



Universidad Nacional Autónoma de México

FACULTAD DE INGENIERÍA
PROTECO

GENERACIÓN:
45

PROYECTO FINAL GNU/LINUX:
SHELL 3000

INTEGRANTES:
Aragón Martínez Mitzy Yaneth
Peralta Romero José Guadalupe

Semestre 2024-1

22 de septiembre de 2023

Índice

| | |
|---|-----------|
| 1. Introducción | 2 |
| 2. Desarrollo | 4 |
| 2.1. Comando ayuda | 4 |
| 2.2. Comando para juego | 5 |
| 2.3. Comando infosis | 14 |
| 2.4. Comando fecha y hora | 16 |
| 2.5. Comando para la búsqueda de archivos | 17 |
| 2.6. Comando para los créditos | 19 |
| 2.7. Comando para reproductor mp3 | 20 |
| 3. Conclusiones | 26 |

1. Introducción

En el presente trabajo se hace uso del Sistema Operativo GNU/Linux para la generación de una terminal en la cual un usuario pueda acceder a una serie de opciones que le permitan navegar entre las funciones permitidas, así como obtener información del sistema.

GNU/Linux es un sistema operativo open source, es decir, es un código diseñado de manera que sea accesible al público que permite a cualquiera utilizar, modificar y distribuir el sistema de forma gratuita. Este sistema operativo proporciona una plataforma estable y segura para la ejecución de programas, la gestión de archivos y la flexibilidad para adaptar el sistema a necesidades específicas del usuario, es por estos motivos que se adapta a las necesidades requeridas para el desarrollo de la terminal, principalmente porque permite la ejecución de Shell Scripts.

Así pues Shell Script es un programa que permite la creación de instrucciones que posteriormente son ejecutadas por una shell, es decir, por un intérprete de comandos. La principal función de estos programas es que, mediante la combinación de una serie de comandos y secuencias de control de flujo se puedan automatizar tareas y realizar procesos más complejos de los que un solo comando puede efectuar.

Son por estas características que para el desarrollo y ejecución de este trabajo se hace uso de Shell Script en Linux para la creación de comandos que permitan el acceso a diferentes archivos, información o actividades como lo son:

- Un sistema de acceso que permita al usuario ingresar una identificación y una contraseña para el acceso a la terminal
- Una interfaz por medio de la cual el usuario a través de una línea de comandos pueda interactuar con el programa presentado. Los comandos disponibles podrán ejecutarse adecuadamente además de que se anularan las salidas forzadas del programa.

Por medio de esta interfaz el usuario será capaz de ingresar a los siguientes comandos:

- Un comando que ofrezca ayuda al usuario al proporcionar una lista de los comandos que puede seleccionar
- Un comando que proporcione la información del sistema, es decir, la memoria RAM, la arquitectura del sistema y la versión del sistema operativo
- Un comando que proporcione la fecha y hora al usuario
- Un comando que al proporcionar el nombre de un archivo y ruta en la cual está localizado pueda buscarlo y mostrar si el archivo en cuestión se encuentra en el lugar indicado.
- Un comando que despliegue un juego de gato, en el cual pueden interactuar dos jugadores para definir al ganador.
- Un comando que ofrezca una interfaz gráfica que despliegue un reproductor mp3, en el cual el usuario pueda reproducir canciones, logrando cambiar entre estas, además de observar cuál es la canción que se está reproduciendo.

- Un comando que muestre los créditos de los programadores que trabajaron en la creación de la terminal proporcionada

La creación de esta terminal tiene como propósito englobar y reforzar los conocimientos adquiridos por los desarrolladores al concluir con su preparación en el curso de Linux.

2. Desarrollo

2.1. Comando ayuda

Este comando te brinda información que te da acceso a un menú de opciones a las cuales el usuario puede ingresar. Este menú despliega los nombres específicos por medio de los cuales el usuario puede llamarlos para ejecutarlos en la terminal.

```
ubuntu@Linux@MiTerminal:~$ ayuda

-----
|   || Bienvenido a la Shell 3000 !!! |
-----

Lista de los comandos disponibles en la shell

1)ayuda:Despliega una lista de los comandos exclusivos de la shell 3000
2)infosis.sh: Este comando te da los valores de la Ram, l arquitectura del sistema y la version del SO
3)fechayhora.sh: Este comando te va a dar la fecha y la hora
4)busqueda.sh:Busca un archivo en un directorio en especifico
5)creditos.sh:Te indica los creadores de la shell 3000
6)juego.sh:Te permite ingresar a un juego de gato :3
7)mp3.sh:Te permite reproducir musiquita
ubuntu@Linux@MiTerminal:~$
```

Figura 1: Comando ayuda

Código:

```
1 echo ""
2 echo "-----"
3 echo "~|~Bienvenido a la Shell 3000 !!!|~"
4 echo "-----"
5 echo ""
6 echo "Lista de los comandos disponibles en la shell"
7 echo ""
8 echo "1)ayuda:Despliega una lista de los comandos exclusivos de la shell 3000"
9 echo "2)infosis.sh:-Este comando te da los valores de la Ram, l arquitectura del"
10 echo "3)fechayhora.sh:-Este comando te va a dar la fecha y la hora"
11 echo "4)busqueda.sh:Busca un archivo en un directorio en especifico"
12 echo "5)creditos.sh:Te indica los creadores de la shell 3000"
13 echo "6)juego.sh:Te permite ingresar a un juego de gato :3"
14 echo "7)mp3.sh:Te permite reproducir musiquita"
```

