls' has been entered, and the output is 'informe.sh limpieza.sh menutareas.sh usuarios.sh'.

```
valentina@GASTON-PC:/mnt/c/Users/Gaston/Downloads/SISTEMAS OPERATIVOS$ ls
informe.sh  limpieza.sh  menutareas.sh  usuarios.sh
valentina@GASTON-PC:/mnt/c/Users/Gaston/Downloads/SISTEMAS OPERATIVOS$ |
```

#### 5. Asignar Permisos a los Scripts:

Ahora como quinto paso y antes de ejecutarlos hay que otorgarles a todos los scripts sus permisos de ejecución. Para hacerlo y teniendo en cuenta los nombres que les pusimos anteriormente se usa el siguiente comando:

`chmod +x menutareas.sh usuarios.sh informe.sh limpieza.sh`

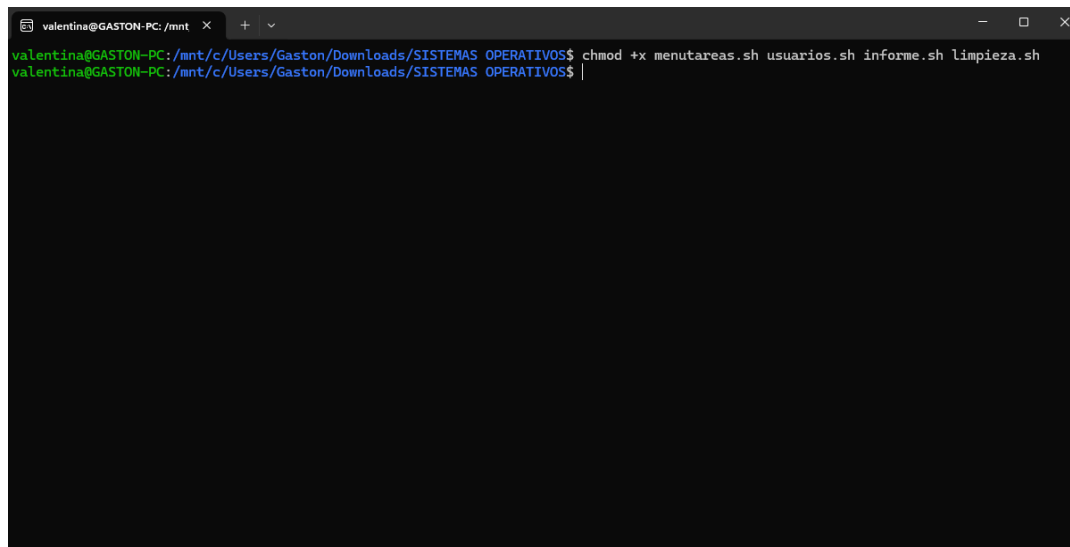


Sistemas Operativos

Lucas Martin Treser

Comisión B

Yamila Sanchez, Valentina olmos, Giuliana Manzo

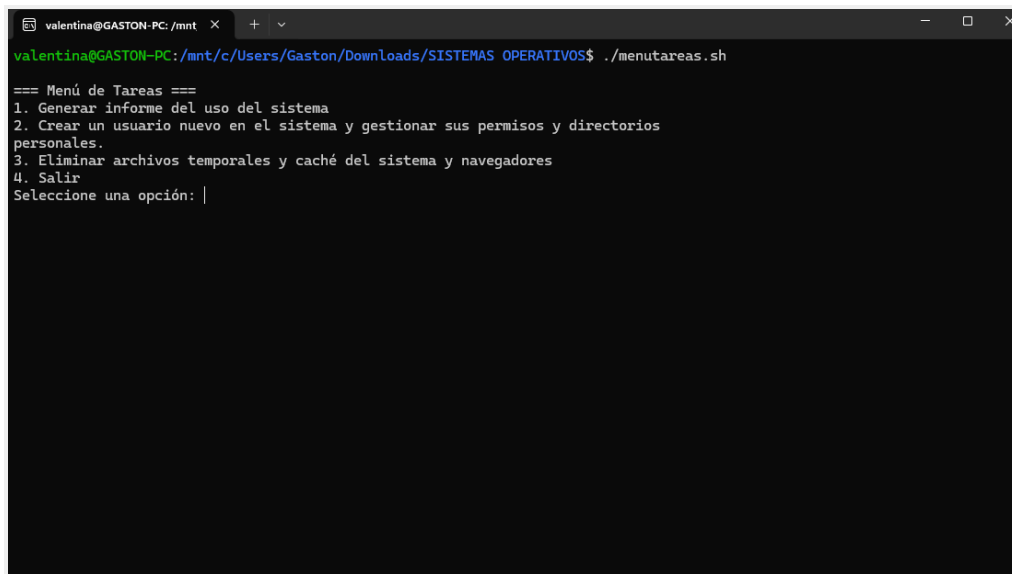


```
valentina@GASTON-PC: /mnt x + v
valentina@GASTON-PC: /mnt/c/Users/Gaston/Downloads/SISTEMAS OPERATIVOS$ chmod +x menutareas.sh usuarios.sh informe.sh limpieza.sh
valentina@GASTON-PC: /mnt/c/Users/Gaston/Downloads/SISTEMAS OPERATIVOS$
```

