Trabajo Final de Sistemas Operativos (Scripts automatizados)
Alumnos: Teo Cagnolo y German Rodriguez

1. ***Introducción del proyecto***
   El proyecto tiene como objetivo desarrollar 3 scripts en Bash que automaticen tareas básicas de administración del sistema operativo Linux.
   Se implementaron tres scripts principales: info.sh, backup.sh y clean.sh, los cuales permiten obtener información del sistema, realizar copias de respaldo automatizadas, y limpiar archivos temporales y cache; El menu principal (menu.sh) integra las funcionalidades de manera interactiva, con validaciones de entrada y uso de colores para mejorar la experiencia de usuario.

2. ***Requisitos técnicos, Herramientas y Dependencias***
   El proyecto fue desarrollado en **entorno Linux (Raspberry Pi OS)**, utilizando el **intérprete Bash** y herramientas básicas incluidas en la mayoría de las distribuciones.
   - Sistema operativo: Debian (32-bit) o Raspberry Pi OS.
     Memoria Ram: 4gb
     Disco duro virtual: 20gb
   - Intérprete de comandos: Bash
   - Comandos utilizados:
     .tar .gz (crea un archivo comprimido)
     find (Busca archivos viejos y los borra)
     tee -a (Muestra mensajes y los guarda en el log)
     dos2unix (Convierte archivos Windows a formato Linux)
   - Dependencias:
     dos2unix: instalar (sudo apt install dos2unix -y) ejecutar (dos2unix *.sh)
   - Permisos requeridos: chmod +x menu.sh info.sh clean.sh backup.sh

3. ***Desarrollo y explicación del código***
   Cada script cumple una función específica dentro del sistema, y juntos hacen un conjunto de herramientas básicas de administración automatizada

   - Info.sh
     Este script genera un informe del sistema y lo guarda en la carpeta `logs/`.

     Crea el directorio logs si no existe.
     Crea un archivo con nombre basado en la fecha actual (`info-YYYYMMDD-HHMMSS.log`).

Usa comandos simples (`echo`, `date`, `uname -r`) y muestra la versión del sistema operativo, la versión del Kernel y la Información general de CPU, memoria y disco.
Toda la salida se guarda en log y se informa la ubicación del archivo generado

- Backup.sh
  Este script está encargado de automatizar la creación de copias de respaldo y gestionar la eliminación de backups antiguos.

  Define una carpeta de origen (`SRC`) y una de destino (`DEST`).
  Crea ambas carpetas si no existen.
  Comprime la carpeta de origen dentro de `backups/` usando `tar -czf`.
  Asigna nombres únicos basados en la fecha (`backup-YYYYMMDD-HHMMSS.tar.gz`).
  Implementa una rotación simple: elimina automáticamente los backups con más de 7 días utilizando `find -mtime`.

- Clean.sh
  Este script está orientado a la limpieza de archivos temporales y cachés.

  Crea un archivo log con sello de tiempo (`clean-YYYYMMDD-HHMMSS.log`).
  Borra archivos temporales en `/tmp`.
  Limpia la caché del usuario en `~/.caché`.
  Elimina cachés de navegadores comunes como Firefox y Chromium si existen.
  Usa `tee -a` para mostrar en pantalla y escribir en el log al mismo tiempo.

- Menu.sh
  Script principal que funciona como **interfaz interactiva** entre los tres anteriores.