2.2. Comando para juego

Este comando te permite ingresar a una interfaz que te muestre un juego, en este caso sería el juego de gato en el cual pueden participar dos jugadores, “X” y “O”. A modo de pantalla de inicio aparecen dos posibles opciones, jugar o no jugar gato, en caso de la primera opción se desplegará lo siguiente:

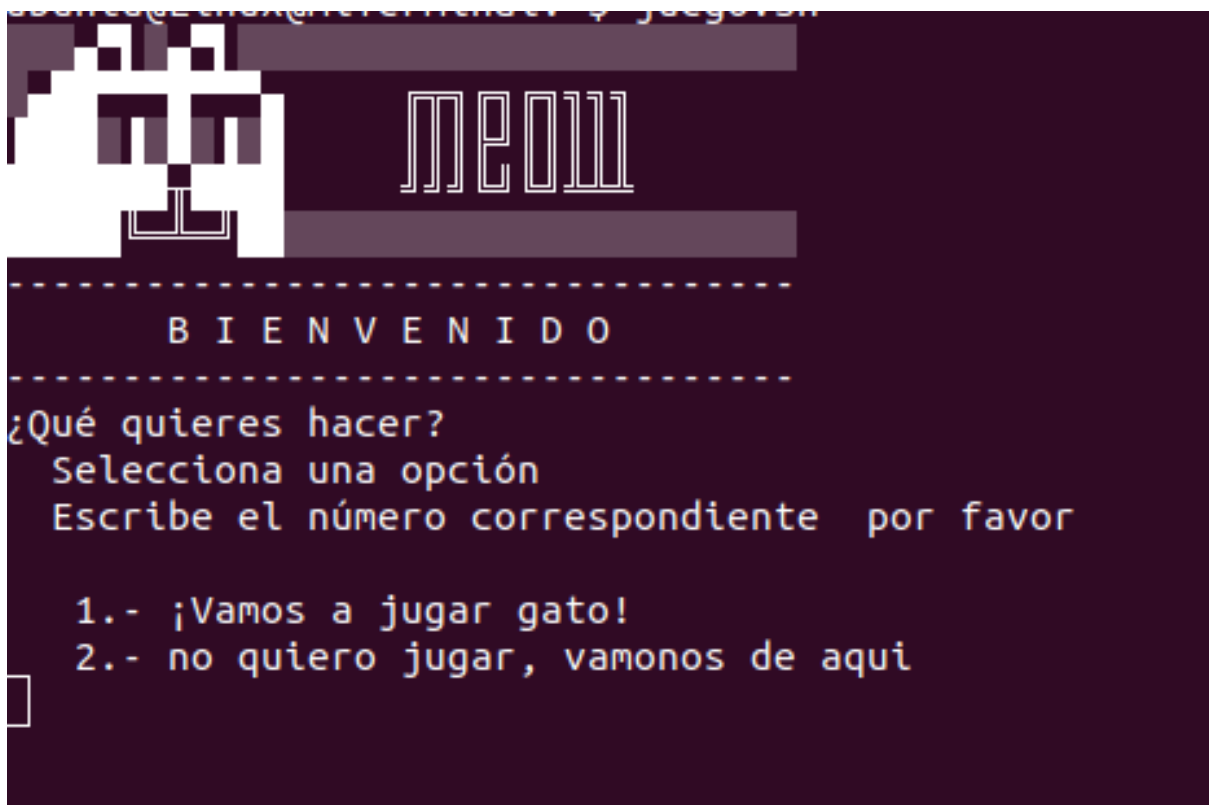


Figura 2: Captura de jueguito de gato miaw

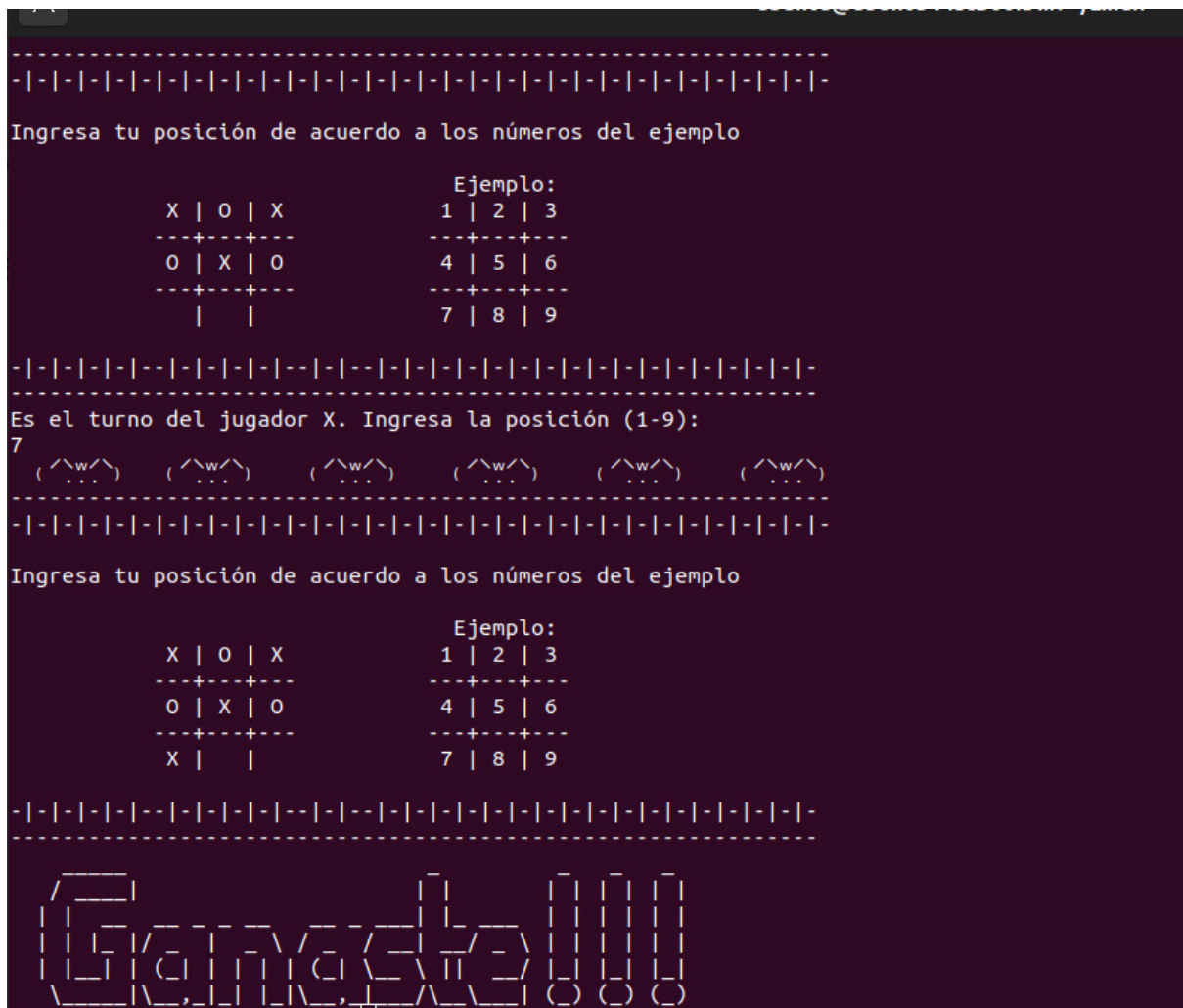


Figura 4: Captura de jugueto de gato miaw



Figura 5: Captura de jugueto de gato miaw

En caso de que el usuario seleccione la segunda opción, es decir, no jugar se desplegará el siguiente mensaje



Figura 6: Captura de jueguito de gato miaw

```

1 #!/bin/bash
2 
3 tablero=(" " " " " " " " " " " ")
4 
5 
6 mostrar_tablero() {
7 
8     echo "~~~~~"
9     echo ""
10    ↪ -----
11    ↪ ""
12    
13    echo ""
14    ↪ -|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
15    ↪ ""
16    
17    echo ""
18    echo " Ingresa tu posicin de acuerdo a los nmeros del ejemplo"
19    echo ""
