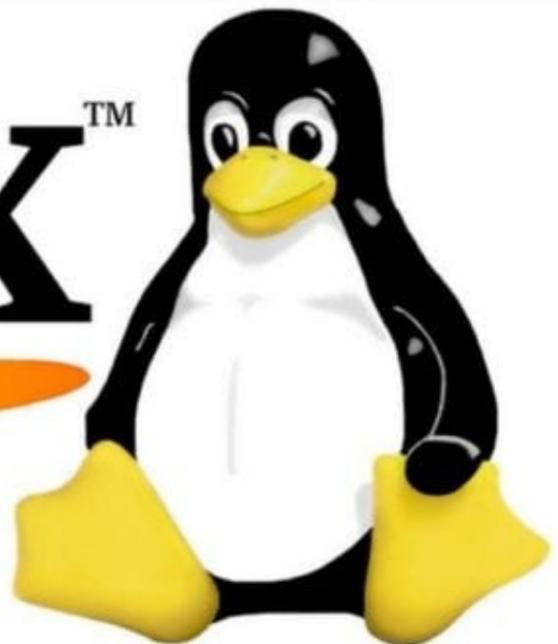


M 0 6 S C R I P T I N G

Linux™



Shell Scripting

ALFONSO DOMÍNGUEZ
GARCÍA | SIS1

ÍNDICE

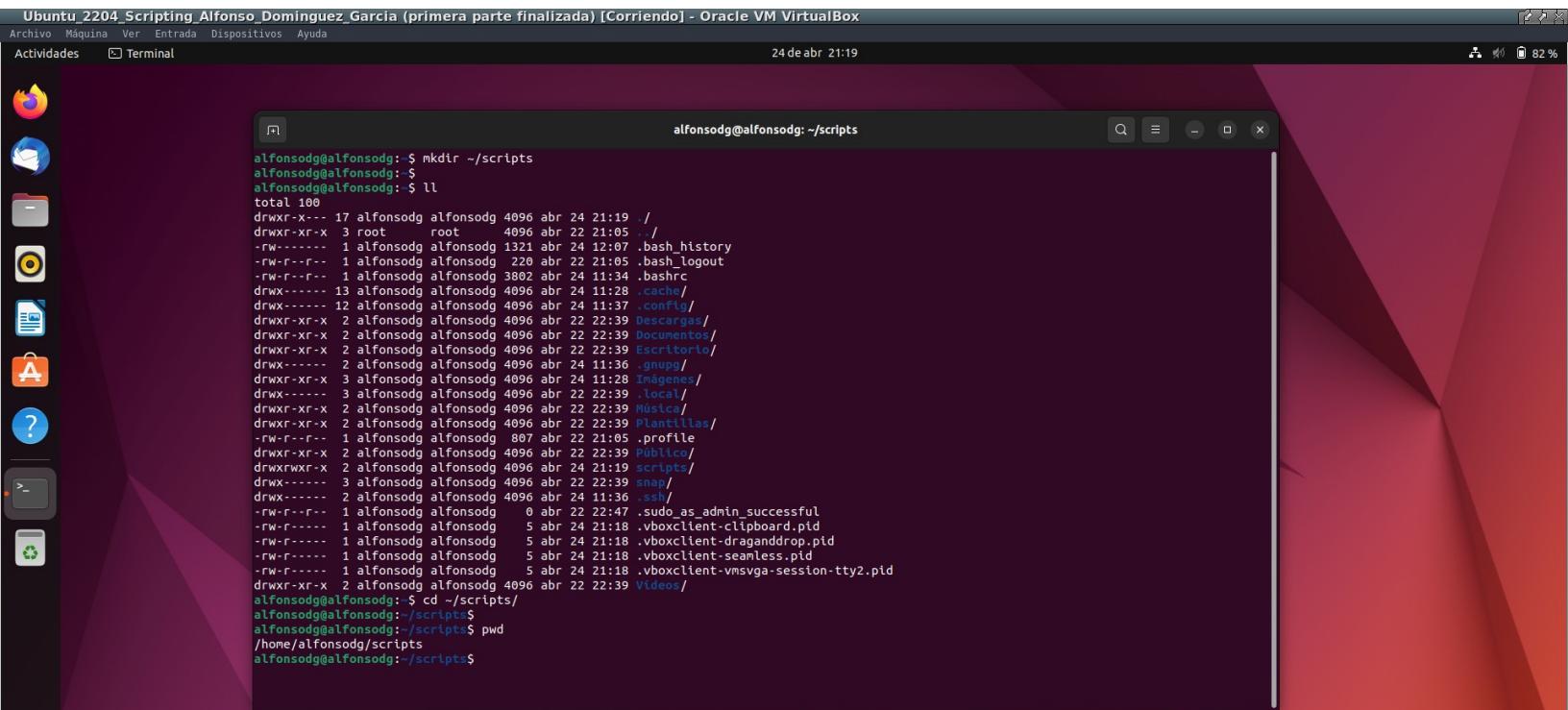
ACT. 1.....	3
ACT. 2.....	8
ACT. 3.....	11
ACT. 4.....	12
ACT. 5.....	16
ACT. 6.....	19
ACT. 6 EXTRA.....	25
ACT. 7.....	28
ACT. 8.....	30
ACT. 9.....	34
ACT. 10.....	43
ACT. 11.....	49
ACT. 12.....	60
ACT. 13.....	62
ACT. 14.....	63
ACT. 14 EXTRA.....	66
Webgrafía.....	71

ACT. 1

- a) Primero crearemos el directorio ejecutando el siguiente comando

mkdir ~/scripts

En el propio home del usuario, en mi caso → **/home/alfonsodg/scripts**



The screenshot shows a terminal window titled "Terminal" with the command "alfonsodg@alfonsodg: ~/scripts". The user runs "mkdir ~/scripts" and then lists the contents of the new directory with "ll". The output shows a large number of files and subdirectories, mostly hidden (starting with .), including ".bash_logout", ".bashrc", ".cache", ".config", "Descargas", "Documentos", "Escritorio", ".gnupg", "Imágenes", ".local", ".Music", ".Plantillas", ".profile", "Público", "scripts", "snap", ".ssh", ".sudo_as_admin_successful", ".vboxclient-clipboard.pid", ".vboxclient-draganddrop.pid", ".vboxclient-seamless.pid", ".vboxclient-vmsvga-session-tty2.pid", and "Videos". The terminal window is part of a desktop environment with a dark theme.

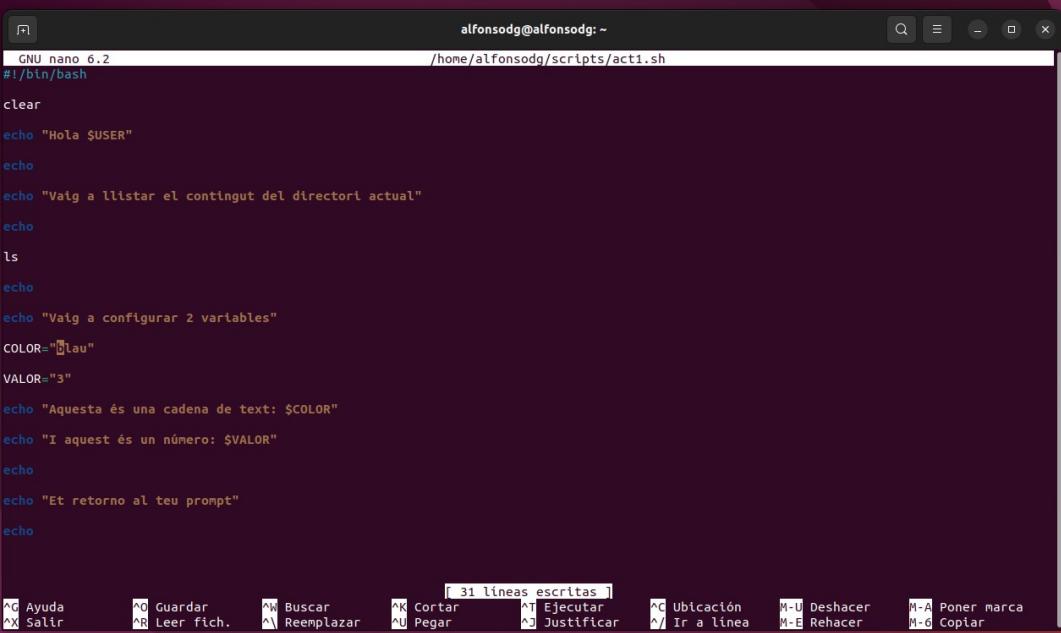
```
alfonsodg@alfonsodg: ~$ mkdir ~/scripts
alfonsodg@alfonsodg: ~$ ll
total 100
drwxr-x--- 17 alfonsodg alfonsodg 4096 abr 24 21:19 /
drwxr-xr-x  3 root    root    4096 abr 22 21:05 ../
-rw-r--r--  1 alfonsodg alfonsodg 1321 abr 24 12:07 .bash_history
-rw-r--r--  1 alfonsodg alfonsodg 220 abr 22 21:05 .bash_logout
-rw-r--r--  1 alfonsodg alfonsodg 3802 abr 24 11:34 .bashrc
drwx----- 13 alfonsodg alfonsodg 4096 abr 24 11:28 .cache/
drwx----- 12 alfonsodg alfonsodg 4096 abr 24 11:37 .config/
drwxr-xr-x  2 alfonsodg alfonsodg 4096 abr 22 22:39 Descargas/
drwxr-xr-x  2 alfonsodg alfonsodg 4096 abr 22 22:39 Documentos/
drwxr-xr-x  2 alfonsodg alfonsodg 4096 abr 22 22:39 Escritorio/
drwx----- 2 alfonsodg alfonsodg 4096 abr 24 11:36 .gnupg/
drwxr-xr-x  3 alfonsodg alfonsodg 4096 abr 24 11:28 Imágenes/
drwx----- 3 alfonsodg alfonsodg 4096 abr 22 22:39 .local/
drwxr-xr-x  2 alfonsodg alfonsodg 4096 abr 22 22:39 .Music/
drwxr-xr-x  2 alfonsodg alfonsodg 4096 abr 22 22:39 .Plantillas/
-rw-r--r--  1 alfonsodg alfonsodg 807 abr 22 21:05 .profile
drwxr-xr-x  2 alfonsodg alfonsodg 4096 abr 22 22:39 Público/
drwxrwxr-x  2 alfonsodg alfonsodg 4096 abr 24 21:19 scripts/
drwx----- 3 alfonsodg alfonsodg 4096 abr 22 22:39 snap/
drwx----- 2 alfonsodg alfonsodg 4096 abr 24 11:36 .ssh/
drwx----- 1 alfonsodg alfonsodg 0 abr 22 22:47 .sudo_as_admin_successful
-rw-r----- 1 alfonsodg alfonsodg 5 abr 24 21:18 .vboxclient-clipboard.pid
-rw-r----- 1 alfonsodg alfonsodg 5 abr 24 21:18 .vboxclient-draganddrop.pid
-rw-r----- 1 alfonsodg alfonsodg 5 abr 24 21:18 .vboxclient-seamless.pid
-rw-r----- 1 alfonsodg alfonsodg 5 abr 24 21:18 .vboxclient-vmsvga-session-tty2.pid
drwxr-xr-x  2 alfonsodg alfonsodg 4096 abr 22 22:39 Videos/
alfonsodg@alfonsodg: ~$ cd ~/scripts/
alfonsodg@alfonsodg: ~/scripts$ pwd
/home/alfonsodg/scripts
alfonsodg@alfonsodg: ~/scripts$
```

-El siguiente paso será crear el fichero "act1.sh" dentro de mi directorio ~/scripts y le daré permisos de ejecución.

-Para ello ejecutamos el siguiente comando:

touch ~/scripts/act1.sh

-Muestro captura de pantalla del propio script **act1.sh** creado:



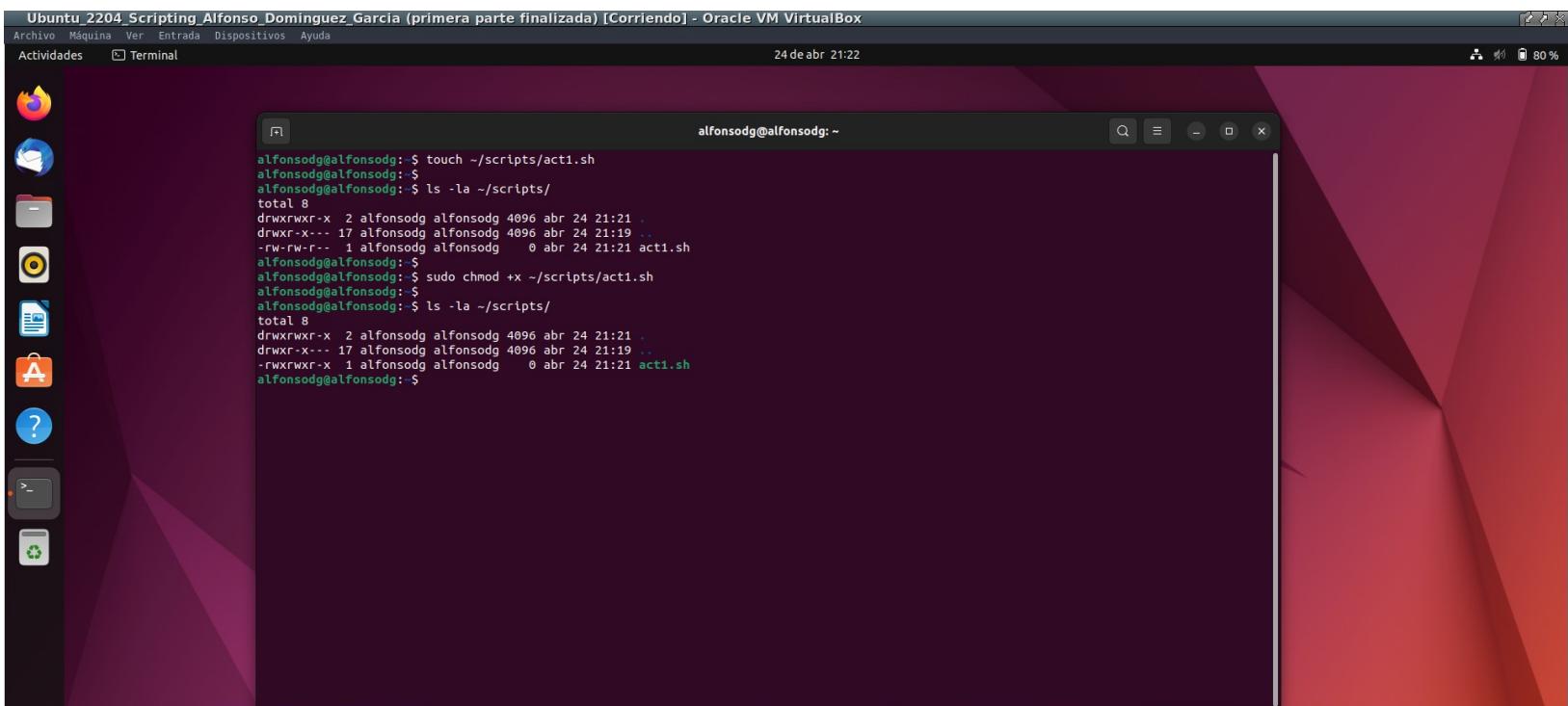
The screenshot shows a terminal window titled "GNU nano 6.2" with the command "/home/alfonsodg/scripts/act1.sh". The script content is as follows:

```
#!/bin/bash
clear
echo "Hola $USER"
echo
echo "Vaig a llistar el contingut del directori actual"
echo
ls
echo
echo "Vaig a configurar 2 variables"
COLOR="\033[34m"
VALOR="3"
echo "Aquesta és una cadena de text: $COLOR"
echo "I aquest és un número: $VALOR"
echo
echo "Et retorno al teu prompt"
echo
```

The terminal window has a dark background and light-colored text. The bottom status bar shows keyboard shortcuts for various functions like Ayuda (Help), Guardar (Save), and Ejecutar (Execute). The title bar indicates the session is running on "Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (primera parte finalizada) [Corriendo] - Oracle VM VirtualBox".

