



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
FACULTAD DE INGENIERÍA ELECTRÓNICA



Nombre del proyecto

Descripción del proyecto

Presentado a: Ing. Diego Alejandro Barragan Vargas

Juan Diego Báez Guerrero, Cód.: 2336781.

Resumen— El manejo de archivos y directorios es una habilidad fundamental en el uso de sistemas operativos basados en Linux. Esta práctica busca desarrollar competencias en la organización, manipulación y gestión de archivos mediante comandos de terminal. Se abordan tareas clave como la navegación en el sistema de archivos, la modificación de permisos, la búsqueda de archivos, la administración de procesos y la instalación de paquetes. Además, se enfatiza la importancia de la automatización de tareas y la documentación de procedimientos en entornos Unix.

Abstract— File and directory management is a fundamental skill in using Linux-based operating systems. This practice aims to develop competencies in organizing, manipulating, and managing files through terminal commands. Key tasks include navigating the file system, modifying permissions, searching for files, managing processes, and installing packages. Additionally, the importance of task automation and procedure documentation in Unix environments is emphasized.

I INTRODUCCIÓN

El manejo de archivos y directorios es una de las habilidades fundamentales en la administración de sistemas operativos basados en Linux. A través de la terminal, los usuarios pueden realizar tareas esenciales como la creación, modificación, eliminación y organización de archivos y directorios, así como gestionar permisos y procesos en el sistema. Estos conocimientos son clave tanto para usuarios principiantes como para administradores de sistemas, ya que permiten un control más eficiente sobre los recursos del sistema y la automatización de tareas repetitivas mediante el uso de scripts en Bash [1], [3].

Linux sigue el principio de "todo es un archivo", lo que significa que no solo los documentos y carpetas se representan como archivos, sino también dispositivos, procesos y configuraciones del sistema. Por ello, entender la estructura de archivos y directorios es fundamental para interactuar correctamente con el sistema operativo. Además, el uso de comandos como ls, cd, mkdir, rm, chmod y chown permite navegar y administrar archivos con precisión y seguridad [4], [2].

El manejo de permisos y seguridad en Linux es otro aspecto crítico. Los permisos de usuario y grupo determinan quién puede acceder, modificar o ejecutar un archivo, lo que protege la integridad del sistema y los datos. Herramientas

como chmod, chown y ls -l ayudan a gestionar estos permisos, garantizando que solo los usuarios autorizados realicen modificaciones en archivos importantes [6], [7].

Asimismo, la administración de procesos es esencial para optimizar el rendimiento del sistema. Comandos como ps, top, htop y kill permiten monitorear y gestionar tareas en ejecución, asegurando que los recursos se utilicen de manera eficiente. Estos comandos son de gran utilidad para detectar procesos que consumen muchos recursos y terminarlos cuando sea necesario. En el campo de la Ingeniería Electrónica, la gestión eficiente de procesos y recursos computacionales es fundamental para el desarrollo de aplicaciones embebidas, automatización industrial y sistemas de control en tiempo real [5], [8].

Por último, el uso de herramientas para la instalación y actualización de paquetes es un aspecto fundamental de la administración de sistemas Linux. Con apt se pueden instalar, actualizar y eliminar programas de manera sencilla, facilitando la gestión de software en el sistema. Esto es especialmente relevante en entornos de desarrollo de hardware y software embebido, donde la compatibilidad y actualización de paquetes juegan un papel crucial en la integración de sistemas digitales [2].

En este informe, se explorarán estas herramientas y conceptos a través de una serie de actividades prácticas diseñadas para reforzar el manejo de archivos y directorios, permisos, procesos y paquetes en Linux. La documentación del proceso será subida a un repositorio en GitHub para su posterior análisis y revisión.

II MARCO TEÓRICO

A. Sistema de Archivos en Linux

El sistema de archivos de Linux sigue una estructura jerárquica organizada en directorios y subdirectorios. El directorio raíz '/' es el punto de partida, y a partir de él se organizan carpetas como '/home' (directorios de usuarios), '/etc' (archivos de configuración) y '/var' (archivos de regis-



UNIVERSIDAD SANTO TOMÁS

PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

FACULTAD DE INGENIERÍA ELECTRÓNICA



tro) [1].