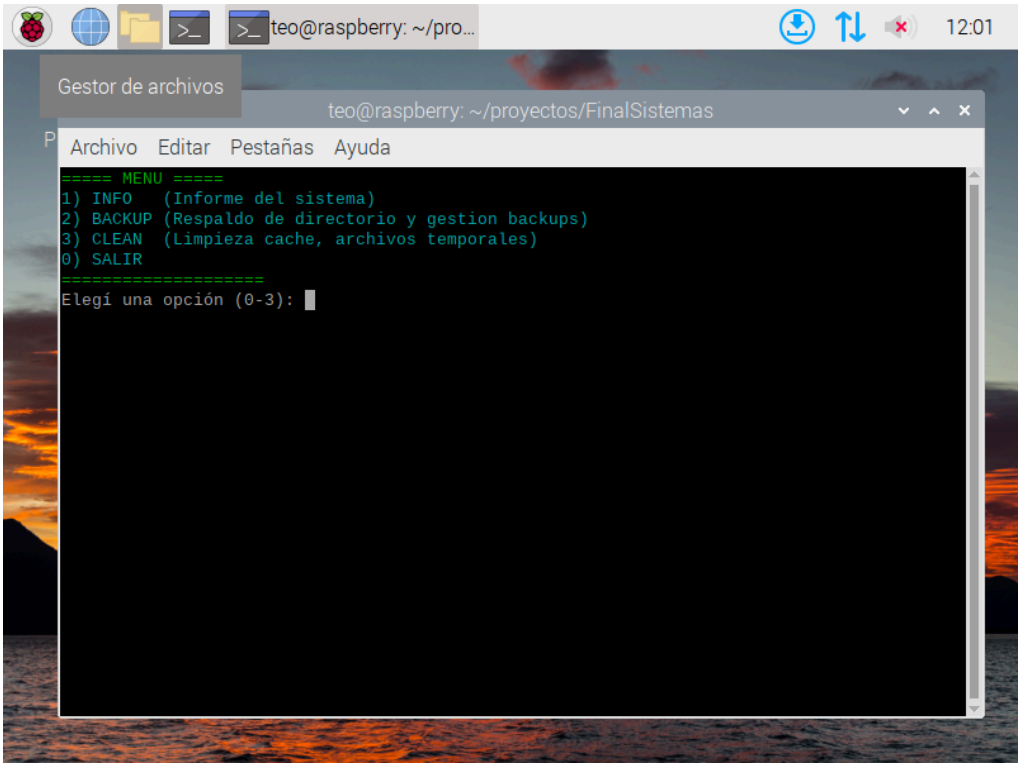
  Ofrece opciones numéricas (1 a 3) y una opción 0 para salir.
  Valida la entrada del usuario (solo acepta 0–3).
  Ejecuta el script correspondiente según la opción elegida.
  Pausa al finalizar cada ejecución para permitir leer el resultado antes de volver al menú.

4. ***Pruebas y validaciones***

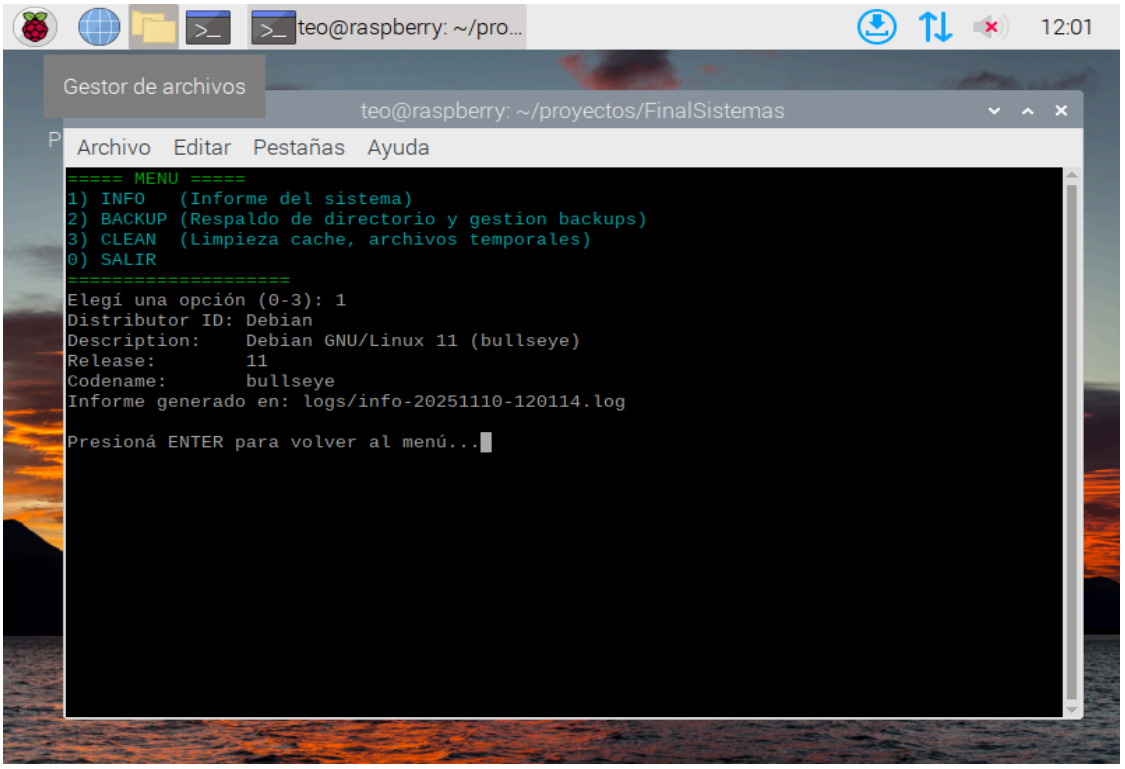
Las pruebas se realizaron dentro de una máquina virtual (virtualbox 6.1.34) con raspberry Pi OS, utilizando LXTerminal como entorno de ejecución.
Se verificó el funcionamiento de cada script de forma individual y desde el menú principal.