## 6. Ejecución del script:

Una vez que se haya verificado que todos los scripts están presentes en la carpeta asignada y que tengan sus correspondientes permisos, se puede ejecutar el script principal mediante el siguiente comando:

```
./menutareas.sh
```



```
valentina@GASTON-PC: /mnt x + v
valentina@GASTON-PC: /mnt/c/Users/Gaston/Downloads/SISTEMAS OPERATIVOS$ ./menutareas.sh

=== Menú de Tareas ===
1. Generar informe del uso del sistema
2. Crear un usuario nuevo en el sistema y gestionar sus permisos y directorios personales.
3. Eliminar archivos temporales y caché del sistema y navegadores
4. Salir
Seleccione una opción: |
```

## Pruebas de las funcionalidades del script

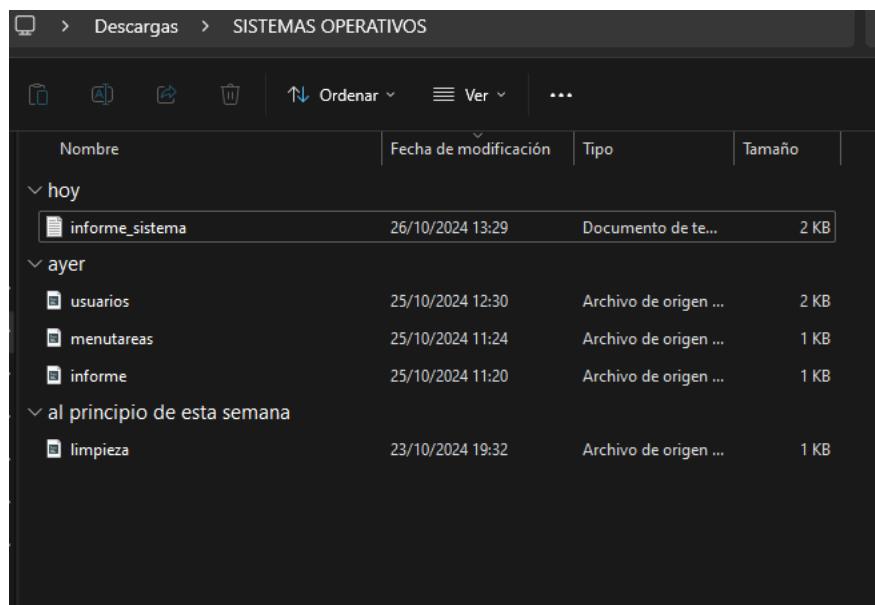
Para asegurar que los scripts funcionen correctamente y cumplan con su propósito, a continuación, se describen y muestran los métodos utilizados y los resultados obtenidos.