20    echo " ~~~~~Ejemplo:"
21    echo " ~~~-${tablero[0]}-|-${tablero[1]}-|-${tablero[2]}~~~~~1-|2-|3-"
22    echo " ~---+--+--+--+--+-----+--+--+--"
23    echo " ~~~-${tablero[3]}-|-${tablero[4]}-|-${tablero[5]}~~~~~4-|5-|6-"
24    echo " ~---+--+--+--+--+-----+--+--+--"
25    echo " ~~~-${tablero[6]}-|-${tablero[7]}-|-${tablero[8]}~~~~~7-|8-|9-"
26    echo ""
27    echo ""
28    ↪ -|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
29    ↪ ""
30    
31    echo ""
32    ↪ -----
33    ↪ ""
34 }
35 
36 verificar_ganador() {
37     local ganador=""
38     
39     for i in 0 3 6; do
40         if [[ "${tablero[$i]}" == "${tablero[${i}+1])}" && "${tablero[$i]}" ==
            ↪ "${tablero[${i}+2])}" && "${tablero[$i]}" != "-" ]]; then
41             ganador="${tablero[$i]}"
42         fi
43     done
44 
45     for i in 0 1 2; do
46         if [[ "${tablero[$i]}" == "${tablero[${i}+3])}" && "${tablero[$i]}" ==
            ↪ "${tablero[${i}+6])}" && "${tablero[$i]}" != "-" ]]; then
47             ganador="${tablero[$i]}"
48         fi
49     done