-Podemos dar permisos de ejecución ejecutando el comando:

sudo chmod +x ~/scripts/act1.sh



A screenshot of a Linux desktop environment, specifically Ubuntu 22.04, running in Oracle VM VirtualBox. The desktop has a dark purple and red geometric background. On the left is a vertical dock with icons for various applications like a browser, file manager, and terminal. A terminal window is open in the center, titled 'alfonsodg@alfonsodg: ~'. The terminal shows the following command-line session:

```
alfonsodg@alfonsodg: $ touch ~/scripts/act1.sh
alfonsodg@alfonsodg: $ ls -la ~/scripts/
total 8
drwxrwxr-x  2 alfonsodg alfonsodg 4096 abr 24 21:21 .
drwxr-x--- 17 alfonsodg alfonsodg 4096 abr 24 21:19 ..
-rw-rw-r--  1 alfonsodg alfonsodg    0 abr 24 21:21 act1.sh
alfonsodg@alfonsodg: $ sudo chmod +x ~/scripts/act1.sh
alfonsodg@alfonsodg: $ ls -la ~/scripts/
total 8
drwxrwxr-x  2 alfonsodg alfonsodg 4096 abr 24 21:21 .
drwxr-x--- 17 alfonsodg alfonsodg 4096 abr 24 21:19 ..
-rwxrwxr-x  1 alfonsodg alfonsodg    0 abr 24 21:21 act1.sh
alfonsodg@alfonsodg: $
```

b) El siguiente paso será ejecutar ya el fichero sin estar situado dentro del directorio de ~/scripts



A screenshot of a terminal window titled 'alfonsodg@alfonsodg: ~'. The terminal shows the following command-line session:

```
alfonsodg@alfonsodg:~$ act1.sh
act1.sh: orden no encontrada
alfonsodg@alfonsodg:~$
```

-Y para poder añadir o exportar el path deberíamos de escribir → **export PATH="\$PATH:~/scripts"**
muestro captura de pantalla a continuación.

```

GNU nano 6.2          .bashrc
alias alert='notify-send --urgency=low -t "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n 1)"'
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -q posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PATH="$PATH:~/scripts"

```

-Si ahora ejecutamos el comando act1.sh fuera del directorio scripts debería de funcionar correctamente el script y lo muestra en el propio bash.

```

alfonsodg@alfonsodg: ~
Hola alfonsodg

Vaig a llistar el contingut del directori actual

Descargas Escritorio Música Público snap
Documentos Imágenes Plantillas scripts Videos

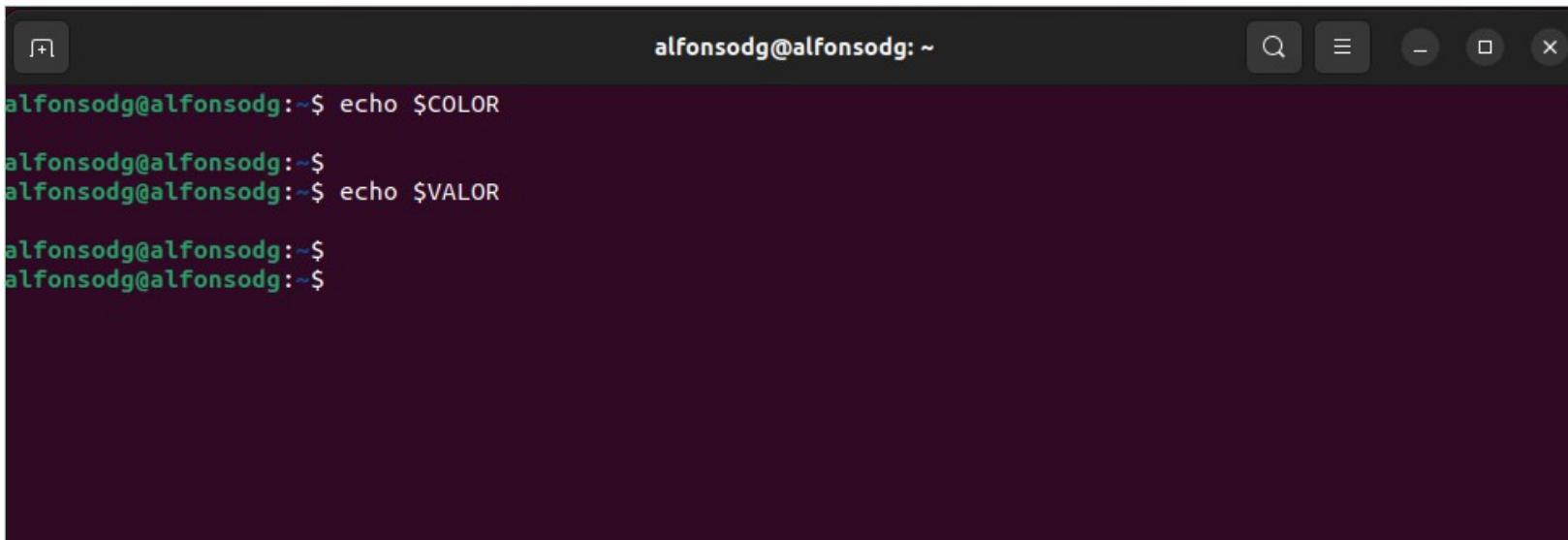
Vaig a configurar 2 variables
Aquesta és una cadena de text: blau
I aquest és un número: 3

Et retorno al teu prompt

alfonsodg@alfonsodg:~$ 

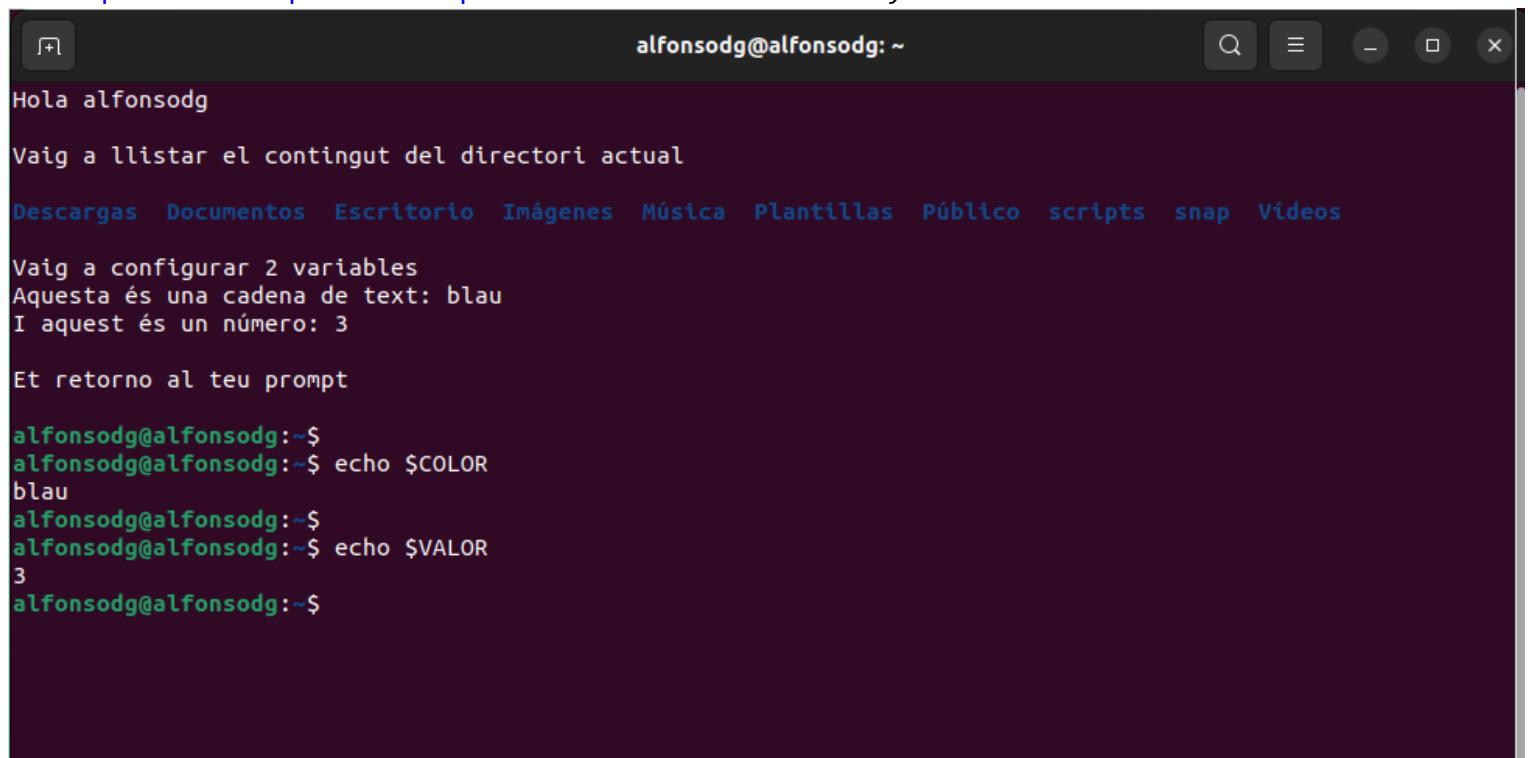
```

- c) Al ejecutar los comando `$COLOR` y `$VALOR` podemos ver que no muestran nada ya que son variables de entorno locales y NO globales. Por lo tanto se encuentran en el propio script `act1.sh` establecidas.



```
alfonsodg@alfonsodg:~$ echo $COLOR
alfonsodg@alfonsodg:~$ echo $VALOR
alfonsodg@alfonsodg:~$
```

-Para poder hacerlas globales o externas al propio script lo que hay que hacer es ejecutar el comando `source act1.sh` y si volvemos a ejecutar las variables que antes eran locales `$COLOR` y `$VALOR` podemos ver que ahora sí que devolverán los valores “blau” y “3”.



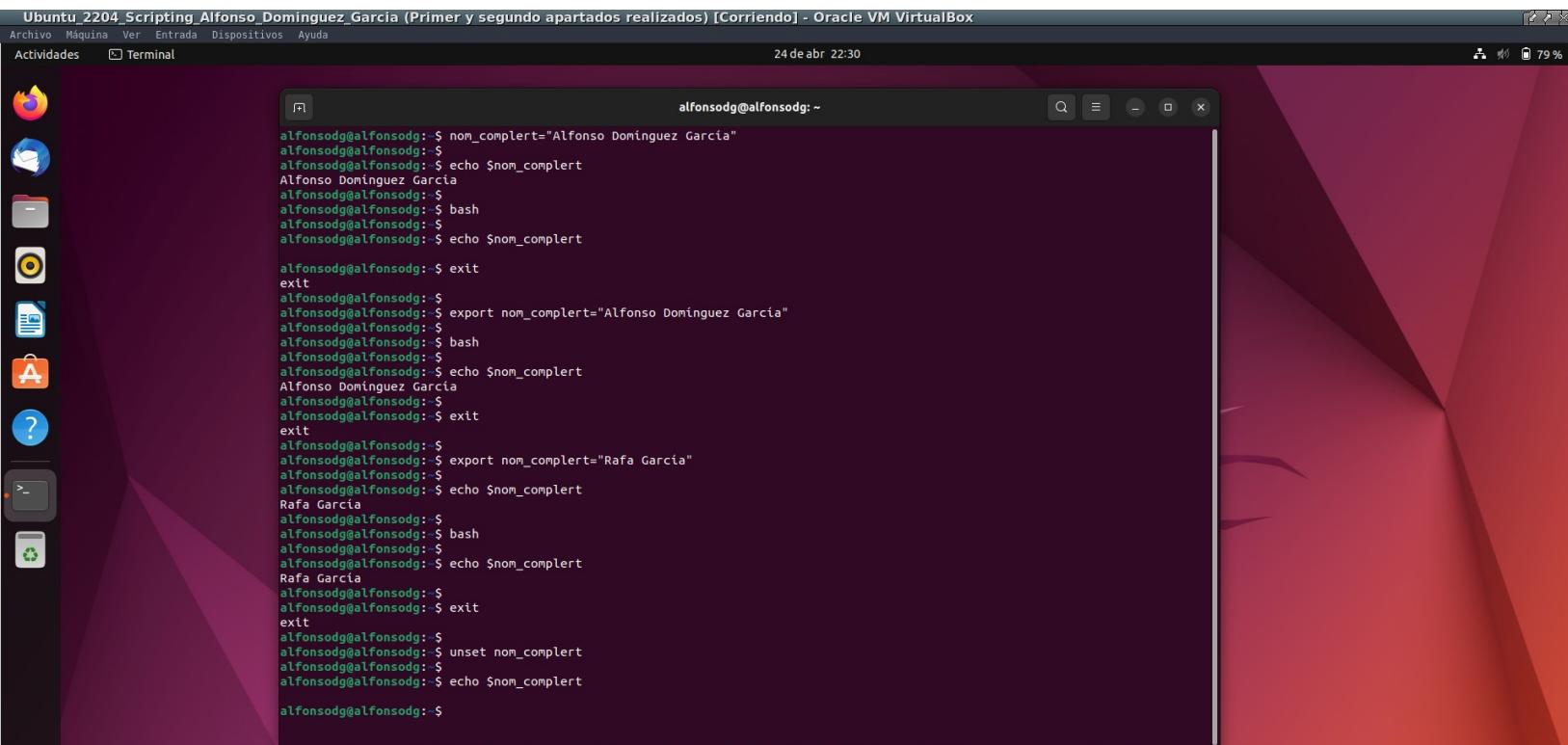
```
alfonsodg@alfonsodg:~$ source act1.sh
Hola alfonsodg
Vaig a llistar el contingut del directori actual
Descargas Documentos Escritorio Imágenes Música Plantillas Público scripts snap Vídeos
Vaig a configurar 2 variables
Aquesta és una cadena de text: blau
I aquest és un número: 3
Et retorno al teu prompt

alfonsodg@alfonsodg:~$ echo $COLOR
blau
alfonsodg@alfonsodg:~$ echo $VALOR
3
alfonsodg@alfonsodg:~$
```

ACT. 2

-Todos los apartados: a), b), c), d), e), f) y g) → Muestro captura de pantalla a continuación:

-Contestando al apartado f) → Al exportar la variable “nom_complert=Rafa García” se sustituirá la variable local por la nueva variable global o variable de entorno que pueden usarse en ambos shells tanto del shell padre como del subshell hijo.



The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open in the center, titled "alfonsodg@alfonsodg: ~". The terminal displays the following command-line session:

```
alfonsodg@alfonsodg:~$ nom_complert="Alfonso Dominguez Garcia"
alfonsodg@alfonsodg:~$ echo $nom_complert
Alfonso Dominguez Garcia
alfonsodg@alfonsodg:~$ bash
alfonsodg@alfonsodg:~$ bash
alfonsodg@alfonsodg:~$ echo $nom_complert
alfonsodg@alfonsodg:~$ exit
exit
alfonsodg@alfonsodg:~$ export nom_complert="Alfonso Dominguez Garcia"
alfonsodg@alfonsodg:~$ echo $nom_complert
Alfonso Dominguez Garcia
alfonsodg@alfonsodg:~$ exit
alfonsodg@alfonsodg:~$ Rafa Garcia
alfonsodg@alfonsodg:~$ bash
alfonsodg@alfonsodg:~$ bash
alfonsodg@alfonsodg:~$ echo $nom_complert
Rafa Garcia
alfonsodg@alfonsodg:~$ unset nom_complert
alfonsodg@alfonsodg:~$ echo $nom_complert
alfonsodg@alfonsodg:~$
```

h)

Variable Entorno Reservada	Descripción
\$HOME	El directorio raíz del usuario actual
\$PATH	Una lista separada por ':' de directorios en los que se buscarán ejecutables de comandos
\$PWD	El directorio de trabajo actual
\$OLDPWD	El directorio de trabajo anterior
\$USER	El nombre de usuario del usuario actual
\$SHELL	La ruta del shell actual
\$TERM	El tipo de terminal utilizado por el usuario
\$HOSTNAME	El nombre del host del sistema
\$OSTYPE	El tipo de sistema operativo (por ejemplo, linux-gnu)
\$BASH_VERSION	La versión de Bash
\$SECONDS	El número de segundos desde que se inició el shell
\$RANDOM	Un número entero aleatorio entre 0 y 32767
\$0	El nombre del script actua
\$1, \$2, \$3,	Los argumentos del script
\$#	El número de argumentos pasados al script
\$*	Todos los argumentos del script como una sola cadena
\$@	Todos los argumentos del script como una lista de cadenas
\$?	El código de salida del último comando ejecutado
\$\$	El ID del proceso actual
\$!	El ID del último proceso ejecutado en segundo plano
\$IFS	El separador de campos interno\$LINEOEI número de línea actual en el script
\$PIPESTATUS	Un arreglo de códigos de salida de cada comando en una tubería
\$PPID	El ID del proceso padre
\$PS1	El prompt primario del shell
\$PS2	El prompt secundario del shell

\$IFS	El separador de campos interno
\$UID	El ID del usuario actual
\$EUID	El ID de usuario efectivo actual

ACT. 3

-En la actividad número 3 crearemos el script **rectangle.sh** que pasará dos argumentos **\$1** y **\$2** que pasará el usuario al ejecutar el script **rectangle.sh**

-Muestro captura de pantalla conforme muestro el propio script (captura derecha) y su ejecución (captura izquierda). Básicamente este script permite multiplicar la base por la altura del triangulo rectángulo usando una variable llamada **RESULT**.