### 1. Generar un informe con el uso actual de la CPU, memoria y disco, y guardarlo en un archivo log.

Este script genera un informe del uso del sistema y lo guarda en un archivo .log, incluyendo información sobre la CPU, memoria y disco.

```
=== Menú de Tareas ===  
1. Generar informe del uso del sistema  
2. Crear un usuario nuevo en el sistema y gestionar sus permisos y directorios personales.  
3. Eliminar archivos temporales y caché del sistema y navegadores  
4. Salir  
Seleccione una opción: 1  
Generando informe del sistema...  
Informe generado en 'informe_sistema.log'.
```

El archivo log se guarda en la ubicación donde están los scripts



```
informe_sistema.log
Archivo  Editar  Ver

=== Informe del Sistema - Sat Oct 26 16:29:00 UTC 2024 ===

=== Uso de CPU ===
top - 16:29:00 up 43 min, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 27 total, 1 running, 26 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 6881.2 total, 6243.9 free, 638.8 used, 230.4 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 6242.4 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
    1 root        20   0   21692  12856  9532  S   0.0   0.2   0:00.41 systemd
    2 root        20   0    2616   1440  1320  S   0.0   0.0   0:00.00 init-sy+
    6 root        20   0    2616    132   132  S   0.0   0.0   0:00.00 init

=== Uso de Memoria ===
              total        used        free      shared  buff/cache   available
Mem:          6.7Gi        638Mi        6.1Gi         3.0Mi        230Mi        6.1Gi
Swap:          2.0Gi           0B         2.0Gi

=== Uso de Disco ===
Filesystem      Size  Used Avail Use% Mounted on
none            3.4G   0    3.4G   0% /usr/lib/modules/5.15.153.1-microsoft-standard-WSL2
none            3.4G  4.0K   3.4G   1% /mnt/wsl
drivers         931G  676G  255G  73% /usr/lib/wsl/drivers
/dev/sdc        1007G  1.8G  954G   1% /
none            3.4G   80K   3.4G   1% /mnt/wslg
none            3.4G   0    3.4G   0% /usr/lib/wsl/lib
rootfs          3.4G  2.2M   3.4G   1% /init
none            3.4G  496K   3.4G   1% /run
none            3.4G   0    3.4G   0% /run/lock
none            3.4G   0    3.4G   0% /run/shm
tmpfs           4.0M   0    4.0M   0% /sys/fs/cgroup
none            3.4G   96K   3.4G   1% /mnt/wslg/versions.txt
none            3.4G   96K   3.4G   1% /mnt/wslg/doc
C:\             931G  676G  255G  73% /mnt/c

Ln 1, Col 1  1.855 caracteres.  100%  Unix (LF)  UTF-8
```

## 2. Crear usuarios nuevos en el sistema y gestionar sus permisos y directorios personales.

Este script permite crear nuevos usuarios en el sistema y gestionar sus permisos y directorios personales.

```
valentina@GASTON-PC: /mnt
valentina@GASTON-PC: /mnt/c/Users/Gaston/Downloads/SISTEMAS OPERATIVOS$ ./menutareas.sh

=== Menú de Tareas ===
1. Generar informe del uso del sistema
2. Crear un usuario nuevo en el sistema y gestionar sus permisos y directorios personales.
3. Eliminar archivos temporales y caché del sistema
4. Salir
Seleccione una opción: 2
1. Crear nuevo usuario
2. Salir
Seleccione una opción: 1
Ingrese el nombre del nuevo usuario: camila
Ingrese la contraseña para el nuevo usuario:
[sudo] password for valentina:
```

Se ingresan los datos solicitados (nombre de usuario y contraseña) para el nuevo usuario y contraseña del usuario actual para confirmar los cambios.

```
Seleccione los permisos para el directorio personal:
1. Solo el usuario (700)
2. Usuario y grupo (770)
3. Todos (777)
Opción: 3
Usuario 'camila' creado exitosamente con su directorio personal.
```

Se presenta al usuario un menú con tres opciones para elegir los permisos del directorio personal del nuevo usuario:

Para 700: Solo el propietario tiene permisos de lectura, escritura y ejecución.

Para 770: El propietario y el grupo tienen permisos de lectura, escritura y ejecución; otros no tienen permisos.

Para 777: Todos los usuarios tienen permisos de lectura, escritura y ejecución.