```

```

41
42 if [[ "${tablero[0]}" == "${tablero[4]}" && "${tablero[0]}" == "${tablero
    ↪ [8]}" && "${tablero[0]}" != "-" ]]; then
43     ganador="${tablero[0]}"
44 fi
45
46 if [[ "${tablero[2]}" == "${tablero[4]}" && "${tablero[2]}" == "${tablero
    ↪ [6]}" && "${tablero[2]}" != "-" ]]; then
47     ganador="${tablero[2]}"
48 fi
49
50 echo "$ganador"
51 }
52
53 jugar_gato() {
54     local jugador="X"
55     local empate=false
56     local flag=1
57     while [ $flag == 1 ]; do
58         mostrar_tablero
59         flag=0
60         echo "Es el turno del jugador $jugador. Ingresa la posición (1-9):"
61         read posicion
62
63         if [[ ! "$posicion" =~ ^[1-9]$ ]]; then
64             echo "Por favor ingresa un número aceptable del 1 al 9"
65             continue
66         fi
67
68         local indice=$((posicion - 1))
69
70         if [[ "${tablero[$indice]}" == "-" ]]; then
71             tablero[$indice]=$jugador
72         else
73             echo "Posición ya utilizada, por favor ingresa otra:3"
74             flag=1
75             continue
76         fi
77
78         ganador=$(verificar_ganador)
79
80         for i in $(seq 0 8); do
81             if [[ "${tablero[$i]}" == "-" ]]; then
82                 flag=1
83             fi
84         done
85
86         if [[ "$ganador" == "X" || "$ganador" == "O" ]]; then
87             mostrar_tablero
88
89             echo "~~~~~"

```



```

141         ;;
142 2)
143     echo "-----"
144     echo "Que~amargado,-Yoshi-dice~adiooooo"
145     echo "-----"
146 echo ""
147 echo ""
148 echo ""
149 echo ""
150 echo ""
151 echo ""
152 echo ""
153 echo ""
154 echo ""
155 echo ""
156 echo ""
157 echo ""
158 echo ""
159 echo ""
160 echo ""
161 echo ""
162 echo ""
163 echo ""
164 echo ""
165 echo ""
166 echo ""
167 echo ""
168 echo ""
169 echo ""
170 echo ""
171 echo ""
172 echo ""
173 echo ""
174 echo ""
175 echo ""
176 echo ""
177 echo ""
178
179     exit
180     ;;
181 *)
182     echo "esa-no-es-una-opcin-valida"
183     ;;
184 esac
185
186 }
187
188 menu_principal

```

2.3. Comando infosis

En este comando el usuario puede acceder a la información del sistema en el que se encuentra, tal información está conformada por la memoria RAM, la arquitectura del sistema y la versión del sistema operativo. Para obtener esta información de la RAM se accede a la carpeta “meminfo” de la cual se extrae la información de MemTotal; para obtener la arquitectura del sistema se accede a la carpeta version en donde se busca y accede a “model name”; para obtener la versión del sistema operativo se accede a la carpeta de os-release en la cual se busca y extrae la información de “PRETTY_NAME”. Posteriormente estas informaciones extraídas se guardan en variables que se muestran al usuario.

```
ubuntu@Linux@MiTerminal:~$ infosis.sh
-----
|           I N F O S I S           |
-----
Información del sistema:
Memoria RAM: 8024080 kB
Arquitectura del sistema: x86
Versión del SO: Ubuntu 22.04.3 LTS

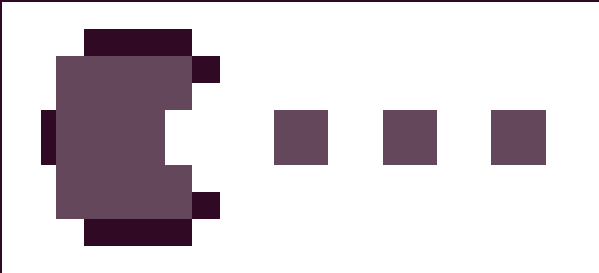
-----

-----
ubuntu@Linux@MiTerminal:~$
```

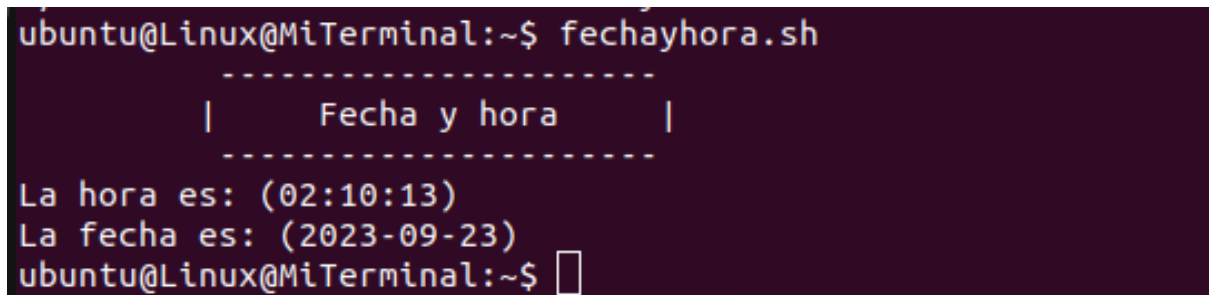
Figura 7: Captura de comando infosis

Código:

```
1 GNU nano 6.2 infosis.sh
2 # Arquitectura del Sistema desde /proc/version
3 arquitectura=$(grep 'model-name' /proc/cpuinfo | head -n 1 | awk '{print-$4}
   ↪ ')
4 if [[ "$arquitectura" == "lm" ]]; then
5     arquitectura="x86_64"
6 else
7     arquitectura="x86"
8 fi
9 # Versin del SO desde /etc/os-release
10 version_so=$(grep 'PRETTY_NAME' /etc/os-release | awk -F '=' '{print-$2}
   ↪ ' | tr -d '"')
11
12
13 echo "-----"
14 echo "|~~~~~I-N-F-O-S-I-S~~~~~|"
15 echo "-----"
16
17 echo "Informacin-del-sistema:"
18 echo "Memoria-RAM:-$memoria_ram-kB"
19 echo "Arquitectura-del-sistema:-$arquitectura"
20 echo "Versin-del-SO:-$version_so"
21
22
23 echo ""
24 echo "-----"
25 echo ""
26 echo ""
27 echo ""
28 echo ""
29 echo ""
30 echo "-----"
```