Para la gestión eficiente de archivos y directorios, Linux proporciona una serie de comandos esenciales, como 'ls' para listar contenido, 'cd' para cambiar de directorio y 'pwd' para mostrar la ubicación actual [2].

B. Manipulación de Archivos y Directorios

La manipulación de archivos en Linux se realiza mediante comandos como 'mkdir' para crear directorios, 'rm' para eliminarlos y 'mv' para renombrarlos o moverlos. La creación y edición de archivos puede hacerse con 'touch', 'nano' o 'vim' [3].

Los permisos de archivos en Linux se administran con 'chmod', que permite definir niveles de acceso para el propietario, grupo y otros usuarios. Esto es crucial para garantizar la seguridad y privacidad de la información almacenada en un sistema multiusuario [4].

C. Gestión de Procesos en Linux

Un proceso en Linux representa una instancia en ejecución de un programa. Se pueden listar con 'ps', 'top' o 'htop'. Para administrar procesos en segundo plano, se utilizan operadores como '&' y comandos como 'jobs', 'fg' y 'bg' [5].

La terminación de procesos se realiza con 'kill' seguido del identificador de proceso (PID). En caso de procesos que no responden, 'kill -9' se usa para forzar su detención [6].

D. Gestión de Paquetes y Automatización

El sistema de gestión de paquetes 'apt' en distribuciones basadas en Debian permite la instalación y actualización de software con comandos como 'sudo apt update' y 'sudo apt install'. Esto facilita la administración de herramientas en entornos Linux [7].

La automatización de tareas se logra mediante scripts Bash (.sh), permitiendo ejecutar múltiples comandos de manera secuencial. Los scripts pueden programarse con 'cron' para su ejecución en horarios definidos [8].

III PROCEDIMIENTO Y RESULTADOS

A. Navegación

1). Uso de CD

El primer código a ejecutar es cd el cual se utiliza para regresar al directorio personal como se evidencia en la figura 1

```
juan@juan-HP-Laptop-14-ck2xxx:~$ cd
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 1: Uso de CD. Fuente: Elaboración propia.

Se observa en la figura 1, la ubicación actual es (juan@juan-HP-Laptop-14-ck2xxx: ~\$), el cual es el directorio personal.

2). Verificación de Ubicación

Para esta sección se utiliza el comando PWD (print working directory), este comando sirve para ver en que directorio se está ubicado.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ pwd
/home/juan
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 2: Verificación de Ubicación. Fuente: Elaboración propia.

Como se evidencia en la figura 2 se puede observar que está en el directorio personal por la ruta /home/juan.

3). Listado de Archivos y Directorios

Ahora revisaremos los directorios con el comando ls -l como se evidencia en la figura ??

```
juan@juan-HP-Laptop-14-ck2xxx:~$ ls -l
total 440
drwxr-xr-x 2 juan juan 4096 mar 10 15:36 Descargas
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Documentos
drwxr-xr-x 2 juan juan 4096 mar 3 15:28 Escritorio
drwxr-xr-x 2 juan juan 4096 mar 15 20:21 Imágenes
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Música
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Plantillas
drwxrwxr-x 3 juan juan 4096 mar 11 11:38 Practica_Linux
drwxrwxr-x 4 juan juan 4096 mar 11 11:20 Practica_Ubuntu
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Público
-rw-rw-r-- 1 juan juan 770 feb 28 12:55 roscore.log
drwx----- 5 juan juan 4096 mar 5 08:18 snap
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Videos
-rw-r--r-- 1 juan juan 197408 feb 28 01:26 windows10.qcow2
-rw-r--r-- 1 juan juan 199008 mar 5 09:08 WindowsVM.qcow2
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 3: Listado de Archivos y Directorios. Fuente: Elaboración propia.

En la figura 3 se puede ver la salida del comando ejecutado.

- d al inicio: Indica que es un directorio como los son, Descargas, Documentos, Escritorio, etc.
- -rw-r--r--: Permisos de archivos como, roscore.log, Windows10.qcow2.
- Tamaño en bytes - La columna con valores como 4096 muestra el tamaño del archivo o directorio.
- Fecha de modificación - Indica cuándo fue la última vez que se modificó cada archivo o carpeta.
- Nombre del archivo o directorio - Última columna.