Luego de esto el usuario ha sido creado exitosamente

### **3. Eliminar archivos temporales, caché del sistema y de navegadores para liberar espacio en disco.**

Este script está diseñado para realizar una limpieza eficiente de archivos temporales y caché, tanto del sistema operativo como de los navegadores web más comunes que son google chrome y firefox.

```
=== Menú de Tareas ===
1. Generar informe del uso del sistema
2. Crear un usuario nuevo en el sistema y gestionar sus permisos y directorios personales.
3. Eliminar archivos temporales y caché del sistema
4. Salir
Seleccione una opción: 3
Iniciando limpieza de archivos temporales y caché del sistema y navegadores...
Eliminando archivos temporales y cache del sistema...
Limpieza completada. Se ha liberado espacio en disco.
```

## Desarrollo y explicación detallada del código.

### 1) Script `menutareas.sh`

Este script principal ejecuta uno de los tres scripts secundarios en función de la opción seleccionada por el usuario en el menú interactivo. Cada script secundario se encarga de realizar una tarea específica:

```
1  #!/bin/bash
2
3  while true; do
4      echo -e "\n=== Menú de Tareas ==="
5      echo "1. Generar informe del uso del sistema"
6      echo "2. Crear un usuario nuevo en el sistema y gestionar sus permisos y directorios
7  personales."
8      echo "3. Eliminar archivos temporales y caché del sistema y navegadores"
9      echo "4. Salir"
10     read -r -p "Seleccione una opción: " opcion
11
12     case $opcion in
13         1) ./informe.sh ;;
14         2) ./usuarios.sh ;;
15         3) ./limpieza.sh ;;
16         4) echo "Saliendo..."; break ;;
17         *) echo "Opción no válida. Intente nuevamente." ;;
18     esac
19 done
```

- **#!/bin/bash:** Es la línea "shebang", que le indica al sistema que use el intérprete de Bash para ejecutar este script.
- **while true; do:** Inicia un bucle infinito (while loop) que repetirá continuamente el menú hasta que el usuario elija salir, por otro lado true significa que la condición del bucle siempre es verdadera, por lo que se ejecutará indefinidamente.
- **echo -e "\n=== Menú de Tareas ===":** Imprime un encabezado para el menú de tareas. El flag -e permite que echo interprete caracteres especiales como \n, que crea una nueva línea antes del encabezado.
- **read -r -p "Seleccione una opción: " opcion :** Muestra el mensaje "Seleccione una opción: " y espera a que el usuario ingrese una para almacenar su entrada en la variable opcion.

- **case \$opcion in ... esac:** Evalúa el valor de “opcion” y ejecuta el script correspondiente:
- **1) ./informe.sh ;::** Llama al script informe.sh para generar el informe del sistema.
- **2) ./usuarios.sh ;::** Llama al script usuarios.sh para gestionar usuarios.
- **3) ./limpieza.sh ;::** Llama al script limpieza.sh para limpiar archivos.
- **4) echo "Saliendo..."; break ;::** Muestra "Saliendo..." y usa break para salir del bucle y finalizar el menú.
- **\*) echo "Opción no válida. Intente nuevamente." ;::** Si la opción no es válida, muestra un mensaje de error.
- **done:** Marca el final del bucle while, volviendo a mostrar el menú hasta que el usuario elija la opción de salir (opción 4).

## 2)Script `informe.sh`

Este script genera un informe básico del uso de recursos del sistema y guarda los resultados en un archivo `.log`.

```
1  #!/bin/bash
2  # Generar informe del uso del sistema
3  echo "Generando informe del sistema..."
4  {
5      echo "=== Informe del Sistema - $(date) ==="
6
7      # Información de la CPU
8      echo -e "\n=== Uso de CPU ==="
9      top -bn1 | head -10
10
11     # Información de la memoria
12     echo -e "\n=== Uso de Memoria ==="
13     free -h
14
15     # Información del disco
16     echo -e "\n=== Uso de Disco ==="
17     df -h
18 } >> informe_sistema.log
19
20 # Finalización
21 echo "Informe generado en 'informe_sistema.log'."
```

- **#!/bin/bash**: indica que es un script de Bash.
- **echo "Generando informe del sistema..."**: Le notifica al usuario que el proceso de generación de informes está comenzando.
- **echo "=== Informe del Sistema - \$(date) ==="**: Escribe en el archivo el encabezado "Informe del Sistema" seguido de la fecha y hora actuales obtenidas por `$(date)`.
- **echo -e "\n=== Uso de CPU ==="**: Imprime el título "Uso de CPU" y un salto de línea (`\n`) para separar esta sección en el archivo.