2.4. Comando fecha y hora

Este comando permite al usuario acceder a la información del sistema de fecha y hora. Para lograr esto se ingresa a la carpeta de rtc la cual gestiona el reloj en tiempo real. Usar la carpeta del componente RTC es una buena opción ya que este hardware mantiene un seguimiento del tiempo aun cuando la computadora está apagada o reiniciada. Una vez en la carpeta rtc se extrae la información de “rtc.time” y “rtc.date”, proporcionando así la hora en formato hora, minuto, segundos y la fecha en formato año, mes y día.



```
ubuntu@Linux@MiTerminal:~$ fechayhora.sh
-----
|      Fecha y hora      |
-----
La hora es: (02:10:13)
La fecha es: (2023-09-23)
ubuntu@Linux@MiTerminal:~$
```

Figura 8: Captura de comando de fecha y hora

Código:

```
1 #!/bin/bash
2
3 echo "-----"
4 echo "-----|-----Fecha-y-hora-----|"
5 echo "-----"
6
7
8
9 hora=$(cat /proc/driver/rtc | grep "rtc.time" | sed 's/rtc.time[-\t]*:[-\t]*//')
10 echo "La-hora-es:($hora)"
11
12 fecha=$(cat /proc/driver/rtc | grep "rtc.date" | sed 's/rtc.date[-\t]*:[-\t]*//')
13
14 echo "La-fecha-es:($fecha)"
15 echo ":3"
```

2.5. Comando para la búsqueda de archivos

Este comando permite la búsqueda de un archivo para lo cual la shell recibe la ruta absoluta ingresada por el usuario así como el nombre del archivo que quiere localizar, posteriormente con “-e” procede a realizar la búsqueda y finalmente si la encuentra proyecta un mensaje informando que encontró lo solicitado, en caso contrario cambia el mensaje indicando que no se encuentra en la dirección proporcionada.

```
ubuntu@Linux@MiTerminal:~$ busqueda.sh
-----
|   ENCONTRAR ARCHIVO   |
-----

vamo a buscar tu archivo

por favor, ingresa la ruta del directorio en donde se encuentra tu carpeta
/home/ubuntu/Descargas

por favor, ingresa el nombre del archivo que quieres encontrar, recuerda especificar el tipo de archivo
totodile.jpg
ufaaaas su archivo totodile.jpg si se encuentra en /home/ubuntu/Descargas
ubuntu@Linux@MiTerminal:~$
```

Figura 9: Captura de comando para búsqueda de archivos

```
ubuntu@Linux@MiTerminal:~$ busqueda.sh
-----
|   ENCONTRAR ARCHIVO   |
-----

vamo a buscar tu archivo

por favor, ingresa la ruta del directorio en donde se encuentra tu carpeta
/home/ubuntu/Descargas

por favor, ingresa el nombre del archivo que quieres encontrar, recuerda especificar el tipo de archivo
totodile.jpg
ufaaaas su archivo totodile.jpg si se encuentra en /home/ubuntu/Descargas
ubuntu@Linux@MiTerminal:~$
```

Figura 10: Captura de comando para búsqueda de archivos

Código:

```
1 #!/bin/bash
2
3 # Un comando que pueda buscar por un archivo en un directorio específico
   ↳ recibe
4 #dos parámetros: La carpeta a buscar y el archivo que va a buscar.
5 #Ej: Linux GEN 44
6
7 echo "~~~~~"
8 echo "~~~~|~~~~ENCONTRAR-ARCHIVO~~~~|~"
9 echo "~~~~~"
10 echo ""
11 echo "vamos a buscar tu archivo"
12 echo ""
13 echo "por favor, ingresa la ruta absoluta del directorio en donde se encuentra tu
   ↳ carpeta"
14 read rutaCarpeta
15 echo ""
16 echo "por favor, ingresa el nombre del archivo que quieres encontrar, recuerda
   ↳ especificar el tipo de archivo"
17 read nombreArchivo
18
19 #rutaCarpeta="$(realpath "$rutaCarpeta")"
20 rutaCarpeta_y_nombreArchivo="$rutaCarpeta/$nombreArchivo"
21
22 if [ -e "$rutaCarpeta_y_nombreArchivo" ]; then
23     echo "ufaaaas su archivo $nombreArchivo si se encuentra en
   ↳ $rutaCarpeta"
24 else
25     echo "nooo puede ser su archivo $nombreArchivo no se ha encontrado
   ↳ en $rutaCarpeta, que triste tu caso"
26
27 fi
```

2.6. Comando para los créditos

Este comando muestra con una función simple de echo los créditos de los programadores que se rifaron semejante proyecto, más chulo de hermoso realizado con todo su corazón, sudor, ansiedad, lágrimas y pocos conocimientos al no ser de cómputo. Tal vez es poco pero es un trabajo honesto.