UNIVERSIDAD SANTO TOMÁS

PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

FACULTAD DE INGENIERÍA ELECTRÓNICA



B. Crear Directorios

1). Creación de Directorios

Se crea un directorio con el nombre "Practica_Linux", se utiliza el código mkdir para crear el directorio necesario, se realiza la verificación de la existencia del directorio como se observa en la figura 4.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ mkdir Practica_Linux
juan@juan-HP-Laptop-14-ck2xxx:~$ ls -l
total 436
drwxr-xr-x 2 juan juan 4096 mar 10 15:36 Descargas
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Documentos
drwxr-xr-x 2 juan juan 4096 mar 3 15:28 Escritorio
drwxr-xr-x 2 juan juan 4096 mar 15 20:22 Imágenes
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Música
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Plantillas
drwxrwxr-x 2 juan juan 4096 mar 15 20:34 Practica_Linux
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Público
-rw-rw-r-- 1 juan juan 770 feb 28 12:55 roscore.log
drwx----- 5 juan juan 4096 mar 5 08:18 snap
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Videos
-rw-r--r-- 1 juan juan 197408 feb 28 01:26 windows10.qcow2
-rw-r--r-- 1 juan juan 199008 mar 5 09:08 WindowsVM.qcow2
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 4: Creación de Directorios. Fuente: Elaboración propia.

Como se observo en la figura 4, se creo exitosamente el directorio que se esta requiriendo.

2). Estructura de Subdirectorios

Ahora crearemos mas directorios dentro del directorio, primero utilizaremos el comando cd para ingresar al directorio Practica_Linux ya creado, el cual llamaremos subdirectorio, crearemos dos, Documentos y el otro llamdo Backup con el código mkdir, como se evidencia en la figura 5.

```
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ mkdir Documentos
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ mkdir Backup
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ ls -l
total 8
drwxrwxr-x 2 juan juan 4096 mar 15 20:36 Backup
drwxrwxr-x 2 juan juan 4096 mar 15 20:35 Documentos
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$
```

Figura 5: Estructura de Subdirectorios. Fuente: Elaboración propia.

En la figura 5, se evidencia la creación de los dos repositorios ya mencionados anteriormente.

C. Navegación entre las carpetas

1). Creación de Archivos

Dentro del directorio Documentos ingresando con el comando cd, crearemos un archivo .txt con el comando touch llamado nota.txt como se evidencia en la figura 6.

```
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ cd Documentos
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$ touch nota.txt
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$ ls -l
total 0
-rw-rw-r-- 1 juan juan 0 mar 15 20:40 nota.txt
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$
```

Figura 6: Creación de Archivos. Fuente: Elaboración propia.

En la figura 6 se puede evidenciar la creación del archivo con el comando ls -l.

2). Edición de Archivos

Para la edición del archivo creado se utiliza el comando nano nota.txt y se evidencia en la figura 7 la interfaz de nano.

Figura 7: Edición de Archivos. Fuente: Elaboración propia.

En la interfaz se escribió el mensaje (Hola, este es mi primer archivo en Linux) como se observa en la figura 7, luego guardamos y confirmamos para que el mensaje quede guardado en el archivo.

3). Copiar Archivos

Copiaremos el archivo nota.txt del directorio Documentos, al directorio Backup con el comando cp, como se evidencia en la figura 8

```
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Backup$ cp ~/Practica_Linux/Documentos/nota.txt ~/Practica_Linux/Backup/
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Backup$ cd
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ cd Documentos
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ cd Backup
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Backup$ ls -l
total 4
-rw-rw-r-- 1 juan juan 41 mar 15 20:40 nota.txt
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Backup$
```

Figura 8: Copiar Archivos. Fuente: Elaboración propia.

Verificamos si la copia fue exitosa utilizando el código ls -l dentro del directorio Backup, como se observa en la figura 8.