- **top -bn1 | head -10**: La primera parte (top -bn1) ejecuta el comando top en modo batch ejecutándose una vez sola para obtener un solo informe de uso de CPU en tiempo real. Y la segunda parte | head -10 toma solo las primeras 10 líneas de la salida de top, que usualmente incluye información de los procesos activos.
- **echo -e "\n=== Uso de Memoria ==="**: Muestra el título "Uso de Memoria" en el archivo de log.
- **free -h**: Ejecuta el comando free con la opción -h para mostrar el uso de la memoria de forma legible (con unidades como MB o GB).
- **echo -e "\n=== Uso de Disco ==="**: Muestra el título "Uso de Disco".
- **>> informe\_sistema.log** : El bloque completo de { ... } se redirige al archivo [informe\\_sistema.log](#), lo que guarda toda la información de CPU, memoria y disco en este archivo.
- **echo "Informe generado en 'informe\_sistema.log'."**: Muestra en la terminal que el informe ha sido generado y guardado en el archivo.



### 3) Script `usuarios.sh`

Este script permite crear nuevos usuarios en el sistema, asignarles contraseñas y configurar los permisos de su directorio personal.

```
1  #!/bin/bash
2  crear_usuario() {
3      read -p "Ingrese el nombre del nuevo usuario: " username
4      read -sp "Ingrese la contraseña para el nuevo usuario: " password
5      echo
6
7      sudo useradd -m -s /bin/bash "$username"
8
9      echo "$username:$password" | sudo chpasswd
10
11     echo "Seleccione los permisos para el directorio personal:"
12     echo "1. Solo el usuario (700)"
13     echo "2. Usuario y grupo (770)"
14     echo "3. Todos (777)"
15     read -p "Opción: " permiso_opcion
16
17     case $permiso_opcion in
18         1)
19             sudo chmod 700 /home/"$username"
20             ;;
21         2)
22             sudo chmod 770 /home/"$username"
23             ;;
24         3)
25             sudo chmod 777 /home/"$username"
26             ;;
27         *)
28             echo "Opción inválida. Se asignará el permiso 700 por defecto."
29             sudo chmod 700 /home/"$username"
30             ;;
31     esac
32
33     echo "Usuario '$username' creado exitosamente con su directorio personal."
34
35
36  mostrar_menu() {
37      echo "1. Crear nuevo usuario"
38      echo "2. Salir"
39  }
40
41  while true; do
42      mostrar_menu
43      read -p "Seleccione una opción: " opcion
44
45      case $opcion in
46          1)
47              crear_usuario
48              ;;
49          2)
50              echo "Saliendo..."
51              exit 0
52              ;;
53          *)
54              echo "Opción inválida. Por favor, seleccione una opción válida."
55              ;;
56      esac
57  done
```

- **#!/bin/bash**: Línea shebang que le indica al sistema que el script debe ejecutarse usando el intérprete de Bash.
- **crear\_usuario()**: Define la función crear\_usuario, que contiene el proceso principal de creación de usuario y asignación de permisos.
- **read -p "Ingrese el nombre del nuevo usuario: " username**: Solicita el nombre del nuevo usuario al usuario que ejecuta el script, mostrando el mensaje "Ingrese el nombre del nuevo usuario: ". El nombre ingresado se almacena en la variable username.
- **read -sp "Ingrese la contraseña para el nuevo usuario: " password**: Solicita la contraseña para el nuevo usuario. La opción -s evita que la contraseña se muestre en pantalla mientras se escribe, y la opción -p muestra el mensaje "Ingrese la contraseña para el nuevo usuario: ". La contraseña se guarda en la variable password.
- **sudo useradd -m -s /bin/bash "\$username"**: Crea un nuevo usuario usando useradd con las opciones:
  - **-m**: Crea un directorio personal para el usuario en /home/<nombre\_de\_usuario>.
  - **-s /bin/bash**: Establece Bash como el intérprete de comandos predeterminado del nuevo usuario.
  - **"\$username"**: Nombre del usuario que se asigna usando el valor almacenado en la variable username.
- **echo "\$username:\$password" | sudo chpasswd**: Establece la contraseña del nuevo usuario usando chpasswd, que toma como entrada la combinación nombre\_usuario:contraseña. Esto permite asignar la contraseña almacenada en password al usuario username.