-Creada de la siguiente forma → **RESULT=\$((\$1 * \$2))**

-Finalmente mostramos una salida de cript simple que será de la siguiente forma → **echo "El área de base \$1 y de altura \$2 es igual a \$RESULT"** de esta forma es equivalente a un print final.

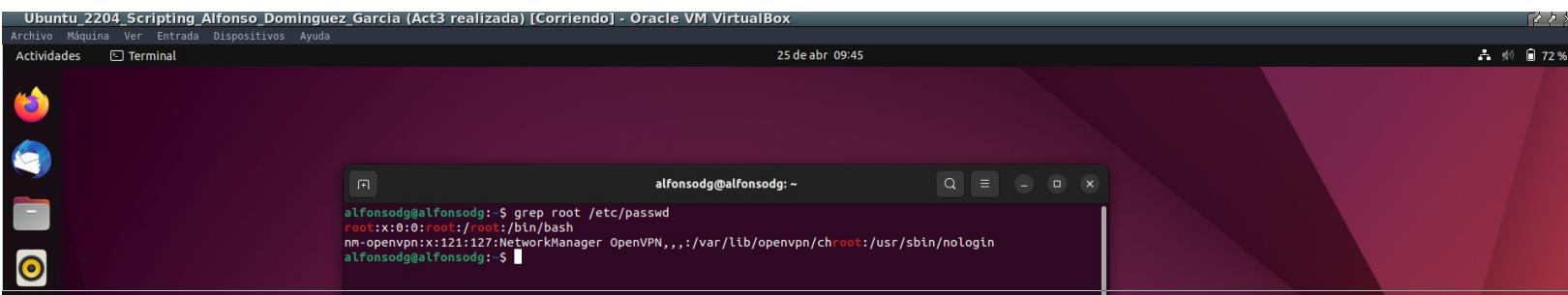
The screenshot shows an Ubuntu 22.04 desktop environment. On the left, a terminal window displays the command `rectangle.sh 10 5` followed by the output `Hola alfonsodg`. Below this, a message states: `El script rectangle.sh permitirá calcular la superficie de un triangulo rectángulo` and `El área de base 10 y altura 5 es igual a 50`. On the right, another terminal window shows the script's code in a `GNU nano 6.2` editor. The code includes `echo "Hola $USER"`, `echo`, `echo "El script rectangle.sh permitirá calcular la superficie de un triangulo rectángulo"`, `echo`, `RESULT=$(($1 * $2))`, and `echo "El área de base $1 y altura $2 es igual a $RESULT"`. The desktop interface includes a dock with various icons and a system tray at the top.

ACT. 4

a)

-Primero debemos de buscar la palabra root usando el comando grep apuntando al fichero /etc/fstab

grep root /etc/passwd



The screenshot shows a terminal window titled "Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (Act3 realizada) [Corriendo] - Oracle VM VirtualBox". The terminal window has a dark background and contains the following text:

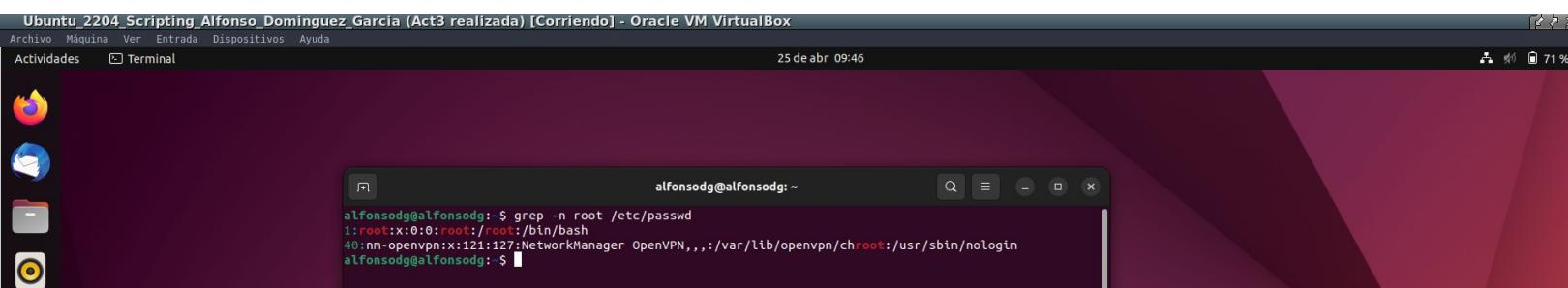
```
alfonsodg@alfonsodg: ~
alfonsodg@alfonsodg: $ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
nn-openvpn:x:121:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
alfonsodg@alfonsodg: $
```

b)

Usando la opción -n al comando grep de root apuntando el fichero /etc/passwd: Ejecutaremos el siguiente comando:

grep -n root /etc/passwd

-Mostrará solamente la palabra root y con las líneas justo a la izquierda



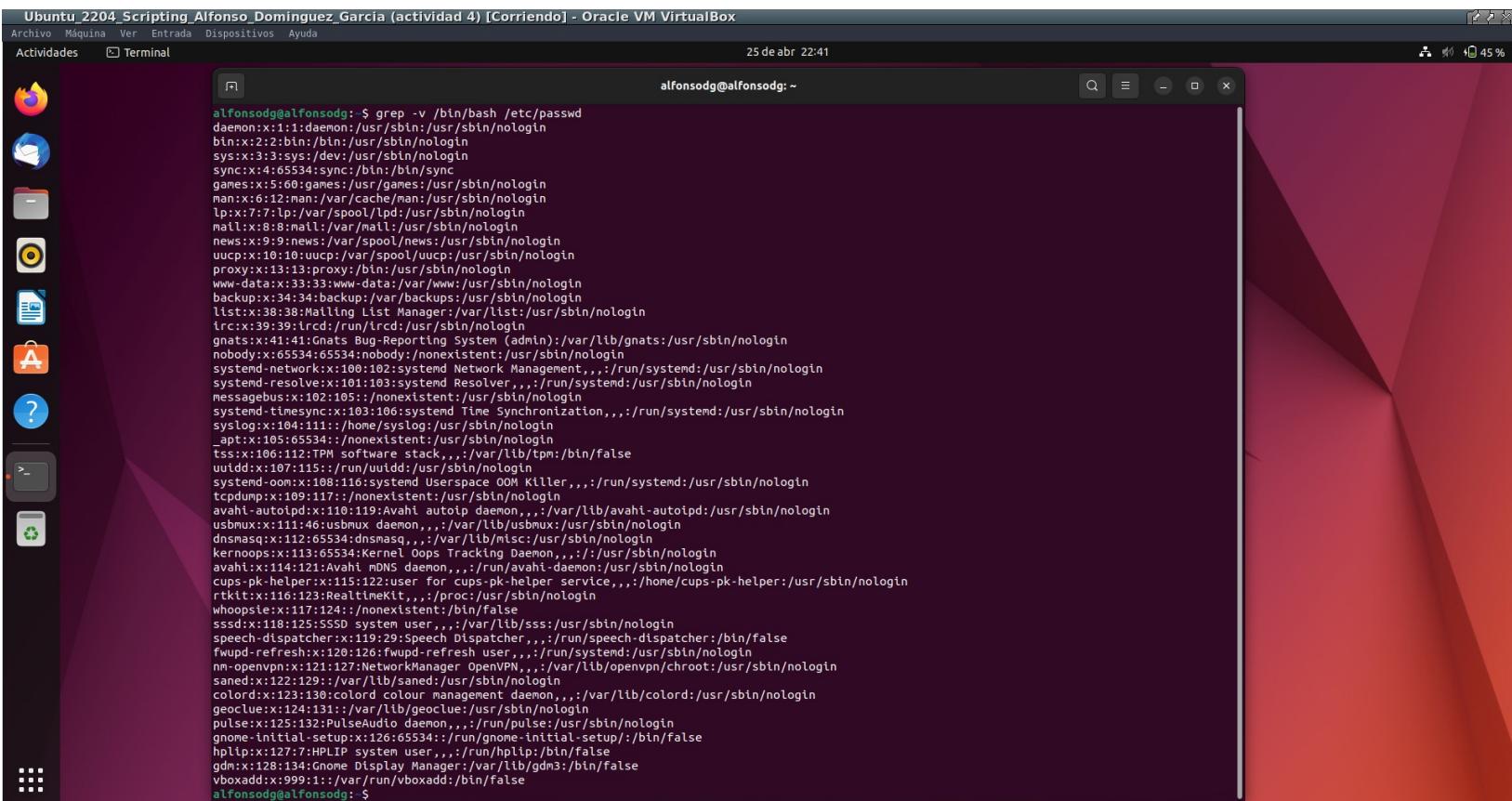
The screenshot shows a terminal window titled "Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (Act3 realizada) [Corriendo] - Oracle VM VirtualBox". The terminal window has a dark background and contains the following text:

```
alfonsodg@alfonsodg: ~
alfonsodg@alfonsodg: $ grep -n root /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
40:nn-openvpn:x:121:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
alfonsodg@alfonsodg: $
```

c)

-Ahora debemos de buscar todos los usuarios que NO utilicen el shell /bin/bash, de esta forma debemos de ejecutar el siguiente comando:

grep -v /bin/bash /etc/passwd



The screenshot shows a terminal window titled "Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox". The terminal displays the command "grep -v /bin/bash /etc/passwd" followed by a long list of user entries from the passwd file. Most users have "/bin/bash" as their shell, but many others like "nobody", "nologin", and various system daemons have different shells like "/bin/false" or "/usr/sbin/nologin". The terminal window has a dark theme and is located on a desktop with a red and orange gradient background.

```
alfonsodg@alfonsodg: ~
alfonsodg@alfonsodg: $ grep -v /bin/bash /etc/passwd
daemon:x:1:1:daemon:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
ircd:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:104:111:/home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/nonexistent:/usr/sbin/nologin
tss:x:106:112:TPM software stack,,,:/var/lib/tmpm:/bin/false
uidd:x:107:115:/run/uidd:/usr/sbin/nologin
systemd-oom:x:108:116:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nologin
tcpdump:x:109:117:/nonexistent:/usr/sbin/nologin
avahi-autopd:x:110:119:Avahi autopd daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/msc:/usr/sbin/nologin
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
avahi:x:114:121:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:115:122:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
rtkit:x:116:123:RealtimeKit,,,:/proc:/usr/sbin/nologin
whoopsie:x:117:124:/nonexistent:/bin/false
sssd:x:118:125:SSSD system user,,,:/var/lib/sssd:/usr/sbin/nologin
speech-dispatcher:x:119:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
fwupd-refresh:x:120:126:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
nm-openvpn:x:121:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
saned:x:122:129:/var/lib/saned:/usr/sbin/nologin
colord:x:123:130:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:124:131:/var/lib/geoclue:/usr/sbin/nologin
pulse:x:125:132:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:126:65534:/run/gnome-initial-setup:/bin/false
hplip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false
vboxadd:x:999:1:/var/run/vboxadd:/bin/false
alfonsodg@alfonsodg: $
```

d)

-Ahora debemos de seguir buscando los usuarios que NO contengan el shell /bin/bash en el fichero /etc/passwd y concatenamos con una PIPE y la opción -v que buscará los que NO contengan tampoco shell NOLOGIN usando el siguiente comando:

grep -v /bin/bash /etc/passwd | grep -v nologin

```
Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 25 de abr 22:43
alfonsodg@alfonsodg: ~
alfonsodg@alfonsodg: $ grep -v /bin/bash /etc/passwd | grep -v nologin
sync:x:4:65534:sync:/bin/sync
tss:x:106:112:TPM software stack,,,:/var/lib/tpm:/bin/false
whoopsie:x:117:124::/nonexistent:/bin/false
speech-dispatcher:x:119:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
gnome-initial-setup:x:126:65534::/run/gnome-initial-setup:/bin/false
hplip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
alfonsodg@alfonsodg: $
```

e)

-Ahora usaremos las opciones -n (mostrará el número de líneas) y la opción -i que ignorará mayúsculas o minúsculas y buscaremos el shell false apuntando al fichero /etc/passwd

grep -ni false /etc/passwd

grep -i -c "false" /etc/passwd

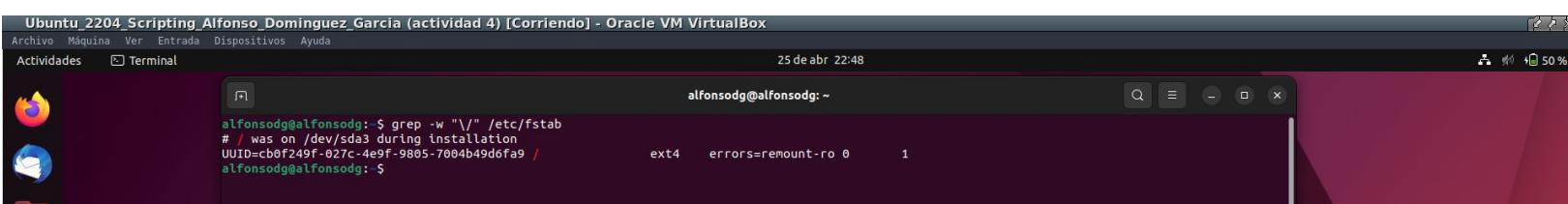
```
Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 25 de abr 22:45
alfonsodg@alfonsodg: ~
alfonsodg@alfonsodg: $ grep -ni false /etc/passwd
25:ts:x:106:112:TPM software stack,,,:/var/lib/tpm:/bin/false
30:whoopsie:x:117:124::/nonexistent:/bin/false
38:speech-dispatcher:x:119:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
45:gnome-initial-setup:x:126:65534::/run/gnome-initial-setup:/bin/false
46:hplip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false
47:gdm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false
49:vboxadd:x:999:1::/var/run/vboxadd:/bin/false
alfonsodg@alfonsodg: $ alfonsodg@alfonsodg: $ grep -i false /etc/passwd | wc -l
7
alfonsodg@alfonsodg: $
```

f)

-El último paso será buscar en el fichero /etc/fstab la partición raíz usando la opción -w que permitirá validar una subcadena o cadena que le pasemos, en este caso habrá que escapar el carácter "/" ya que haremos referencia a "/" de que es el punto raíz montado en /etc/fstab.

-Para escapara este carácter podemos usar lo siguiente "V" d eesta forma devolverá en formato string usando la opción -w la / del punto raíz. Usaremos el siguiente comandos

grep -w "V" /etc/fstab



The screenshot shows a terminal window titled "Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox". The terminal is running on a dark-themed desktop environment. The command "grep -w \"V\" /etc/fstab" was entered, and the output is displayed:

```
alfonsodg@alfonsodg: ~
alfonsodg@alfonsodg: $ grep -w "/" /etc/fstab
# / was on /dev/sda3 during installation
UUID=cb0f249f-027c-4e9f-9805-7004b49d6fa9 /
ext4    errors=remount-ro 0         1
alfonsodg@alfonsodg: $
```

ACT. 5

a)

-Primero creamos el directorio “**classes**” y después crearemos los directorios consecutivamente “**Docs imatges 259**” y luego seguiremos con los ejercicios a continuación:

A screenshot of a terminal window titled "alfonsodg@alfonsodg: ~/classes". The terminal shows the following command history:

```
alfonsodg@alfonsodg:~$ pwd
/home/alfonsodg
alfonsodg@alfonsodg:~$ mkdir classes
alfonsodg@alfonsodg:~$ cd classes/
alfonsodg@alfonsodg:~/classes$ 
alfonsodg@alfonsodg:~/classes$ mkdir Docs imatges 259
alfonsodg@alfonsodg:~/classes$ 
alfonsodg@alfonsodg:~/classes$
```

The terminal is part of a desktop environment with a dark theme, and the desktop icons are visible on the left.

b)

-El siguiente paso será ejecutar un `ls -Id` para listar todos los directorios dentro de classes. Pero, usando el método CLASS.