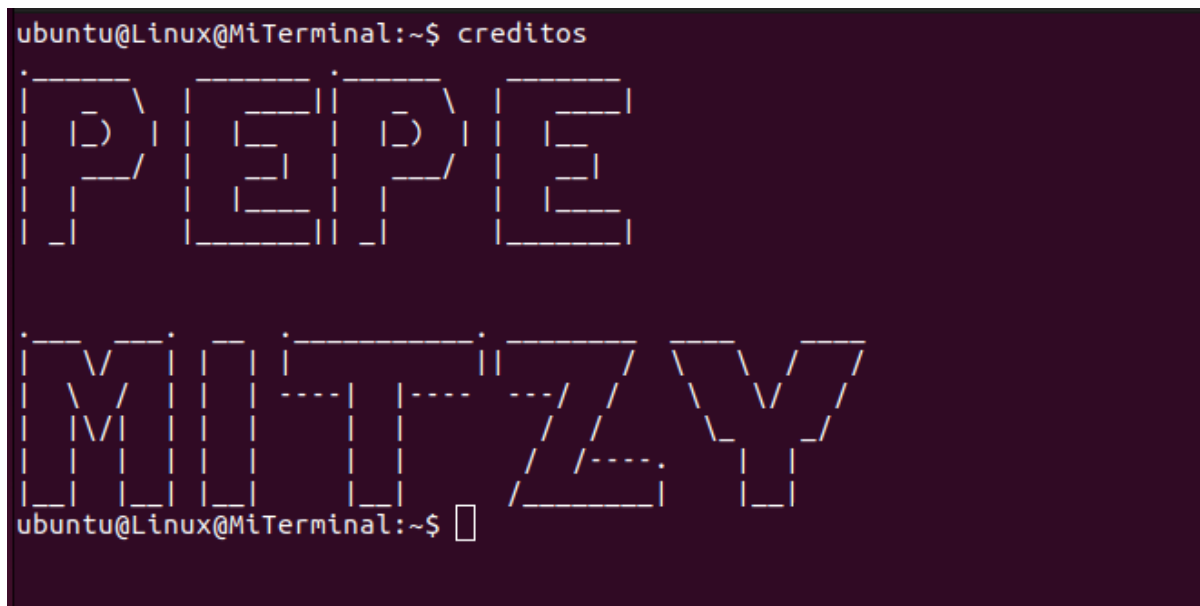


Figura 11: Captura de comando para mostrar los créditos

Código:

1 echo " .----- .----- .----- .----- .----- "

2 echo " |__ _ __ |__ _ __ |__ _ __ |__ _ __ |__ _ "

3 echo " |__ | _) |__ | _ __ |__ | _) |__ | _ __ |__ _ "

4 echo " |__ _ _ /__ |__ _ _ |__ |__ _ _ /__ |__ _ _ |__ _ "

5 echo " |__ |__ _ |__ |__ _ _ |__ |__ _ |__ |__ _ _ |__ _ "

6 echo " |__ |__ _ |__ _ _ |__ |__ _ |__ _ _ |__ _ "

7 echo ""

8 echo ""

9

10 echo " .----- .----- .----- .----- .----- .----- .----- "

11 echo " |__ __ /__ |__ |__ |__ _ |__ _ /__ __ __ __ /__ /__ "

12 echo " |__ __ /__ |__ |__ |__ _ _ |__ |__ _ _ _ /__ /__ __ __ /__ /__ "

13 echo " |__ |__ /__ |__ |__ |__ _ |__ |__ _ /__ /__ __ _ _ /__ _ "

14 echo " |__ |__ |__ |__ |__ |__ _ |__ |__ _ /__ /__ _ _ . |__ |__ |__ _ "

15 echo " |__ |__ |__ |__ |__ |__ _ |__ |__ _ /__ _ _ _ |__ |__ |__ _ "

2.7. Comando para reproductor mp3

Al ingresar a este comando al usuario se le desplegarán dos menús, el primero da información de las opciones que tiene de reproducción de música, de las canciones que puede reproducir así como de la opción de salida. En el segundo menú se muestran las opciones que puede escoger el usuario para mejorar y adaptar su experiencia al reproducir la música, es decir, pausar la canción , reproducir la siguiente o anterior rola, subir o bajar volumen, regresar la canción al principio y salir del reproductor. Posteriormente una vez el usuario lee las opciones puede ingresar las opciones que eligió para que la musiquita empiece a sonar y diga subele a la radio. Este comando se basa en la creación de una segunda terminal en la cual se redimensiona al usuario a la ruta seleccionada para el acceso a las canciones disponibles..

```

mpg123 esta instalado en este sistema.
Ingresa la ruta de la carpeta en donde almacenas tu musica:
Ruta:
/home/pepito/Desktop/musica

*****Reproductor De Musica PYM*****
*****
                      MENU PRINCIPAL                      **
*****
***** 1)Reproducir una cancion                      **
*****
***** 2)Reproducir todas las canciones              **
*****
***** 3)Reproducir todas las canciones en aleator
io *****
***** 4)Ver las canciones de la carpeta              **
*****
*****+* 5)Salir                                      **
*****
*****
*****
*****
*****
*****
*****
                      MENU DEL REPRODUCTOR                      ****
*****
----- [s] --> Detener/Continuar reproduciendo -----
-----
----- [d] --> Canción anterior -----
-----
----- [f] --> Canción siguiente -----
-----
----- [b] --> Regresar la canción al principio ---
-----
----- [+] --> Subir volumen -----
-----
----- [-] --> Bajar volumen -----
-----
----- [q] --> Salir del reproductor -----
-----

Estas opciones son solo validas si se eligen las opciones
1,2 o 3 del menu principal

Elige una opcion del menu principal:
1
¿Qué canción quieres escuchar?:
cancion1
Ahora puedes elegir una opcion del menu del reproductor
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2, an