4). Renombrar Archivos

Para este paso, se utilizo el comando mv para cambiar el nombre del archivo nota.txt a nota_final.txt.

```
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ mv ~/Practica_Linux/Documentos/nota.txt ~/Practica_Linux/Documentos/nota_final.txt
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ cd Documentos
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$ ls -l
total 4
-rw-rw-r-- 1 juan juan 41 mar 15 20:42 nota_final.txt
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$
```

Figura 9: Renombrar Archivos. Fuente: Elaboración propia.



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
FACULTAD DE INGENIERÍA ELECTRÓNICA



Con el comando `cd`, se pasa al directorio Documentos para verificar con `ls -l`, el cambio del nombre del archivo como se puede observar en la figura 9.

D. Eliminar y ver contenido

1). Eliminar Directorios Vacíos

Para eliminar un directorio, se utilizara el comando `rmdir`, en este caso se intenta eliminar el directorio Backup, como se observa en la figura 10.

```
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ rmdir Backup
rmdir: fallo al borrar 'Backup': El directorio no está vacío
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$
```

Figura 10: Eliminar Directorios Vacíos. Fuente: Elaboración propia.

Como se puede evidencia en la figura 10, el comando `rmdir` falla debido a que solo funciona si el directorio esta vacío.

2). Eliminar Directorios con Contenido

Ahora para eliminar el directorio Backup el cual no esta vacío. Se utiliza el comando `rm -r`, para eliminar el directorio.

```
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ rm -r Backup
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ ls -l
total 4
drwxrwxr-x 2 juan juan 4096 mar 15 20:56 Documentos
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$
```

Figura 11: Eliminar Directorios con Contenido. Fuente: Elaboración propia.

Como se puede evidenciar en la figura 11, usando el comando `ls -l`, el directorio Backup se borro con éxito.

3). Mostrar Contenido de Archivos

Se utiliza el comando `cd` para ingresar al directorio Documentos, para poder observar la información dentro del archivo `nota_final.txt`.

```
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux$ cd Documentos
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$ cat nota_final.txt
Hola, este es mi primer archivo en Linux
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$
```

Figura 12: Mostrar Contenido de Archivos. Fuente: Elaboración propia.

Se uso el comando `cat` para observar la información del archivo como se puede observar en la figura 12.

E. Permisos

1). Gestión de Permisos

Para esta sección se utilizo el comando `chmod` para cambiar el permiso del archivo `nota_final.txt`, lo que indica la figura 13, solo el usuario se le permite leer y escribir.

```
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$ chmod 600 nota_final.txt
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$ ls -l
total 4
-rw----- 1 juan juan 41 mar 15 20:42 nota_final.txt
juan@juan-HP-Laptop-14-ck2xxx:~/Practica_Linux/Documentos$
```

Figura 13: Gestión de Permisos. Fuente: Elaboración propia.

Se puede evidenciar con éxito el cambio de los permisos, debido a que el apartado `-rw` y el `1` significa que solo el usuario tiene permisos de edición en el archivo.

F. Buscar y Filtrar

1). Búsqueda de Archivos desde el directorio personal

Desde el directorio personal utilizando el comando `find` `=name "nota_final.txt"`, para buscar un archivo utilizamos el código `find`, el símbolo `~` significa `/home/juan`, y la última sección busca el nombre del archivo en los directorios.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ find ~ -name "nota_final.txt"
/home/juan/Practica_Linux/Documentos/nota_final.txt
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 14: Búsqueda de Archivos desde el directorio personal. Fuente: Elaboración propia.

La figura 14, muestra el éxito de la búsqueda del archivo dentro del directorio personal.

2). Búsqueda en Archivos por medio de una palabra

Ahora con el comando `grep`, se buscara el archivo por medio de una palabra clave, en este caso la palabra `Linux`.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ grep "Linux" Practica_Linux/Documentos/*
Hola, este es mi primer archivo en Linux
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 15: Búsqueda en Archivos por medio de una palabra. Fuente: Elaboración propia.

Como se observa en la figura 15, la búsqueda por medio de palabra clave se realizó.

G. Procesos

1). Monitoreo de Procesos

La figura 16, muestra como se ejecuta el comando `top` el cual monitorea los procesos en tiempo real.