Is -Id [:alnum:]*)

-En este caso estaremos printando los directorios que sean numéricos, alfanuméricos, todos los dígitos del 0-9 en intervalo y el asterisco significan uno o más valores:

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window. The terminal window title is "alfonsodg@alfonsodg: ~/classes". The terminal displays the command "ls -ld [[:alnum:]]*" followed by a list of files: 259, Docs, and imagenes. The desktop background is dark purple/red, and the taskbar at the bottom shows various application icons.

c)

-En este apartado mostrará los valores que contengan en minúsculas los directorios listados. De esta forma mostrará solamente el directorio “**imatges**”:

ls -ld [:lower:]*

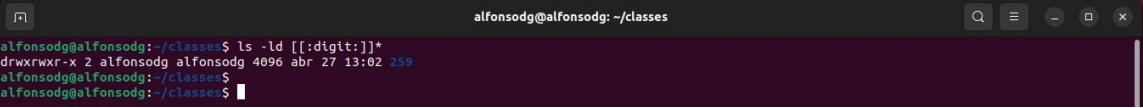


```
Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 27 de abr 13:05
alfonsodg@alfonsodg: ~/classes
alfonsodg@alfonsodg:~/classes$ ls -ld [:lower:]*
drwxrwxr-x 2 alfonsodg alfonsodg 4096 abr 27 13:02 imatges
alfonsodg@alfonsodg:~/classes$ alfonsodg@alfonsodg:~/classes$
```

d)

-Ahora mostraré solamente los directorios que contengan números:

ls -ld [:digit:]*

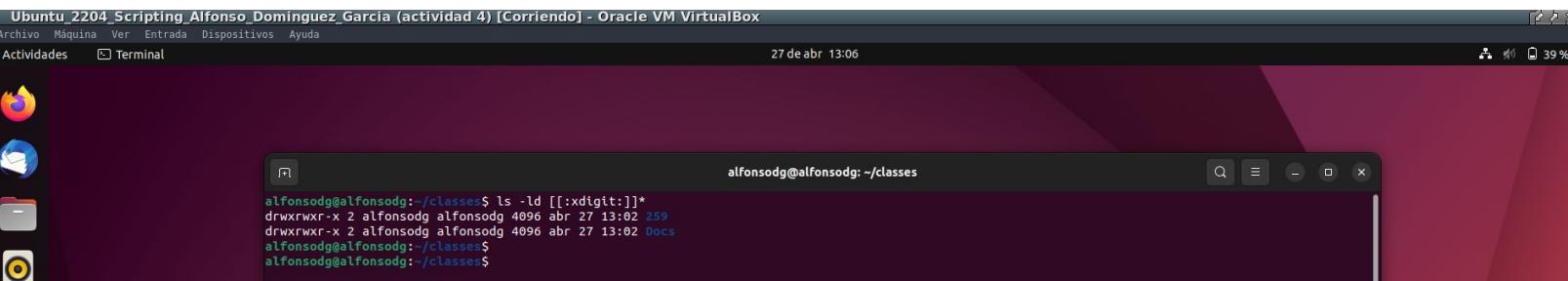


```
Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 27 de abr 13:05
alfonsodg@alfonsodg: ~/classes
alfonsodg@alfonsodg:~/classes$ ls -ld [:digit:]*
drwxrwxr-x 2 alfonsodg alfonsodg 4096 abr 27 13:02 259
alfonsodg@alfonsodg:~/classes$ alfonsodg@alfonsodg:~/classes$
```

e)

-Obtenemos esta salida ya que los directorios **Docs** y **259** son hexadecimales, es decir que contienen dígitos del **0 – 9 en intervalo** y de las letras de la **A-F en mayúsculas o minúsculas**

```
ls -ld [:xdigit:]*
```



The screenshot shows a terminal window titled "Ubuntu 22.04 Scripting Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox". The window is running on a desktop environment with a dark theme. The terminal displays the following command and its output:

```
alfonsodg@alfonsodg:~/classes$ ls -ld [:xdigit:]*
drwxrwxr-x 2 alfonsodg alfonsodg 4096 abr 27 13:02 259
drwxrwxr-x 2 alfonsodg alfonsodg 4096 abr 27 13:02 Docs
alfonsodg@alfonsodg:~/classes$
```

The terminal window has a standard Linux-style interface with a title bar, a scroll bar, and a status bar at the bottom showing the date and time.

ACT. 6

a)

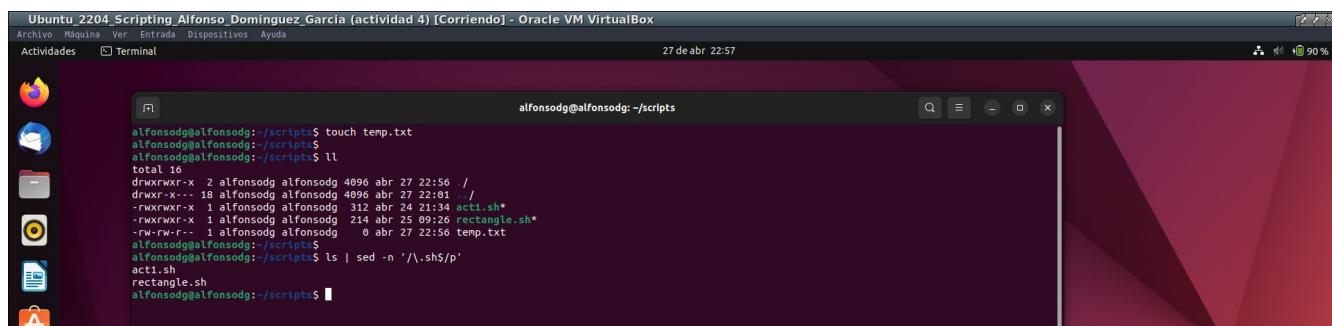
-Primero tendremos que crear el fichero **temp.txt** usando el comando **touch temp.txt** dentro de nuestro directorio scripts.

touch temp.txt

-Debemos de buscar los archivos dentro del directorio scripts todos nuestros scripts que acaben por **.sh\$** Para ello usaremos **Stream Editor** para poder filtrar los valores

ls | sed -n '/.sh\$/p'

Escaparemos el carácter \ para poder hacer el . De sh:



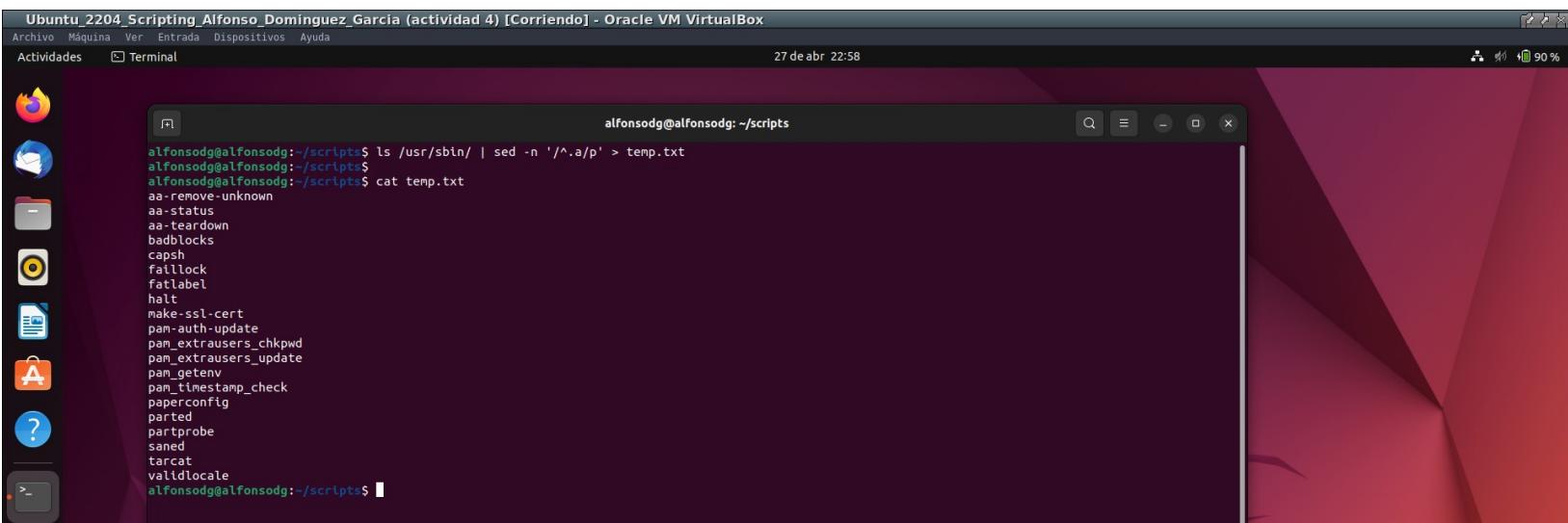
```
Ubuntu_2204.Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 27 de abr 22:57
alfonsodg@alfonsodg:~/scripts
touch temp.txt
alfonsodg@alfonsodg:~/scripts
alfonsodg@alfonsodg:~/scripts$ ll
total 0
drwxrwxr-x  2 alfonsodg alfonsodg 4096 abr 27 22:56 /
drwxr-x--- 18 alfonsodg alfonsodg 4096 abr 27 22:01 ..
-rwxrwxr-x  1 alfonsodg alfonsodg 312 abr 24 21:34 act1.sh*
-rwxrwxr-x  1 alfonsodg alfonsodg 214 abr 25 09:26 rectangle.sh*
-rw-rw-r--  1 alfonsodg alfonsodg   0 abr 27 22:56 temp.txt
alfonsodg@alfonsodg:~/scripts$ alfonsodg@alfonsodg:~/scripts$ ls | sed -n '/.sh$/p'
act1.sh
rectangle.sh
alfonsodg@alfonsodg:~/scripts$
```

b)

-Ahora debemos de mostrar un resultado usando SED que muestre por pantalla los valores que empiezan y contengan un valor para luego mostrar la letra a.

-Es decir, buscará y añadirá a temp.txt todo el contenido que empieza y que como segundo carácter la letra "a"

ls /usr/sbin | sed -n '/^a/p > temp.txt



The screenshot shows a terminal window titled 'Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox'. The terminal window has a dark background and contains the following text:

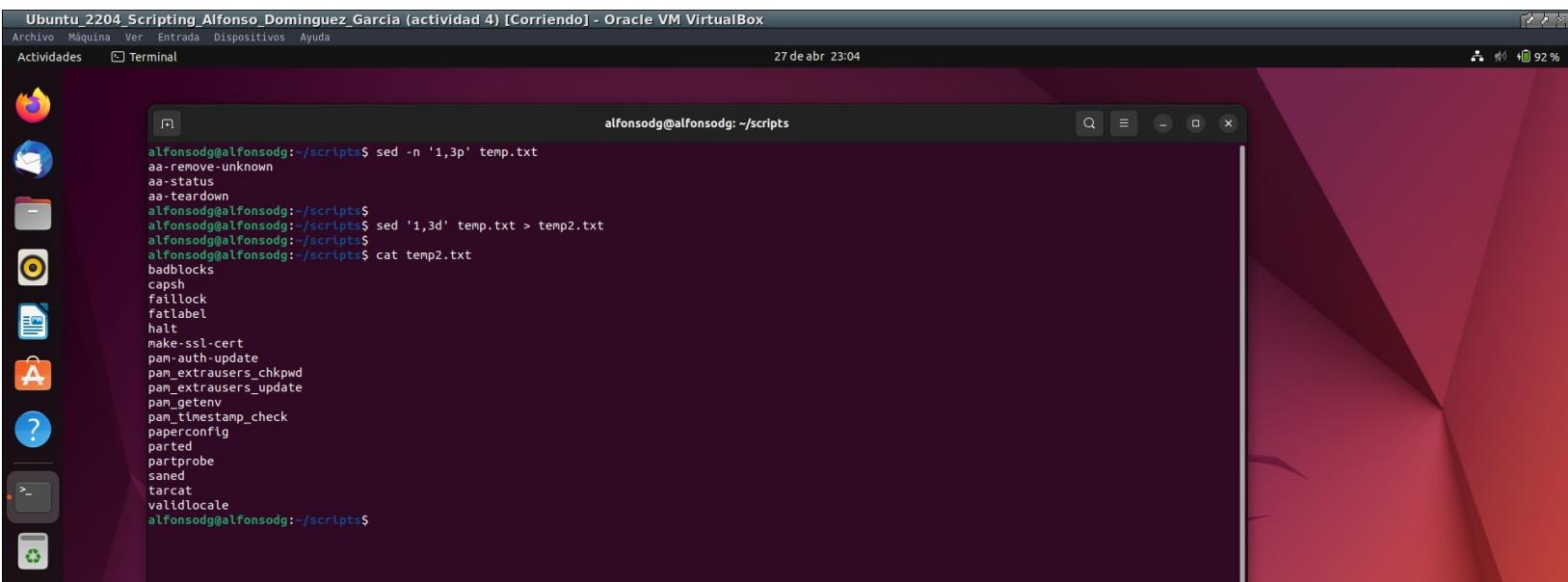
```
alfonsodg@alfonsodg:~/scripts$ ls /usr/sbin | sed -n '/^a/p > temp.txt
alfonsodg@alfonsodg:~/scripts$ cat temp.txt
aa-remove-unknown
aa-status
aa-teardown
badblocks
capsh
faillock
fatlabel
halt
make-ssl-cert
pam-auth-update
pam_extrousers_chpwd
pam_extrousers_update
pam_getenv
pam_timestamp_check
paperconfig
parted
partprobe
saned
tarcat
validlocale
alfonsodg@alfonsodg:~/scripts$
```

c)

-Primero tal y como indica el enunciado mostraremos las tres primeras líneas del fichero temp.txt

sed -n '1,3p' temp.txt

-Luego y para finalizar el ejercicio nos pide que eliminemos estas tres líneas anteriores y redirijamos el comando al fichero temp2.txt de tal forma que mostrará el contenido restante de temp.txt sin contar

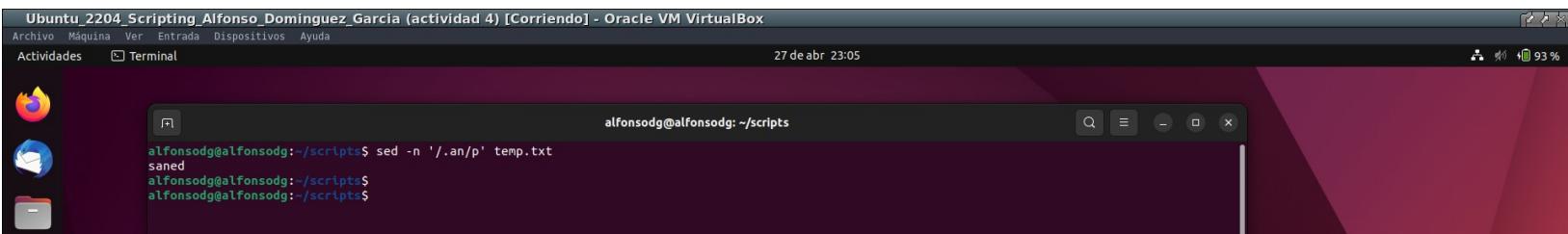


```
Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 27 de abr 23:04
alfonsodg@alfonsodg:~/scripts$ sed -n '1,3p' temp.txt
aa-remove-unknown
aa-status
aa-teardown
alfonsodg@alfonsodg:~/scripts$ sed '1,3d' temp.txt > temp2.txt
alfonsodg@alfonsodg:~/scripts$ cat temp2.txt
badblocks
capslock
faillock
fatlabel
halt
make-ssl-cert
pam-auth-update
pam_extrousers_chpwd
pam_extrousers_update
pam_getenv
pam_timestamp_check
paperconfig
parted
partprobe
saned
tarcat
validlocale
alfonsodg@alfonsodg:~/scripts$
```

d)

-Para mostrar el contenido del fichero temp.txt que contenga la palabra "an" debemos de ejecutar el siguiente comando

sed -n '/.an/p' temp.txt



```
Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 27 de abr 23:05
alfonsodg@alfonsodg:~/scripts$ sed -n '/.an/p' temp.txt
saned
alfonsodg@alfonsodg:~/scripts$ alfonsodg@alfonsodg:~/scripts$
```