```

Figura 12: Captura de comando para reproduccion de mp3

Código:

```
1 # Verifica si mpg123 est instalado
2 if command -v mpg123 &>/dev/null; then
3     echo "mpg123-est-instalado-en-este-sistema."
4 else
5     echo "mpg123-no-est-instalado-en-este-sistema.Porfavor-utilice-en-
    ↪ su-terminal-sudo-apt—install-mpg123-si-desea-utilizar-este-
    ↪ super-reproductor"
6     exit
7 fi
8
9
10
11
12
13
14 control=0
15 #Ruta de la carpeta en donde se encuentra la musica
16 echo -e "\e[0;32mIngresa-la-ruta-de-la-carpeta-en-donde-almacenas-tu-
    ↪ musica:\e[0m"
17 echo -e "\e[1;33mRuta\e[0m:"
18 read ruta
19 cd $ruta
20
21
22 #Empieza el ciclo para mostrar el menu y hacer las operaciones de la
    ↪ prebeplayer
23 while [ $control -ne 1 ];
24 do
25     #Menu de la prebeplayer
26     echo -e "\n\t\e[1;33m*****\e[0m\e[1;31
    ↪ mReproductor-De-Musica-PYM\e[0m\e[1;33m
    ↪ *****\e[0m"
27     echo -e "\t\e[1;33m*****\e[0m~~~~~\e[1;33
    ↪ mMENU-PRINCIPAL\e[0m~~~~~\e[1;33m
    ↪ *****\e[0m"
28     echo -e "\t\e[1;33m*****\e[0m-\e[1;34m1)
    ↪ Reproducir-una-cancion\e[0m~~~~~\e[1;33m
    ↪ *****\e[0m"
29     echo -e "\t\e[1;33m*****\e[0m-\e[1;34m2)
    ↪ Reproducir-todas-las-canciones\e[0m~~~~~\e[1;33m
    ↪ *****\e[0m"
30     echo -e "\t\e[1;33m*****\e[0m-\e[1;34m3)
    ↪ Reproducir-todas-las-canciones-en-aleatorio\e[0m-\e
    ↪ [1;33m*****\e[0m"
31     echo -e "\t\e[1;33m*****\e[0m-\e[1;34m4)Ver-las-
    ↪ canciones-de-la-carpeta\e[0m~~~~~\e[1;33m
    ↪ *****\e[0m"
32     echo -e "\t\e[1;33m*****+*\e[0m-\e[1;34m5)Salir\e
    ↪ [0m~~~~~\e[1;33m*****\e[0m"
33     echo -e "\t\e[1;33m
```

```

34      ↪ *****\
      ↪ e[0m"
echo -e "\n\t\e[1;31m*****\e[0m~~~~~\e[1;34
      ↪ mMENU-DEL-REPRODUCTOR\e[0m~~~~~\e[1;31m
      ↪ *****\e[0m"
35 echo -e "\t\e[1;31m-----\e[0m-\e[1;33m[s]-
      ↪ -->-Detener/Continuar-reproduciendo\e[0m-\e[1;31
      ↪ m-----\e[0m"
36 echo -e "\t\e[1;31m-----\e[0m-\e[1;33m[d]
      ↪ --->-Cancin-anterior\e[0m-\e[1;31m
      ↪ -----\
      ↪ e[0m"
37 echo -e "\t\e[1;31m-----\e[0m-\e[1;33m[f]-
      ↪ -->-Cancin-siguiente\e[0m-\e[1;31m
      ↪ -----\e
      ↪ [0m"
38 echo -e "\t\e[1;31m-----\e[0m-\e[1;33m[b]
      ↪ --->-Regresar-la-cancin-al-principio\e[0m-\e[1;31m
      ↪ -----\e[0m"
39 echo -e "\t\e[1;31m-----\e[0m-\e[1;33m
      ↪ [+]->-Subir-volumen\e[0m-\e[1;31m
      ↪ -----\
      ↪ e[0m"
40 echo -e "\t\e[1;31m-----\e[0m-\e[1;33m
      ↪ [-]->-Bajar-volumen\e[0m-\e[1;31m
      ↪ -----\
      ↪ e[0m"
41 echo -e "\t\e[1;31m-----\e[0m-\e[1;33m[q]
      ↪ --->-Salir-del-reproductor\e[0m-\e[1;31m
      ↪ -----\e[0m"
42 echo -e "\t\e[1;36mEstas-opciones-son-solo-validas-si-se-eligen
      ↪ -las-opciones-1,2-o-3-del-menu-principal\e[0m"
43
44 #Solicita la eleccion de una opcion del menu que se le va a
      ↪ mostar
45 echo -e "\n\e[0;32mElige-una-opcion-del-menu-principal:-\e[0
      ↪ m"
46 read opcion
47 #Control de las acciones para cada opcion del menu
48 case $opcion in
49     1)
50         #Le permite al usuario elegir la cancion que
          ↪ va a escuchar
51         echo -e "\e[1;35mQu-cancin-quieres-
          ↪ escuchar?:-\e[0m"
52         read cancion
53         echo -e "\e[0;32mAhora-puedes-elegir-una-
          ↪ opcion-del-menu-del-reproductor\e[0m
          ↪ "
54         #Comando que se encarga de reproducir la
          ↪ cancin que se le indica en formato