- **case \$permiso\_opcion in ... esac:** Estructura case que verifica el valor de permiso\_opcion y ejecuta comandos de acuerdo a la elección:
  - **1):** Si el usuario elige 1, se ejecuta `sudo chmod 700 /home/"$username"`, otorgando solo al usuario acceso total a su directorio personal.
  - **2):** Si el usuario elige 2, se ejecuta `sudo chmod 770 /home/"$username"`, dando acceso total al usuario y al grupo.
  - **3):** Si el usuario elige 3, se ejecuta `sudo chmod 777 /home/"$username"`, dando acceso a todos.
  - **\*):** Si la entrada no es válida, muestra un mensaje y aplica el permiso 700 como valor predeterminado.
- **while true; do ... done:** Bucle infinito (while) que ejecuta el menú de opciones hasta que el usuario elija salir.
- **mostrar\_menu:** Llama a la función mostrar\_menu para mostrar las opciones.
- **read -p "Seleccione una opción: " opcion :** Solicita la selección del usuario y la guarda en la variable opcion.
- **case \$opcion in ... esac:** Verifica el valor de opcion y ejecuta comandos según la elección:
  - **1):** Ejecuta crear\_usuario para crear un nuevo usuario.
  - **2):** Muestra "Saliendo..." y usa exit 0 para salir del script.
  - **\*):** Muestra un mensaje de error si la entrada no es válida, y repite el menú.

## 4) Script `limpieza.sh`

Este script realiza la limpieza de archivos temporales y de caché del sistema, así como de navegadores como Google Chrome y Firefox.

```
1  #!/bin/bash
2
3  echo "Iniciando limpieza de archivos temporales y caché del sistema y navegadores..."
4
5  echo "Eliminando archivos temporales y cache del sistema..."
6  sudo rm -rf /tmp/*
7  sudo rm -rf ~/.cache/*
8
9  if [ -d ~/.cache/google-chrome ]; then
10     echo "Eliminando caché de Google Chrome..."
11     sudo rm -rf ~/.cache/google-chrome/*
12 fi
13
14 if [ -d ~/.cache/mozilla/firefox ]; then
15     echo "Eliminando caché de Firefox..."
16     sudo rm -rf ~/.cache/mozilla/firefox/*
17 fi
18
19
20 echo "Limpieza completada. Se ha liberado espacio en disco."
```

- **#!/bin/bash**: Shebang que indica al sistema que el script debe ejecutarse usando el intérprete de Bash.
- **echo "Eliminando archivos temporales y cache del sistema..."**: Imprime un mensaje para señalar que se está eliminando la caché y los archivos temporales del sistema.
- **sudo rm -rf /tmp/\***: Elimina todos los archivos y carpetas en el directorio /tmp, usado temporalmente por el sistema y los programas.
- **sudo rm -rf ~/.cache/\***: Borra todo el contenido de la carpeta ~/.cache del usuario, donde se almacenan datos temporales para mejorar el rendimiento.
- **if [ -d ~/.cache/google-chrome ]; then**: Verifica si la carpeta ~/.cache/google-chrome existe (lo que indica que Chrome ha sido usado en el sistema).