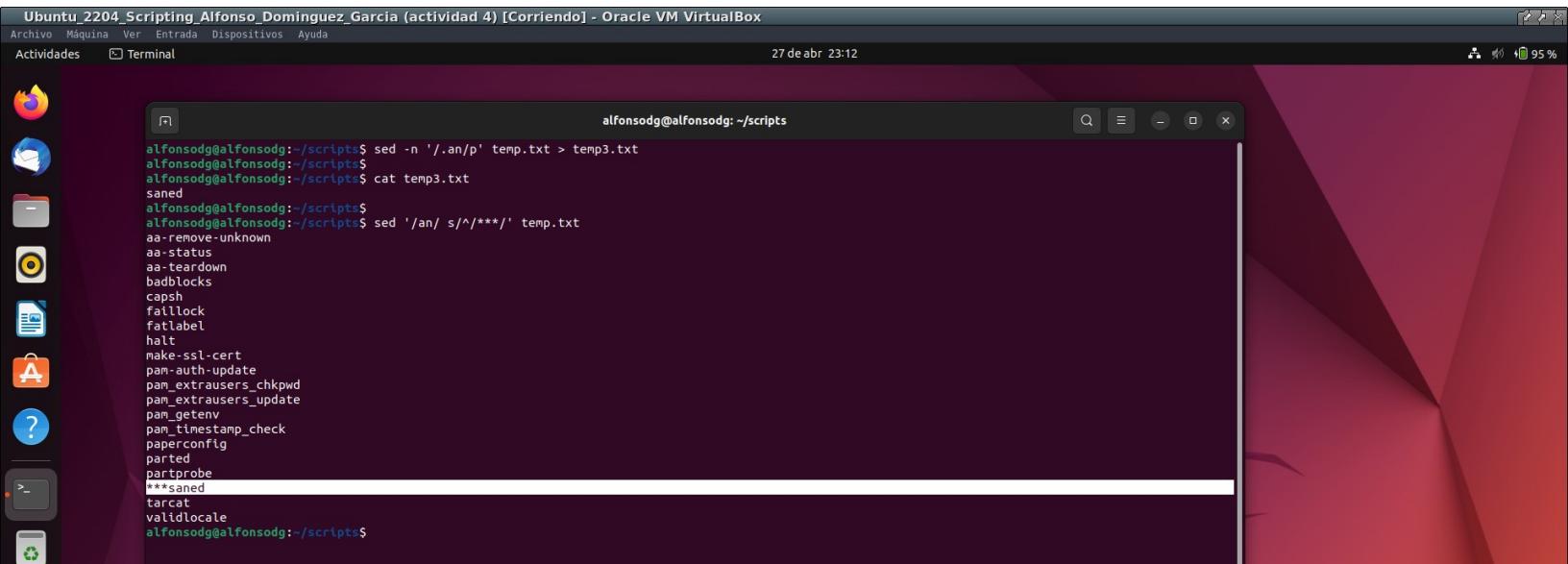
e)

-El siguiente paso será redirigir este comando con el contenido que solamente contenga los caracteres “an” en el fichero temp3.txt

-El último paso será que en el fichero temp.txt tendremos que sustituir los valores que contengan “an” por “***”, para ello debemos de ejecutar el siguiente comando:

sed -n '/an/ s/^//' temp.txt**

-Que mostrará con triple asterisco justo delante de la palabra, el problema es que el fichero NO contiene ningún MAN o que empiece por man. Solamente en “an”



The screenshot shows a terminal window titled "Terminal" running on an Ubuntu 22.04 desktop environment. The terminal window has a dark theme. The command history in the terminal shows the following sequence:

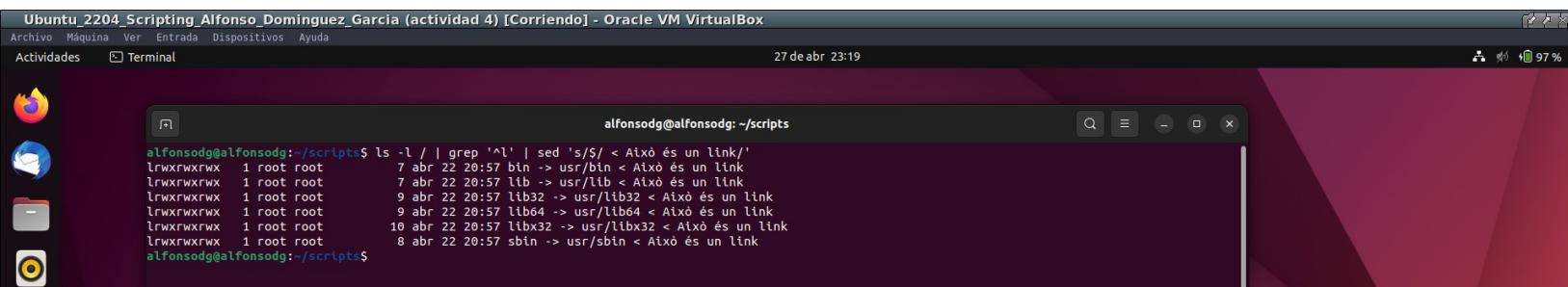
```
alfonsodg@alfonsodg:~/scripts$ sed -n '/.an/p' temp.txt > temp3.txt
alfonsodg@alfonsodg:~/scripts$ cat temp3.txt
saned
saned
alfonsodg@alfonsodg:~/scripts$ alfonsodg@alfonsodg:~/scripts$ sed '/an/ s/^/**/' temp.txt
aa-remove-unknown
aa-status
aa-teardown
badblocks
caps
faillock
fatlabel
halt
make-ssl-cert
pam-auth-update
pam_extrousers_chpwd
pam_extrousers_update
pam_getenv
pam_timestamp_check
paperconfig
parted
partprobe
***saned
tarcat
validlocale
alfonsodg@alfonsodg:~/scripts$
```

The output of the command shows that the word "saned" appears twice, each preceded by three asterisks ("***"). This indicates that the regular expression '/an/' matched both the full word "saned" and the prefix "saned" of the word "saned".

f)

-Para poder mostrar todos los valores del directorio raíz que contienen enlace simbólico es ejecutar primero un ls -l y ejecutamos también una combinación usando **grep '^l'** que empiecen por **l o link**

-Y finalmente usaremos el SED para sustituir al final de línea usando '**s/\$/ < Això és un link/**' de esta forma printará al final de cada línea de los ficheros simbólicos mostrar el texto que hemos creado.



The screenshot shows a terminal window titled 'Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox'. The terminal content is as follows:

```
alfonsodg@alfonsodg:~/scripts$ ls -l / | grep '^l' | sed 's/$/ < Això és un link/'  
lrwxrwxrwx 1 root root 7 abr 22 20:57 bin -> usr/bin < Això és un link  
lrwxrwxrwx 1 root root 7 abr 22 20:57 lib -> usr/lib < Això és un link  
lrwxrwxrwx 1 root root 9 abr 22 20:57 lib32 -> usr/lib32 < Això és un link  
lrwxrwxrwx 1 root root 9 abr 22 20:57 lib64 -> usr/lib64 < Això és un link  
lrwxrwxrwx 1 root root 10 abr 22 20:57 libx32 -> usr/libx32 < Això és un link  
lrwxrwxrwx 1 root root 8 abr 22 20:57 sbin -> usr/sbin < Això és un link  
alfonsodg@alfonsodg:~/scripts$
```

Otra forma de hacerlo → **ll / | sed -n '/^l.*p' | sed 's/\$/ < Això és un link/'**

g)

-Muestro captura de pantalla a continuación del script “blancs.sh” que mostrará el conteo de líneas en el script que el usuario pase como argumento y luego contará el número de espacios en blanco sobre el fichero pasado como argumento.

-Los comandos principales usando Stream Editor han sido:

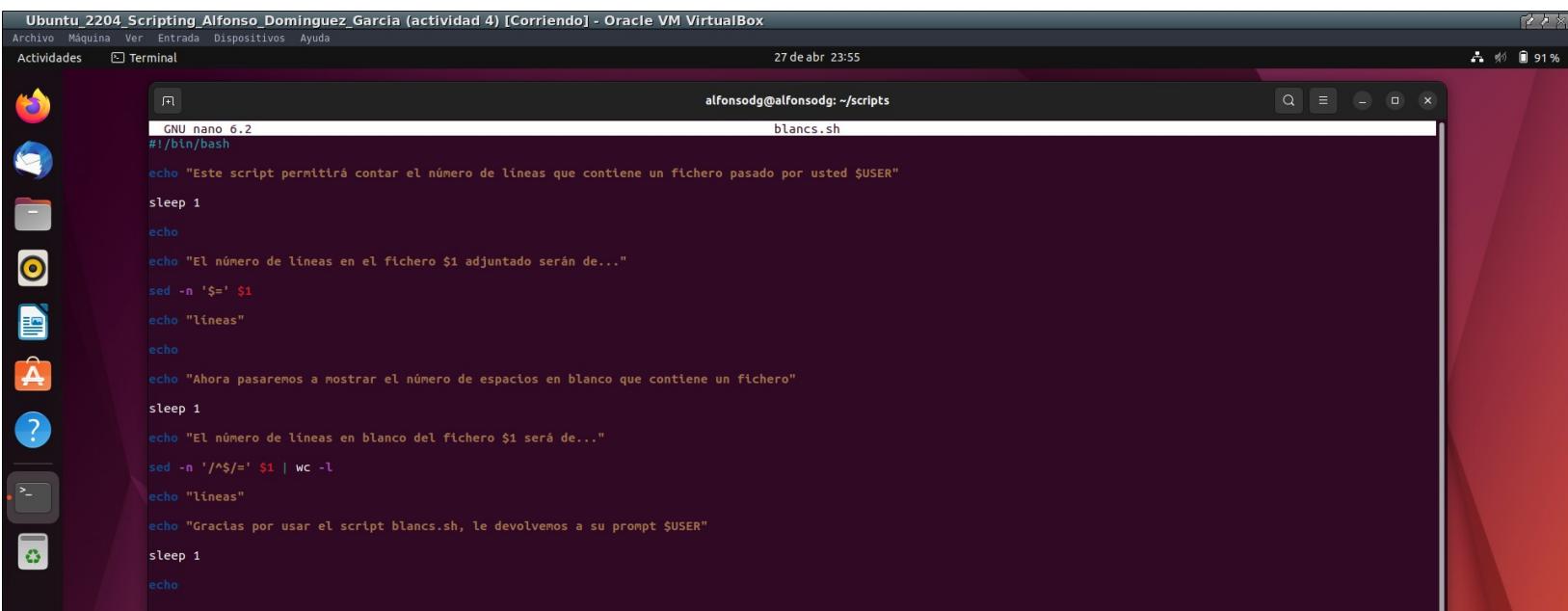
sed -n '\$=' \$1

-Otra forma → **sed -n '/\$/p' \$1**

-Con el comando superior contaremos el número de líneas que tiene el código

sed -n '\$=' \$1 | wc -l

-Y este segundo comando mostrará el número de espacios en blanco que pueda encontrar en el fichero adjuntado como argumento.



The screenshot shows a terminal window titled "Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (actividad 4) [Corriendo] - Oracle VM VirtualBox". The terminal is running a bash script named "blancs.sh". The script uses echo statements to provide instructions and sed commands to count lines and spaces in a file. The terminal window has a dark theme and is located on a desktop with a red background. The script output includes messages about counting lines and spaces, and it ends with a thank you message.

```
GNU nano 6.2
#!/bin/bash

echo "Este script permitirá contar el número de líneas que contiene un fichero pasado por usted $USER"
sleep 1
echo
echo "El número de líneas en el fichero $1 adjuntado serán de..."
sed -n '$=' $1
echo "líneas"
echo
echo "Ahora pasaremos a mostrar el número de espacios en blanco que contiene un fichero"
sleep 1
echo "El número de líneas en blanco del fichero $1 será de..."
sed -n '/$/p' $1 | wc -l
echo "líneas"
echo "Gracias por usar el script blancs.sh, le devolvemos a su prompt $USER"
sleep 1
echo
```

ACT. 6 | EXTRA

h) EXTRA -> "UTILIZANDO SED DENTRO DEL SCRIPT PARA HACER LAS DISTINTOS CAMBIOS
Crear un script en Ubuntu que permita modificar el contenido de un fichero y añadirlo a otro.

El script debe recibir dos parámetros, el nombre del primer fichero y el nombre del segundo fichero. Además, se debe sustituir todas las palabras "Hola" y "adios" (en mayúsculas o minúsculas) por "Hi" y "Bye", respectivamente.

También se debe añadir una línea al principio del segundo fichero indicando que el ejercicio ha sido creado por el usuario que ha ejecutado el script y la fecha y la hora en cuestión. Al ejecutarse el script hay que devolver el siguiente texto por pantalla:

El fichero X y el fichero Y han sido modificados por el usuario Z correctamente. No es necesario comprobar que existen los ficheros ya que no hemos visto condicionales.

El fichero1 debe de tener el siguiente contenido: Hola mundo Adiós mundo Hola, ¿cómo estás? adios, nos vemos pronto Hola amigos Adiós amigos hola universo adiós universo hola a todos adiós a todos"

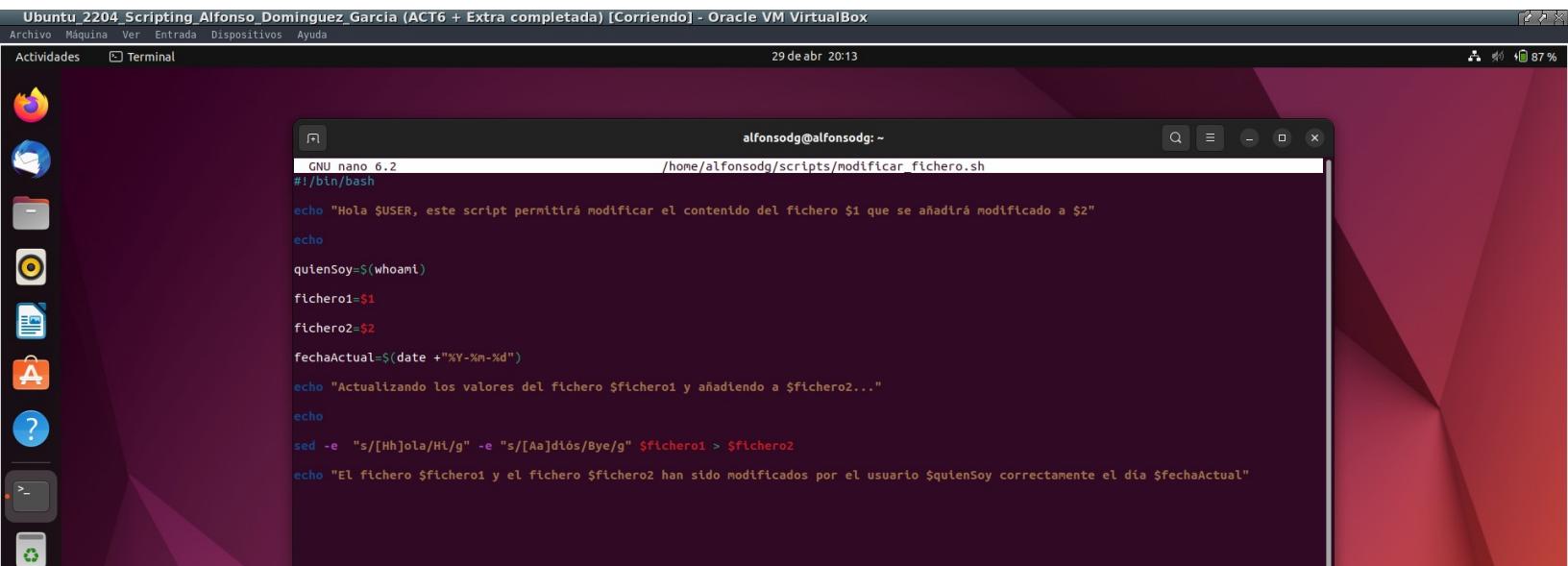
-Lo primero que haremos será crear un script que he llamado "**modificar_fichero.sh**" al que le tendremos que pasar dos parámetros de fichero que existan previamente que serán **\$1 y \$2**

-En mi caso las he almacenado en dos variables distintas que se llamarán **fichero1=\$1 y fichero2=\$2** | También he almacenado quien es el usuario usado actualmente en la variable **quienSoy=\$(whoami)** | Y finalmente también he creado la variable de fecha actual usando el comando fecha | **fechaActual=\$(date + "%Y-%m-%d")**

-Finalmente y lo importante de este comando final de manera que modificamos el contenido del fichero **\$fichero1** y lo añadiremos a **\$fichero2** el contenido modificado.

sed -e "s/[Hh]ola/Hi/g" -e "s/[Aa]diós/Bye/g" \$fichero1 > \$fichero2

-Muestro captura de pantalla con el contenido de mi script creado anteriormente:



The screenshot shows a terminal window titled "Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (ACT6 + Extra completada) [Corriendo] - Oracle VM VirtualBox". The window is running on a desktop environment with a dark purple background. The terminal title bar includes the session name, host name "alfonsodg@alfonsodg:~", and the date/time "29 de abr 20:13". The terminal window contains the following content:

```
GNU nano 6.2                               /home/alfonsodg/scripts/modificar_fichero.sh
#!/bin/bash

echo "Hola $USER, este script permitirá modificar el contenido del fichero $1 que se añadirá modificando a $2"
echo
quiénSoy=$(whoami)
fichero1=$1
fichero2=$2
fechaActual=$(date +"%Y-%m-%d")
echo "Actualizando los valores del fichero $fichero1 y añadiendo a $fichero2..."
echo
sed -e "s/[Hh]ola/Hl/g" -e "s/[Aa]diós/Bye/g" $fichero1 > $fichero2
echo "El fichero $fichero1 y el fichero $fichero2 han sido modificados por el usuario $quiénSoy correctamente el dia $fechaActual"
```

-Aquí muestro captura de pantalla con la ejecución del script (*captura a mano derecha y conforme se pasan previamente como argumentos los fichero1.txt y fichero2.txt*). Luego muestro el contenido de *fichero1.txt* y *fichero2.txt* usando el comando *cat* (*captura de pantalla a mano derecha justo debajo de la ejecución*) | Finalmente en la *captura de pantalla (a mano izquierda)* muestro la creación de los ficheros *fichero1.txt* y *fichero2.txt* y *como conforme hago una comprobación entre ambos ficheros para mostrar que el fichero1.txt está vacío y más tarde se ha modificado el contenido y añadido al fichero2.txt*

The screenshot shows a Linux desktop environment with several open windows. In the top right, there's a terminal window titled 'alfonsodg@alfonsodg: ~' with the command 'modificar_fichero.sh fichero1.txt fichero2.txt'. It outputs a message about modifying the contents of fichero1.txt and adds it to fichero2.txt. It also shows an error message from sed. The date in the window is 29 abr 20:09.