UNIVERSIDAD SANTO TOMÁS

PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

FACULTAD DE INGENIERÍA ELECTRÓNICA

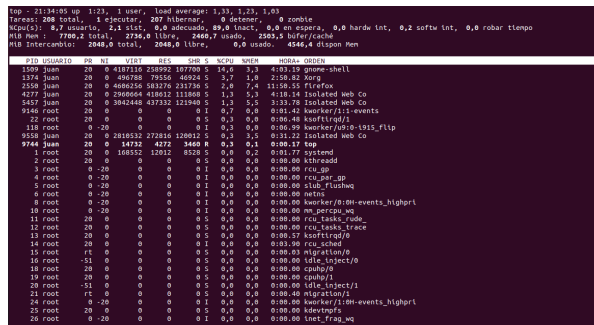


Figura 16: Monitoreo de Procesos. Fuente: Elaboración propia.

- La primera línea indica la hora del sistema, el tiempo de actividad, el número de usuarios conectados y la carga promedio.
- La segunda línea muestra el estado de las tareas del sistema: procesos en ejecución, suspendidos, detenidos y zombis.
- La tercera línea detalla el uso de la CPU, desglosando el porcentaje usado por el usuario (us), el sistema (sy), los procesos en espera (wa), etc.
- La cuarta línea muestra el uso de la memoria RAM y el espacio de intercambio (swap).
- La tabla inferior lista los procesos activos con información clave como:

- PID: Identificador del proceso.
- Usuario: Usuario que ejecuta el proceso.
- PRI/Nice: Prioridad del proceso.
- VIRT/RES: Memoria virtual y residente usada.
- CPU %/MEM %: Uso de CPU y memoria.
- Comando: Programa o proceso en ejecución.

2). Ejecución de Procesos en Segundo Plano

Para ejecutar en segundo plano se utilizó el comando `sleep 300 &`, el símbolo `&` indica que se ejecute en segundo plano

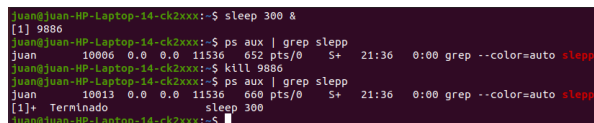


Figura 17: Ejecución de Procesos en Segundo Plano. Fuente: Elaboración propia.

En la figura 17, indica la verificación de la actividad de

segundo plano, y con el comando `kill`, se finaliza esta tarea en segundo plano.

H. Paquetes

1). Actualización de Paquetes

Para este apartado se utiliza el comando `sudo apt update` para actualizar la lista de paquetes disponibles.

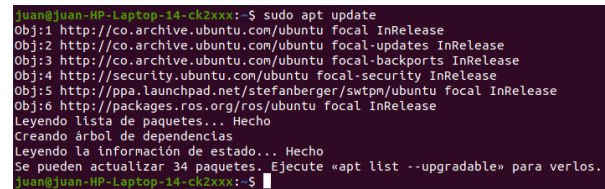


Figura 18: Actualización de Paquetes. Fuente: Elaboración propia.

En la figura 18 se puede observar como se actualiza los listados de los paquetes.

2). Instalación de Software

Ahora se instalará el programa `cowsay` (El cual es un ejemplo), con el código `sudo apt install cowsay` (Comando para instalar programas en Ubuntu).

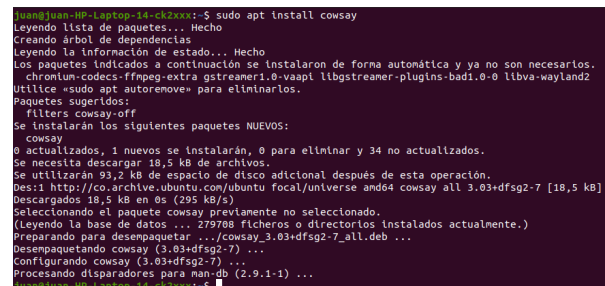


Figura 19: Instalación de Software. Fuente: Elaboración propia.

La figura 19, evidencia la correcta instalación del programa `cowsay`.

I. Parte Final

1). Creación de un Directorio de Registros

Como primer paso se eliminaron los directorios creados, y se crea un directorio llamado `Logs`.