- **echo "Eliminando caché de Google Chrome...":** Imprime un mensaje si la carpeta existe.
- **sudo rm -rf ~/.cache/google-chrome/\*:** borra los archivos dentro de esta carpeta para eliminar la caché acumulada.
- **fi:** Cierra la condición if.
- **if [ -d ~/.cache/mozilla/firefox ]; then:** Verifica si la carpeta ~/.cache/mozilla/firefox existe (indica que Firefox ha sido usado en el sistema).
- **echo "Eliminando caché de Firefox...":** Imprime un mensaje si la carpeta existe.
- **sudo rm -rf ~/.cache/mozilla/firefox/\*:** borra los archivos dentro de esta carpeta para eliminar la caché de Firefox.
- **echo "Limpieza completada. Se ha liberado espacio en disco.":** Muestra un mensaje final para informar que la limpieza de archivos temporales y caché ha terminado y que se ha liberado espacio en disco.

## **Reflexiones finales sobre el desarrollo**

Aprendimos que la mezcla de aprender sobre sistemas operativos y ponerlo en práctica con scripts de Bash muestra lo útiles y poderosas que son estas herramientas. Cada vez que ejecutamos un script, pudimos ver cómo podemos automatizar tareas diarias, hacer el trabajo más fácil y ser más eficientes en la gestión de sistemas. Y saber hacer esto es clave para cualquier persona que trabaje en tecnología, ya que ayuda a lidiar con los problemas diarios de una manera más sencilla.

## **Dificultades encontradas y posibles mejoras.**

### **Compatibilidad entre diferentes entornos**

Un desafío importante para nosotras fue la compatibilidad entre diferentes entornos, ya que usamos todas un sistema operativo distinto al que se necesita para ejecutar los scripts. Esto provocó incompatibilidades que impidieron que los scripts funcionaran de forma uniforme. La solución consistió en instalar Ubuntu (es un WSL, Windows Subsystem for Linux), lo que permitió emular un entorno Linux en sistemas Windows y facilitó la ejecución y prueba de los scripts en un entorno más similar al de producción.

### **Dificultades con la Comprensión del Código en Bash**

Una de las dificultades que enfrentamos fue entender y trabajar con el código en Bash, ya que ninguna de nosotras tenía experiencia previa en este lenguaje. La falta de conocimiento en Bash generó errores, lo que ralentizó el desarrollo y requirió tiempo adicional para investigar y aprender sobre el lenguaje. Y para abordar esta dificultad, recurrimos a recursos en línea y documentación específica de Bash, lo que nos ayudó a adquirir los conocimientos necesarios para completar el proyecto.

### **Comunicación Limitada sobre los Cambios**

Finalmente, la comunicación y coordinación del equipo podría mejorarse en proyectos futuros. La falta de claridad respecto a algunos objetivos ocasionó confusiones, por lo que llevar a cabo reuniones para establecer los objetivos definidos puede fortalecer la colaboración y evitar problemas en la implementación de cambios.

Sistemas Operativos

Lucas Martin Treser

Comisión B

Yamila Sanchez, Valentina olmos, Giuliana Manzo

## Bibliografía

- <https://www.hostinger.es/tutoriales/bash-script-linux>
- <https://www.freecodecamp.org/espanol/news/tutorial-de-programacion-de-bash-script-de-shell-de-linux-y-linea-de-comandos-para-principiantes/>
- <https://bioinfo.fmed.uba.ar/docs/guia-del-cluster/bash-scripting/>
- [https://youtu.be/0tIZhTAuNuU?si=piGOCb\\_LPJmo-iQX](https://youtu.be/0tIZhTAuNuU?si=piGOCb_LPJmo-iQX)
- [https://youtu.be/EKFK83mNsyo?si=uYqOo83W3GOE3W\\_9](https://youtu.be/EKFK83mNsyo?si=uYqOo83W3GOE3W_9)
- <https://youtu.be/rOHQ4Jof3gE?si=jZJLPoFBqHxgST5E>
- <https://code.visualstudio.com/>
- <https://www.hostinger.com.ar/tutoriales/que-es-ubuntu>
- <https://ubuntu.com/>
- [https://youtu.be/JdXqCHT1rq0?si=bKB5\\_oFZYanjOPRT](https://youtu.be/JdXqCHT1rq0?si=bKB5_oFZYanjOPRT)
- [https://youtu.be/44ziZ12rJwU?si=JSLOXCL\\_pkZ5JpNQ](https://youtu.be/44ziZ12rJwU?si=JSLOXCL_pkZ5JpNQ)