Below this, another terminal window shows the output of 'll' in the '/scripts' directory, listing various files including 'fichero1.txt' and 'fichero2.txt'.

To the left, a file manager window shows the contents of 'fichero1.txt' and 'fichero2.txt'. The file 'fichero1.txt' contains the text 'Hola mundo', 'Adiós mundo', and 'Hola, ¿cómo estás?'. The file 'fichero2.txt' contains the text 'Adiós amigos', 'Hola amigos', 'Hola universo', 'adiós universo', 'hola a todos', 'adiós a todos', 'Bye mundo', 'Hi mundo', 'Hi, ¿cómo estás?', 'adios, nos vemos pronto', 'Hi amigos', 'Bye amigos', 'Hl universo', 'Bye universo', 'Hl a todos', and 'Bye a todos'.

At the bottom, another terminal window shows the output of 'cat ~/scripts/fichero1.txt' and 'cat ~/scripts/fichero2.txt' respectively, displaying the same content as the file manager window.

ACT. 7

a)

-Primero crearemos el fichero **alumnes** (se puede añadir la extensión .txt, en mi caso no he añadido una extensión tal y como indica el ejercicio). El siguiente paso será crear un fichero con el siguiente formato:

nombreUsuario:nombrePersona:apellidoPersona:teléfonoPersona

-Solamente tendremos que sustituir los campos de cada uno por lo que deseemos que salga en el script:

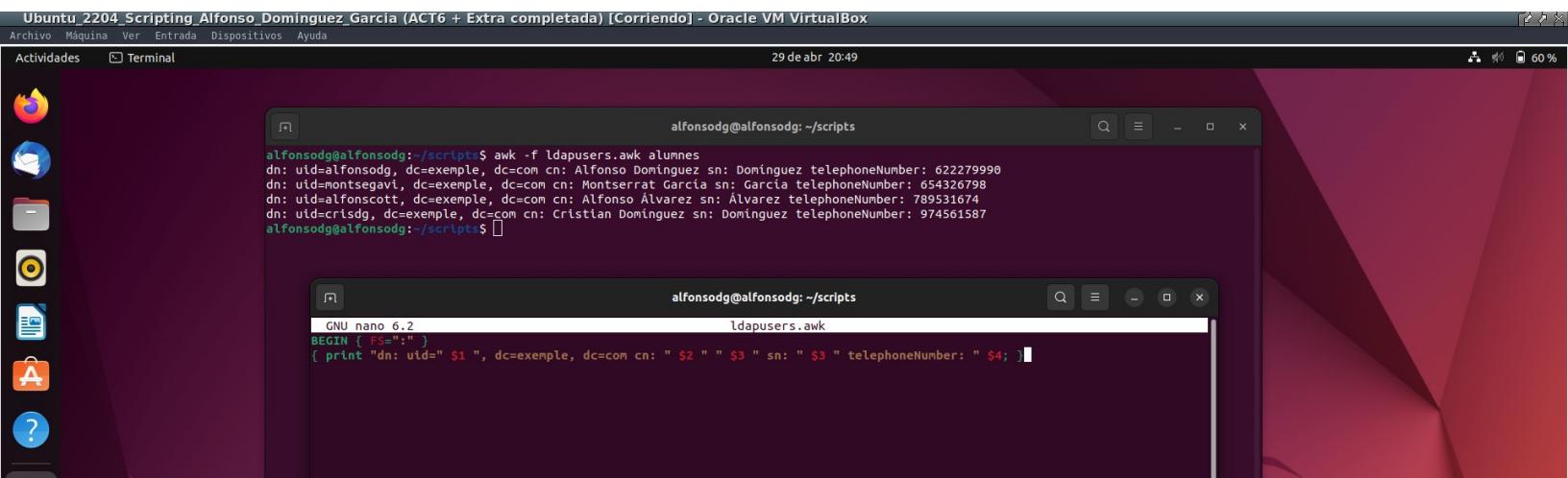
```
alfonsodg@alfonsodg: ~/scripts
alumnes
GNU nano 6.2
alfonsodg:Alfonso:Dominguez:622279990
montsegavil:Montserrat:Garcia:654326798
alfonscott:Alfonso:Álvarez:789531074
crtsgdg:Cristian:Dominguez:974561587
```

b)

-La primera línea del script **ldapusers.awk** significará usar un BEGIN que nos permitirá utilizar el separador **FS=":"** que separará por columnas en formato ":" cada columna del fichero anterior **alumnes**.

-Y finalmente en una única línea deberíamos de printar usando el comando **print { "dn: uid=" \$1 ", dc=example, dc=com cn: " \$2 " " \$3 " sn: " \$3 " telephoneNumber: " \$4; }** y cerramos el print con ";" y espacio al final del símbolo }

-Muestro captura de pantalla final con el contenido y el print final ejecutando el fichero usando el comando → **awk -f ldapusers.awk alumnes**



The screenshot shows a Linux desktop environment with a dark theme. At the top, there's a menu bar for "Ubuntu 22.04 Scripting Alfonso_Dominguez_Garcia (ACT6 + Extra completada) [Corriendo] - Oracle VM VirtualBox". Below it is a dock with various icons. In the center, there's a terminal window titled "alfonsodg@alfonsodg: ~/scripts" showing the output of the awk command. Below it is a nano editor window also titled "alfonsodg@alfonsodg: ~/scripts" containing the awk script code.

```
alfonsodg@alfonsodg:~/scripts$ awk -f ldapusers.awk alumnes
dn: uid=alfonsodg, dc=example, dc=com cn: Alfonso Dominguez sn: Dominguez telephoneNumber: 622279990
dn: uid=montsegavi, dc=example, dc=com cn: Montserrat Garcia sn: Garcia telephoneNumber: 654326798
dn: uid=alfonscott, dc=example, dc=com cn: Alfonso Alvarez sn: Alvarez telephoneNumber: 789531674
dn: uid=crisdg, dc=example, dc=com cn: Cristian Dominguez sn: Dominguez telephoneNumber: 974561587
alfonsodg@alfonsodg:~/scripts$ 
```

```
GNU nano 6.2                               ldapusers.awk
BEGIN { FS=":" }
{ print "dn: uid=" $1 ", dc=example, dc=com cn: " $2 " " $3 " sn: " $3 " telephoneNumber: " $4; }
```

ACT. 8

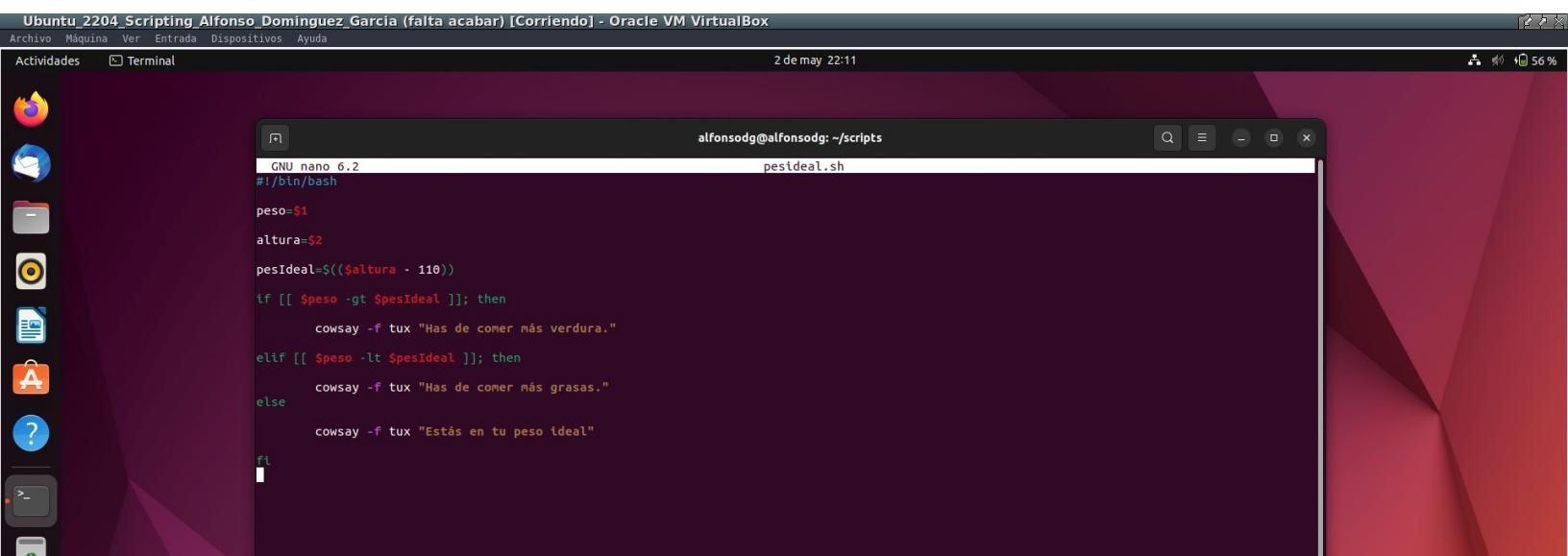
a)

-El primer paso será crear el script llamado “**pesideal.sh**” y que contenga la siguiente estructura de fichero .sh

-En este script lo que he realizado es primero almacenamos dos variables “**peso=\$1**” que pasará el usuario como argumento \$1 y lo guardamos en la variable “peso” y luego la segunda variable “**altura=\$2**” que pasará el usuario la altura.

-A continuación hacemos una pequeña operación → **pesIdeal=\$((altura - 110))** que es el cálculo que nos pide el enunciado, luego lo comparamos en un condicional que diga \$peso es mayor (usando -gt que es **Greater Than**) a la variable de \$pesIdeal pues pintaremos mediante **cowsay -f tux** (que mostrará el pingüino de Linux) un mensaje que diga “**Has de comer más verdura.**”

-Luego hacemos un segundo condicional que diga que si el **\$peso es Lower Than o -lt** entonces printará “**Has de comer más grasas**” y finalmente realizaremos una última condición que sea else y printamos por pantalla “**Estás en tu peso ideal**”



The screenshot shows a desktop environment with a terminal window titled "pesideal.sh" open in the "nano" text editor. The script content is as follows:

```
GNU nano 6.2
#!/bin/bash

peso=$1
altura=$2
pesIdeal=$((altura - 110))

if [[ $peso -gt $pesIdeal ]]; then
    cowsay -f tux "Has de comer más verdura."
elif [[ $peso -lt $pesIdeal ]]; then
    cowsay -f tux "Has de comer más grasas."
else
    cowsay -f tux "Estás en tu peso ideal"
fi
```

-Muestro captura de pantalla ejecutando el script **pesideal.sh**

The screenshot shows a terminal window titled "Terminal" running on an Ubuntu 22.04 LTS system. The window title bar includes the text "Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (falta acabar) [Corriendo] - Oracle VM VirtualBox". The terminal window displays the following session:

```
alfonsodg@alfonsodg:~/scripts$ pesideal.sh 60 170
< Estás en tu peso ideal >
-----
\ \
  . .
 |:./ |
 // \ \
 (( ))
('`')=(``)
alfonsodg@alfonsodg:~/scripts$ pesideal.sh 74 190
< Has de comer más grasas. >
-----
\ \
  . .
 |:./ |
 // \ \
 (( ))
('`')=(``)
alfonsodg@alfonsodg:~/scripts$ pesideal.sh 80 173
< Has de comer más verdura. >
-----
\ \
  . .
 |:./ |
 // \ \
 (( ))
('`')=(``')
```

The script outputs a ASCII art representation of a person's face, which changes based on the input parameters. The first run (60, 170) results in a neutral face. The second run (74, 190) has a slightly larger nose and mouth, indicating more fats. The third run (80, 173) has a very large nose and mouth, indicating more vegetables.

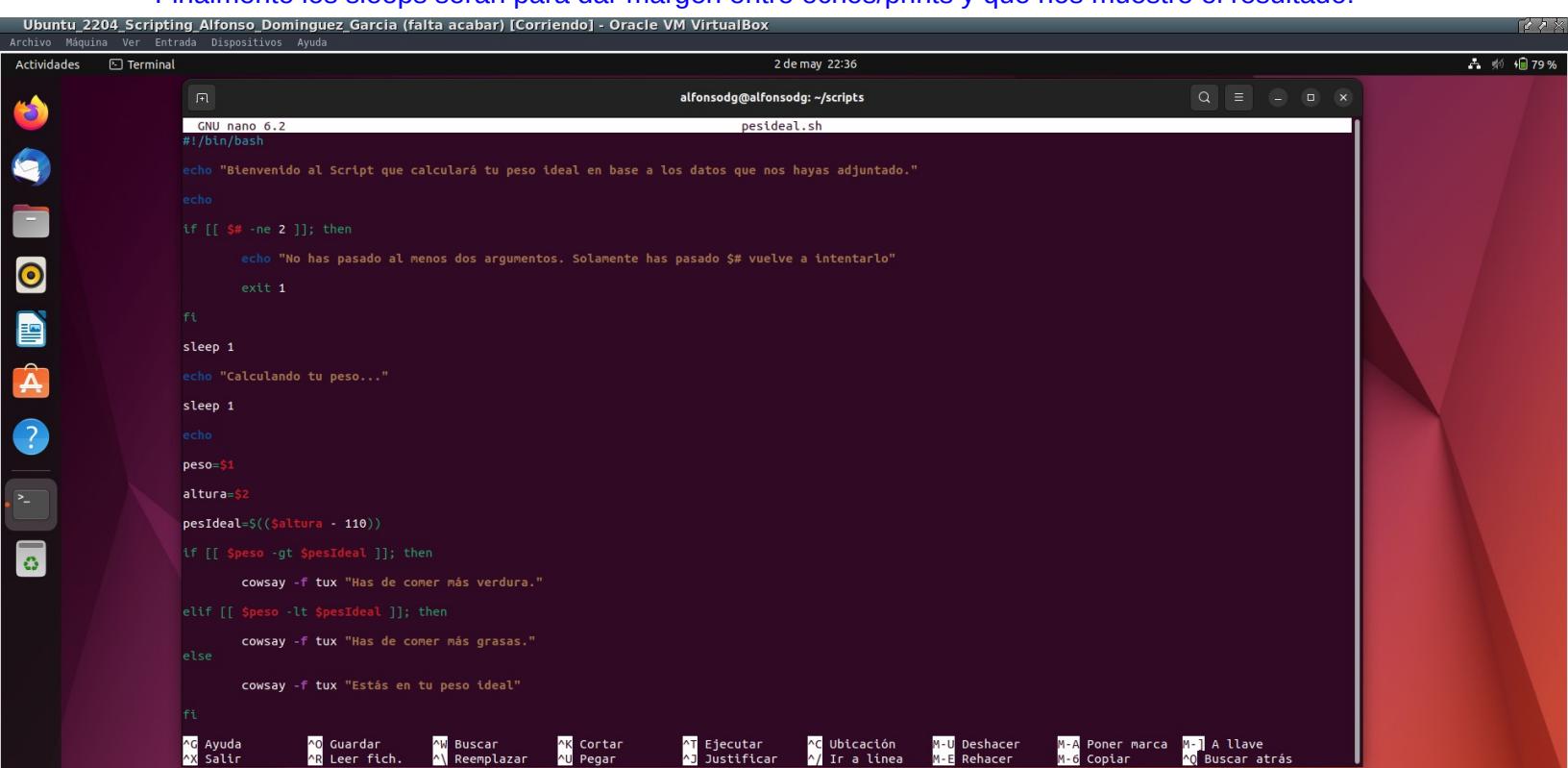
b)

-Ahora hemos añadido un par de líneas más que indican primero la bienvenida al script y un echo para dejar un espacio vacío.