```



```

55         ↪ mp3
56         mpg123 -C /$ruta/"$cancion.mp3"
57
58     ;;
59
60     2)
61         #Reproduce todas las canciones dentro de la
62         ↪ carpeta de musica que el usuario
63         ↪ indico
64         echo -e "\e[1;35mVamos-a-armar-un-
65         ↪ pachangon-con-tu-playlist\e[0m"
66         echo -e "\e[0;32mAhora-puedes-elegir-una-
67         ↪ opcion-del-menu-del-reproductor\e[0m
68         ↪ "
69         #Comando para reproducir todas las
70         ↪ canciones
71         mpg123 -C /$ruta/*
72
73     ;;
74
75     3)
76         #Reproduce todas las canciones dentro de la
77         ↪ carpeta de musica que el usuario le
78         ↪ indico
79         echo -e "\e[1;35mVamos-a-armar-un-
80         ↪ pachangon-aleatorio-con-tu-playlist\e
81         ↪ [0m"
82         echo -e "\e[0;32mAhora-puedes-elegir-una-
83         ↪ opcion-del-menu-del-reproductor\e[0m
84         ↪ "
85         #Comando para reproducir en modo
86         ↪ aleatorio
87         mpg123 -C -z /$ruta/*
88
89     ;;
90
91     4)
92         echo "Las-canciones-son:"
93         ls $ruta
94
95     ;;
96
97     5)
98         control=1
99
100    ;;
101
102    *)
103        #Indica que hubo error si el usuario bobo no
104        ↪ elige algo del menu
105        echo -e "\e[1;35mOpcin-no-vlida,-intentalo-
106        ↪ de-nuevo-plis\e[0m"

```

```
90             ;;
91
92
93         esac
94     done
```

3. Conclusiones

- **Aragón Martínez Mitzy Yaneth:** La realización de este proyecto de una terminal en Linux ha sido un ejercicio valioso que ha proporcionado una comprensión más profunda de este sistema operativo y de cómo interactuamos con él a través de línea de comandos. A lo largo de este proyecto, pudimos mejorar nuestras habilidades de programación, especialmente en Shell Script, así como en la manipulación de archivos, la gestión de procesos, la interpretación de comandos, la gestión de recursos y de seguridad y permisos. Por último, pero no menos importante, desarrollamos nuestra capacidad de trabajo en equipo, organización, y comunicación, dado que trabajamos en conjunto y además utilizamos las herramientas de Git y Github para el manejo del control de versiones.
- **Peralta Romero José Guadalupe:** La ejecución de este proyecto de crear una terminal simulada en el entorno Linux se ha revelado como una experiencia extremadamente beneficiosa que nos ha brindado una comprensión más profunda de cómo opera este sistema operativo y de nuestra interacción con él a través de la línea de comandos. Durante el desarrollo de este proyecto, no solo tuvimos la oportunidad de perfeccionar nuestras habilidades de programación, centrándonos en Shell Script, sino que también adquirimos un dominio más sólido en áreas esenciales como la manipulación de archivos, la administración de procesos, la interpretación de comandos, la gestión de recursos y la implementación de medidas de seguridad y permisos. No obstante, el valor de este proyecto trascendió la mera mejora de habilidades técnicas. Se convirtió en una experiencia de desarrollo personal y de equipo. Trabajar juntos en la creación de esta terminal simulada nos permitió fortalecer nuestras habilidades de trabajo en equipo, organización y comunicación. Fue un terreno fértil para la colaboración, donde aprendimos a enfrentar desafíos técnicos y a resolver problemas de manera conjunta. Además, no podemos pasar por alto la introducción al mundo de Git y Github. Utilizar estas herramientas para el control de versiones fue un paso importante en nuestro proceso de aprendizaje y colaboración. Nos enseñó cómo llevar un seguimiento eficiente de nuestros cambios, mantener la coherencia en el proyecto y facilitar la contribución de cada miembro del equipo. En resumen, este proyecto no solo nos brindó conocimientos técnicos valiosos, sino que también enriqueció nuestras habilidades de colaboración y gestión de proyectos. Nos prepara para abordar futuros desafíos en el vasto campo de la informática con confianza y competencia.