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
FACULTAD DE INGENIERÍA ELECTRÓNICA



```
juan@juan-HP-Laptop-14-ck2xxx:~$ mkdir logs
juan@juan-HP-Laptop-14-ck2xxx:~$ ls -l
total 436
drwxr-xr-x 2 juan juan 4096 mar 10 15:36 Descargas
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Documentos
drwxr-xr-x 2 juan juan 4096 mar 3 15:28 Escritorio
drwxr-xr-x 2 juan juan 4096 mar 15 21:46 Imágenes
drwxrwxr-x 2 juan juan 4096 mar 15 21:50 logs
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Música
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Plantillas
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Público
-rw-rw-r-- 1 juan juan 770 feb 28 12:55 roscore.log
drwx----- 5 juan juan 4096 mar 5 08:18 snap
drwxr-xr-x 2 juan juan 4096 feb 28 00:14 Videos
-rw-r--r-- 1 juan juan 197408 feb 28 01:26 windows10.qcow2
-rw-r--r-- 1 juan juan 199008 mar 5 09:08 WindowsVM.qcow2
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 20: Creación de un Directorio de Registros. Fuente: Elaboración propia.

Se observa en la figura 20, la correcta creación del directorio con el comando `ls -l`.

2). Registro de Fechas

Con el comando `date`, crearemos un archivo de texto con la información de la fecha y hora actual, como se puede evidenciar en la figura 21.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ date > Logs/fecha.txt
juan@juan-HP-Laptop-14-ck2xxx:~$ cat Logs/fecha.txt
sáb 15 mar 2025 21:57:34 -05
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 21: Registro de Fechas. Fuente: Elaboración propia.

En el comando `cat` se puede evidenciar la correcta implementación del código `date` observando la fecha y hora actual al momento de ejecutar el código, la cual se observa en la figura 21.

3). Mensaje de Confirmación

Utilizando el programa descargado anteriormente Cowsay, se puede implementar el código observado en la figura 22.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ cowsay "Ejercicio completado!"
< Ejercicio completado! >
-----
      \   ^__^
       (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 22: Mensaje de Confirmación. Fuente: Elaboración propia.

Se observa el mensaje elegido (Ejercicio completado!) y un diseño en formato ASCII.

4). Ejecución de Script

Como primer paso se crea el archivo Script con `nano mis_comandos.sh`, dentro de `nano` se implementa el comando

(`cosway` Ejercicio completado!), y por ultimo ejecutamos el comando `bash mis_comandos.sh` y observamos la salida en la figura 22.

```
juan@juan-HP-Laptop-14-ck2xxx:~$ nano mis_comandos.sh
juan@juan-HP-Laptop-14-ck2xxx:~$ bash mis_comandos.sh
< Ejercicio completado! >
-----
      \   ^__^
       (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
juan@juan-HP-Laptop-14-ck2xxx:~$
```

Figura 23: Ejecución de Script. Fuente: Elaboración propia.

IV CONCLUSIONES

Resume los hallazgos principales del estudio, responde a los objetivos planteados y sugiere posibles mejoras o investigaciones futuras.

REFERENCIAS

- [1] A. S. Tanenbaum, *Modern Operating Systems*, 3rd ed. Pearson, 2008.
- [2] E. Nemeth, G. Snyder, T. R. Hein, *Linux Administration Handbook*, 2nd ed. Addison-Wesley, 2017.
- [3] M. G. Sobell, *A Practical Guide to Linux Commands, Editors, and Shell Programming*, 4th ed. Prentice Hall, 2013.
- [4] B. Ward, *How Linux Works: What Every Superuser Should Know*, 2nd ed. No Starch Press, 2014.
- [5] R. Love, *Linux Kernel Development*, 4th ed. Addison-Wesley, 2018.
- [6] W. Shotts, *The Linux Command Line: A Complete Introduction*, 2nd ed. No Starch Press, 2019.
- [7] A. Kurniawan, *Linux Basics for Hackers*, No Starch Press, 2019.
- [8] C. Newham and B. Rosenblatt, *Learning the Bash Shell*, 3rd ed. O'Reilly Media, 2005.