-El siguiente y último paso para mejorar el script **pesideal.sh** será crear un condicional que diga si el número de argumentos que el usuario ha pasado es **Not Equal** o bien **-ne a 2** pues el script no continuará hasta que el usuario haya establecido 2 argumentos previos y después calcularemos el peso ideal.

-He añadido un exit 1 para depurar el error en caso de que a lo largo del script haya más de un error.

-Finalmente los sleeps serán para dar margen entre echos/prints y que nos muestre el resultado:



```
GNU nano 6.2                                         alfonasd@alfonsodg: ~/scripts
#!/bin/bash

echo "Bienvenido al Script que calculará tu peso ideal en base a los datos que nos hayas adjuntado."
echo

if [[ $# -ne 2 ]]; then
    echo "No has pasado al menos dos argumentos. Solamente has pasado $# vuelve a intentarlo"
    exit 1
fi

sleep 1
echo "Calculando tu peso..."
sleep 1
echo

peso=$1
altura=$2
pesIdeal=$((altura - 110))

if [[ $peso -gt $pesIdeal ]]; then
    cowsay -f tux "Has de comer más verdura."
elif [[ $peso -lt $pesIdeal ]]; then
    cowsay -f tux "Has de comer más grasas."
else
    cowsay -f tux "Estás en tu peso ideal"
fi
```

-Ahora mostraré captura de pantalla con la ejecución con errores del script, por ejemplo he pasado el script de **pesideal.sh** con un solo argumento y vemos que en echo devuelve el resultado de forma que indica si ha pasado o no el número de argumentos necesarios

-Muestro captura de pantalla con la ejecución final:

The screenshot shows a terminal window titled "alfonsodg@alfonsodg: ~/scripts" running on an Ubuntu 22.04 desktop. The terminal output is as follows:

```
alfonsodg@alfonsodg:~/scripts$ pesideal.sh 88
Bienvenido al Script que calculará tu peso ideal en base a los datos que nos hayas adjuntado.

No has pasado al menos dos argumentos. Solamente has pasado 1 vuelve a intentarlo
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ pesideal.sh
Bienvenido al Script que calculará tu peso ideal en base a los datos que nos hayas adjuntado.

No has pasado al menos dos argumentos. Solamente has pasado 0 vuelve a intentarlo
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ pesideal.sh 88 173
Bienvenido al Script que calculará tu peso ideal en base a los datos que nos hayas adjuntado.

Calculando tu peso...

< Has de comer más verdura. >
-----
 \
  ...
   |o_o |
   // \ \
   ({    })
  ('`' `')
  ('`' `')
alfonsodg@alfonsodg:~/scripts$
```

ACT. 9

a)

-Crearemos el script **dom.sh** que permitirá comprobar al usuario si el mes y año actuales son de traspaso o no.

-El cálculo para saber si es un año de traspaso es el siguiente → **año es divisible entre 4 == 0 y el año es divisible entre 100 que es == 0 o bien si el año es divisible por 400 y es == 0 entonces será una año de traspaso. De lo contrario no.**

-Comprobaremos también si febrero es un mes de traspaso en base al cálculo del año anterior si "febrero" ese año tiene 28 o 29 días siendo así bisiesto o no

```

Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (ACT10 Finalizada) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 6 de may 21:25
alfonsodg@alfonsodg: ~/scripts
GNU nano 6.2
#!/bin/bash

mes=$(date +%B)
year=$(date +%Y)

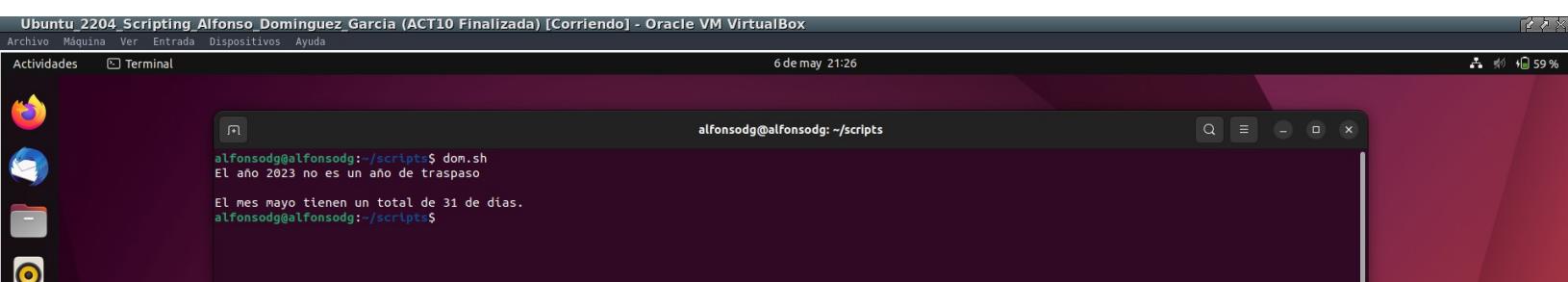
if [ $((year % 4)) -eq 0 -a $((year % 100)) -ne 0 ] || [ $((year % 400)) -eq 0 ]; then
    echo "El año $year es un año de traspaso"
    yearTraspaso=true
else
    echo "El año $year no es un año de traspaso"
    yearTraspaso=false
fi

if [[ $mes -eq "enero" ]] || [[ $mes -eq "marzo" ]] || [[ $mes -eq "mayo" ]] || [[ $mes -eq "julio" ]] || [[ $mes -eq "agosto" ]] || [[ $mes -eq "octubre" ]] || [[ $mes -eq "diciembre" ]]; then
    echo
    echo "El mes $mes tienen un total de 31 de días."
elif [[ $mes -eq "abril" ]] || [[ $mes -eq "junio" ]] || [[ $mes == "septiembre" ]] || [[ $mes -eq "noviembre" ]]; then
    echo
    echo "El mes $mes tiene un total de 30 días."
elif [[ $mes -eq "febrero" ]]; then
    if [[ $yearTraspaso == true ]]; then
        echo "El mes $mes tendrá un total de 29 días."
    else
        echo "El mes $mes tendrá un total de 28 días."
    fi
fi

```

^G Ayuda ^Q Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación ^U Deshacer ^A Poner marca ^L A llave ^Q Anterior ^B Atrás ^R Leer fich. ^R Reemplazar ^U Pegar ^D Justificar ^I Ir a línea ^E Rehacer ^G Copiar ^O Buscar atrás ^W Siguiente ^F Adelante ^S Palabra ant ^S Palabra Siguiente

-Si ejecutamos el script **dom.sh** muestro a continuación su resultado.



The screenshot shows a terminal window titled "Terminal" running on an Ubuntu 22.04 desktop. The window title bar says "Ubuntu 22.04 Scripting Alfonso_Dominguez_Garcia (ACT10 Finalizada) [Corriendo] - Oracle VM VirtualBox". The terminal window displays the following output:

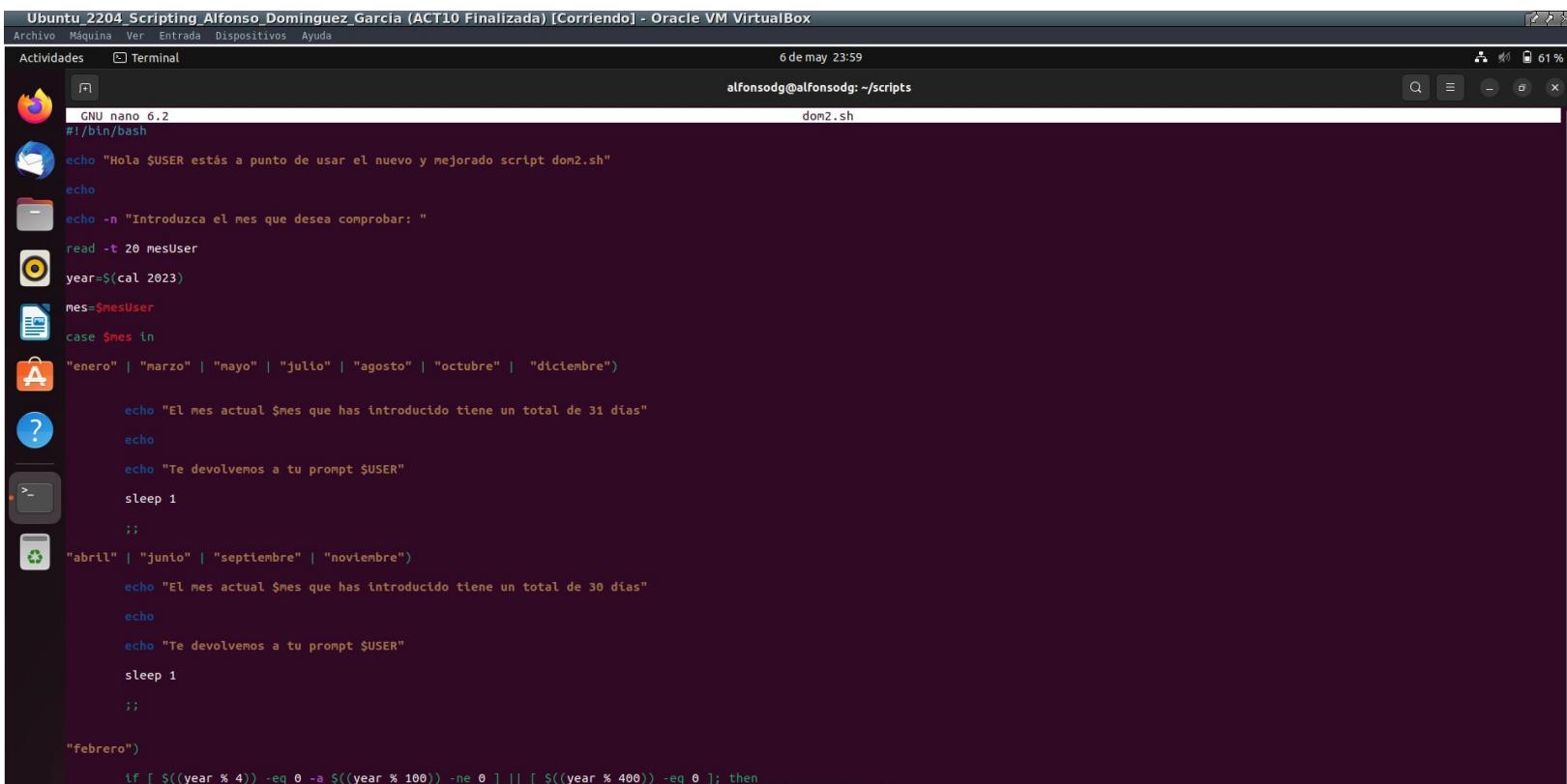
```
alfonsodg@alfonsodg:~/scripts$ dom.sh
El año 2023 no es un año de traspaso
El mes mayo tienen un total de 31 de días.
alfonsodg@alfonsodg:~/scripts$
```

b)

-Muestro capturas de pantalla conforme se ha creado el script mejorado "dom2.sh" que comprobará si el usuario ha introducido un mes y usando la función cal del año 2023 debería de devolvernos los meses que el usuario quiera dentro del año actual (en ningún momento se ha usado la variable date):

-He usado un case con OR ya que si este mes o este mes o este otro mes tienen 31 días o bien tienen 30 días y devolvemos al prompt del usuario.

-En el último punto del script debería de realizar un condicional que muestre si el año es divisible por 4 y por 100 o bien es divisible por 400 entonces dirá si el año es bisiesto o no.



```
GNU nano 6.2
#!/bin/bash
echo "Hola $USER estás a punto de usar el nuevo y mejorado script dom2.sh"
echo
echo -n "Introduzca el mes que desea comprobar: "
read -t 20 mesUser
year=$(cal 2023)
mes=$mesUser
case $mes in
"enero" | "marzo" | "mayo" | "julio" | "agosto" | "octubre" | "diciembre")
    echo "El mes actual $mes que has introducido tiene un total de 31 días"
    echo
    echo "Te devolvemos a tu prompt $USER"
    sleep 1
;;
"abril" | "junio" | "septiembre" | "noviembre")
    echo "El mes actual $mes que has introducido tiene un total de 30 días"
    echo
    echo "Te devolvemos a tu prompt $USER"
    sleep 1
;;
"febrero")
    if [ $((year % 4)) -eq 0 -a $((year % 100)) -ne 0 ] || [ $((year % 400)) -eq 0 ]; then
        echo "El año actual $year es un año bisiesto"
    else
        echo "El año actual $year no es un año bisiesto"
    fi
    echo
    echo "Te devolvemos a tu prompt $USER"
    sleep 1
;;
*)
    echo "El mes $mes que has introducido no existe"
    echo
    echo "Te devolvemos a tu prompt $USER"
    sleep 1
;;
esac
```

```
        echo "El mes actual $mes tiene un total de 29 días, es bisiesto"
else
        echo "El mes actual $mes tiene un total de 28 días"
fi
;;
*)
echo "El mes $mes introducido no existe o no es correcto"
echo
echo "Saliendo del programa..."
exit 2
;;
esac
```

-Muestro la ejecución del script dom2.sh y conforme devolverá para introducir el mes que el usuario desea comprobar.

Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (ACT10 Finalizada) [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal 7 de may 00:00

alfonsodg@alfonsodg:~/scripts\$ dom2.sh

Hola alfonsodg estás a punto de usar el nuevo y mejorado script dom2.sh

Introduzca el mes que desea comprobar: diciembre

El mes actual diciembre que has introducido tiene un total de 31 días

Te devolvemos a tu prompt alfonsodg

alfonsodg@alfonsodg:~/scripts\$

alfonsodg@alfonsodg:~/scripts\$

alfonsodg@alfonsodg:~/scripts\$ dom2.sh

Hola alfonsodg estás a punto de usar el nuevo y mejorado script dom2.sh

Introduzca el mes que desea comprobar: marzo

El mes actual marzo que has introducido tiene un total de 31 días

Te devolvemos a tu prompt alfonsodg

alfonsodg@alfonsodg:~/scripts\$

alfonsodg@alfonsodg:~/scripts\$ dom2.sh abril

Hola alfonsodg estás a punto de usar el nuevo y mejorado script dom2.sh

Introduzca el mes que desea comprobar: abril

El mes actual abril que has introducido tiene un total de 30 días

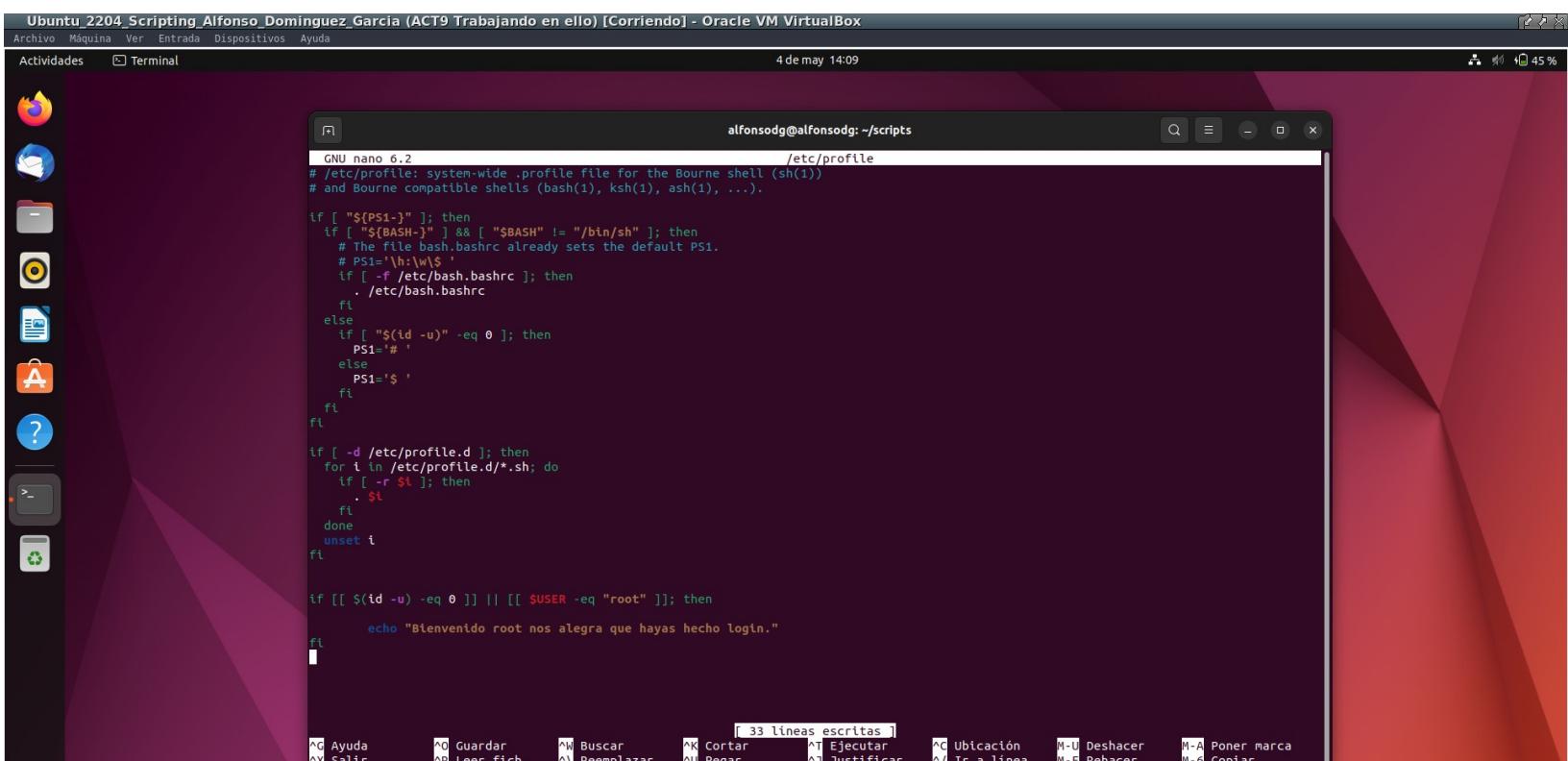
Te devolvemos a tu prompt alfonsodg

alfonsodg@alfonsodg:~/scripts\$

c)

-En esta captura de pantalla muestro como he añadido una línea en el fichero **/etc/profile** que será un condicional que si se cumple una opción u otra usando el operador lógico OR comprobar si el usuario con **id -eq 0** es decir que si el usuario es 0 en este caso root debería de devolver el mensaje de root.