Resultados de las pruebas:

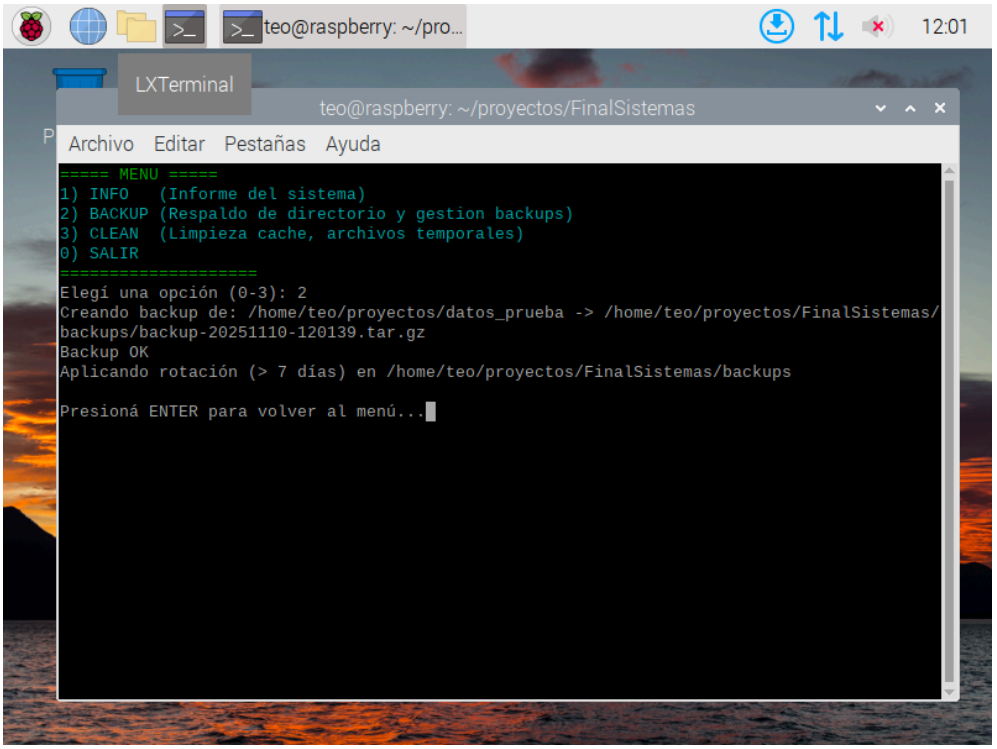
Menu de los scripts: Muestra correctamente las opciones con colores, valida entradas incorrectas y permite volver al menú después de cada ejecución.



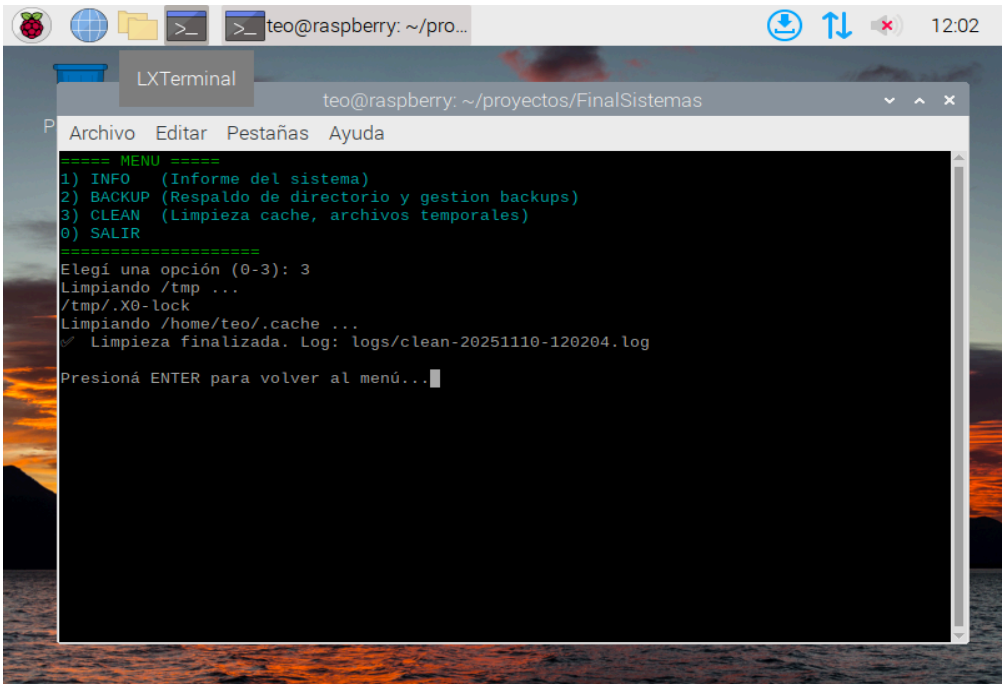
Opción 1 (Información del sistema): Muestra y genera correctamente el archivo `info-*.log` dentro de la carpeta `logs/`.



Opción 2 (Respaldo de directorio y gestión de backup): Creó el archivo comprimido `backup-*.tar.gz` dentro de `backups/` y eliminó correctamente los backups antiguos al superar el límite de 7 días.



Opcion 3 (Limpieza de cache, archivos temporales y cache de navegadores): **Se** eliminó archivos temporales y cachés, dejando registro en `clean-*.log`.



5. ***Reflexiones finales sobre el desarrollo, dificultades encontradas y posibles mejoras***

Durante el desarrollo fuimos aprendiendo a como automatizar tareas que normalmente se hacen a mano, también aprendimos la importancia de los permisos, las estructuras de las carpetas, que los archivos log sirven para dejar registro de todo lo que hace el sistema y mantener un proyecto funcionando dentro de una máquina virtual.

Las principales dificultades fueron hacer andar la máquina virtual, también tuvimos problemas cuando los archivos nos daban error por saltos de línea distintos entre windows y linux. Nos daba el error ($´/r´; orden no encontrada), investigando nos dimos cuenta que los scripts estaban guardados en formato Windows. La solución fue usar el comando (dos2unix) para convertirlos al formato adecuado.

En cuanto a posibles mejoras, se podría hacer una opción en el menú que te muestre todos los logs realizados, también estaría bueno poder elegir la carpeta a respaldar.

En resumen, el proyecto salió bien, no hubo muchos inconvenientes, ya que el resultado final fue un sistema simple, ordenado y funcional, demostrando cómo se pueden automatizar tareas del sistema con pocos comandos y algo de lógica.