-O si **\$USER** que es el usuario actual es igual a "root" pues debería de devolver el mensaje de bienvenida



```

GNU nano 6.2                               /etc/profile
#!/etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "${PS1-}" ]; then
    if [ "$BASH-" ] && [ "$BASH" != "/bin/sh" ]; then
        # The file bash.bashrc already sets the default PS1.
        # PS1='\$ '
        if [ -f /etc/bash.bashrc ]; then
            . /etc/bash.bashrc
        fi
    else
        if [ "$(id -u)" -eq 0 ]; then
            PS1="# "
        else
            PS1='$ '
        fi
    fi
fi

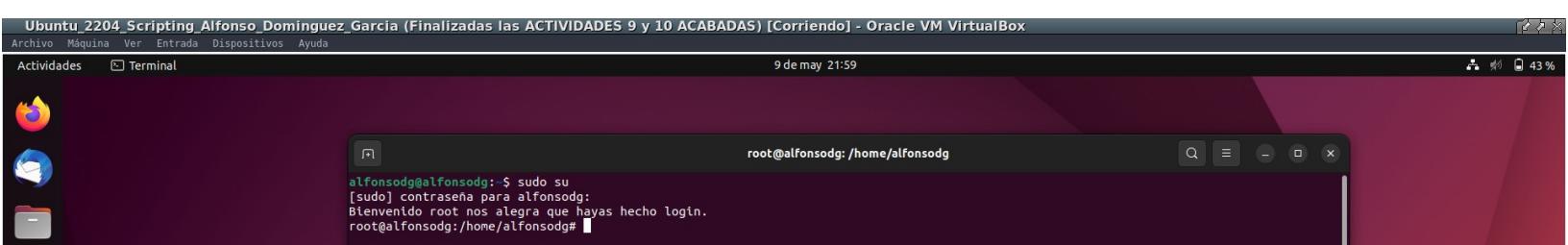
if [ -d /etc/profile.d ]; then
    for i in /etc/profile.d/*.sh; do
        if [ -r $i ]; then
            . $i
        fi
    done
    unset i
fi

if [[ $(id -u) -eq 0 ]] || [[ $USER -eq "root" ]]; then
    echo "Bienvenido root nos alegra que hayas hecho login."
fi

```

The terminal window shows the /etc/profile file being edited in nano. The file contains a conditional block to check if the user is root or if \$USER is 'root'. If either condition is true, it prints the message "Bienvenido root nos alegra que hayas hecho login.". The terminal status bar indicates 33 lines written.

-Muestro captura de pantalla del mensaje que printará por pantalla al realizar un login como usuario **root**.



```

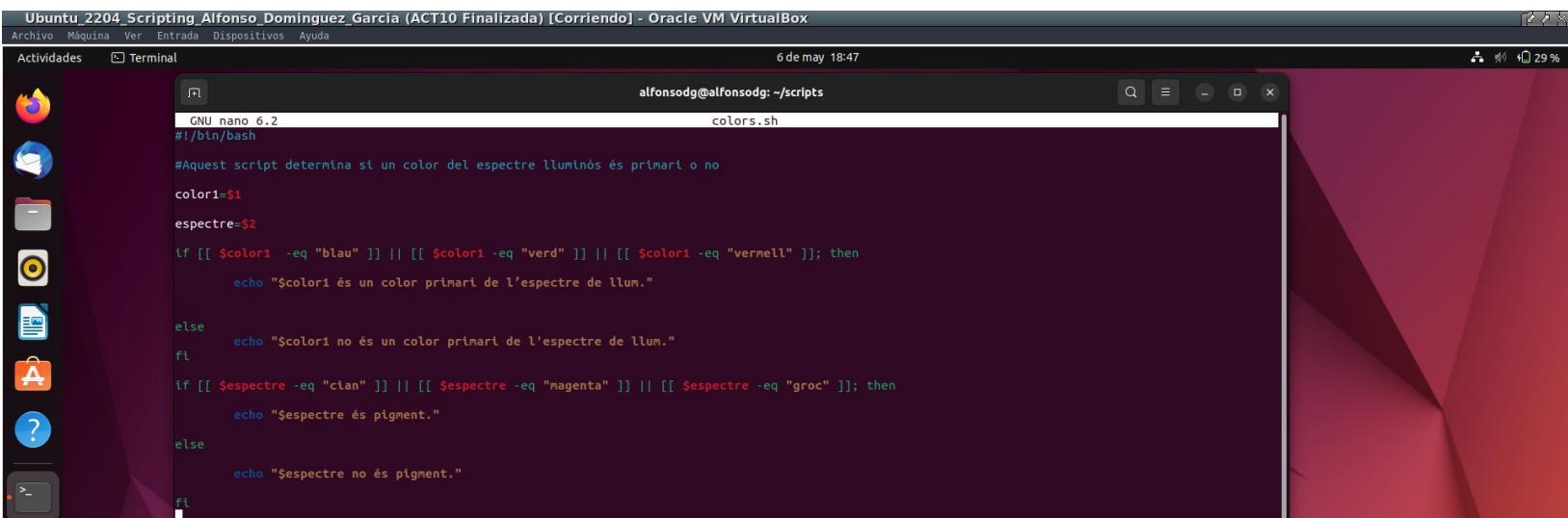
root@alfonsodg: ~
root@alfonsodg: ~$ sudo su
[sudo] contraseña para alfonsodg:
Bienvenido root nos alegra que hayas hecho login.
root@alfonsodg: ~

```

The terminal window shows the user logging in as root. The password is entered, followed by the message "Bienvenido root nos alegra que hayas hecho login." The terminal status bar indicates 9 de may 21:59.

d)

-En este script **colors.sh** lo que realizará es una comprobación de si los valores son “**blau”, “verd”, “vermell”** que será el argumento \$1 pues serán colores primarios del espectro de luz y según **\$2** que será el segundo argumento a insertar pues debería de devolver si es un color del espectro pigmento:



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "colors.sh" and the path is "alfonsodg@alfonsodg: ~/scripts". The terminal content displays a Bash script named "colors.sh" written in Spanish. The script checks if the first argument (\$1) is one of the primary colors ("blau", "verd", or "vermell"). If it is, it prints that it's a primary color of light. If not, it checks if the second argument (\$2) is one of the primary colors of pigments ("cian", "magenta", or "groc"). If it is, it prints that it's a pigment color. Otherwise, it prints that it's not a primary color.

```
GNU nano 6.2                                         colors.sh
#!/bin/bash

#Aquest script determina si un color del espectre lluminòs és primari o no

color1=$1
espectre=$2

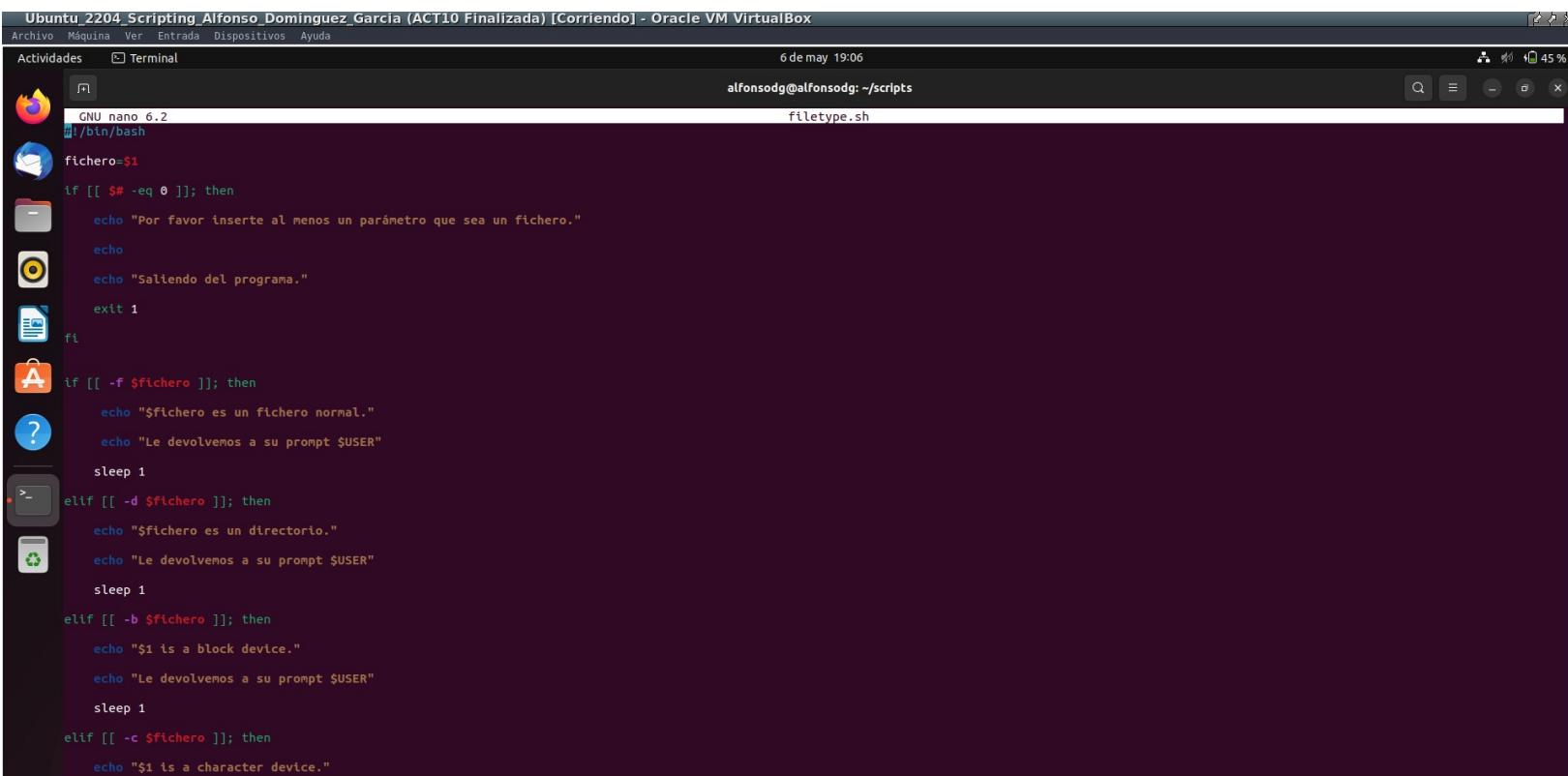
if [[ $color1 == "blau" ]] || [[ $color1 == "verd" ]] || [[ $color1 == "vermell" ]]; then
    echo "$color1 és un color primari de l'espectre de llum."
else
    echo "$color1 no és un color primari de l'espectre de llum."
fi

if [[ $espectre == "cian" ]] || [[ $espectre == "magenta" ]] || [[ $espectre == "groc" ]]; then
    echo "$espectre és pigment."
else
    echo "$espectre no és pigment."
fi
```

e)

-Aquí muestro captura de pantalla del script “**filetype.sh**” que consta en un fichero script que tiene una estructura de condicionales if – elif -else. Las cosas más importantes a destacar sería que en cada condicional se usan las opciones.

-f (saber si el fichero es normal), -d (comprobar si un fichero es un directorio), -c (Comprobar si el fichero o argumento pasado es un carácter).



The screenshot shows a terminal window titled "Ubuntu 22.04 Scripting Alfonso_Dominguez_Garcia (ACT10 Finalizada) [Corriendo] - Oracle VM VirtualBox". The terminal is running the command "filetype.sh". The script content is as follows:

```
GNU nano 6.2
#!/bin/bash

fichero=$1

if [[ $# -eq 0 ]]; then
    echo "Por favor inserte al menos un parámetro que sea un fichero."
    echo
    echo "Saliendo del programa."
    exit 1
fi

if [[ -f $fichero ]]; then
    echo "$fichero es un fichero normal."
    echo "Le devolvemos a su prompt $USER"
    sleep 1
elif [[ -d $fichero ]]; then
    echo "$fichero es un directorio."
    echo "Le devolvemos a su prompt $USER"
    sleep 1
elif [[ -b $fichero ]]; then
    echo "$1 is a block device."
    echo "Le devolvemos a su prompt $USER"
    sleep 1
elif [[ -c $fichero ]]; then
    echo "$1 is a character device."
```

-Muestro la segunda captura de pantalla con parte del código corregido y el final de ejecución:

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "filetype.sh". The terminal content is a shell script written in GNU nano 6.2. The script uses conditional statements (if-elif-else) to determine the type of a file based on its permissions (-c, -S, -p, -L). It includes sleep commands and echo statements to provide feedback to the user. The terminal window is part of a desktop interface with icons for various applications like a browser, file manager, and system tools.

```
GNU nano 6.2
sleep 1
elif [[ -c $fichero ]]; then
    echo "$1 is a character device."
    echo "Le devolvemos a su prompt $USER"
    sleep 1
elif [[ -S $fichero ]]; then
    echo "El fichero $fichero es un SOCKET"
    echo "Le devolvemos a su prompt $USER"
    sleep 1
elif [[ -p $fichero ]]; then
    echo "El fichero $fichero es una PIPE."
    echo "Le devolvemos a su prompt $USER"
    sleep 1
elif [[ -L $fichero ]]; then
    echo "El fichero $fichero es un enlace simbólico."
    echo "Le devolvemos a su prompt $USER"
    sleep 1
else
    echo "El tipo de fichero $fichero no pudo ser determinado, de esta forma cerramos el programa."
    exit 2
fi
```

-Y ahora ejecutamos el script y pasamos como argumento la ruta de un directorio, un fichero o comprobar si es una PIPE. Para ello lo que se debe hacer es crear primero ejecutando el mkfifo PIPE y lo creará como documento. Luego lo pasamos como argumento en el script “**filetype.sh**” y funcionará correctamente y debería devolver que el argumento es una PIPE:

```
Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (ACT10 Finalizada) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 6 de may 19:33
alfonsodg@alfonsodg:/$ filetype.sh /home
El fichero /home es un directorio.

Le devolvemos a su prompt alfonsodg
alfonsodg@alfonsodg:/$ filetype.sh /home/alfonsodg/scripts/fichero1.txt
El fichero /home/alfonsodg/scripts/fichero1.txt es un fichero normal.

Le devolvemos a su prompt alfonsodg
alfonsodg@alfonsodg:/$ mkfifo PIPE
mkfifo: no se puede crear el fichero 'fifo' 'PIPE': El archivo ya existe
alfonsodg@alfonsodg:/$ filetype.sh PIPE
El fichero PIPE es una PIPE.

Le devolvemos a su prompt alfonsodg
alfonsodg@alfonsodg:/$
alfonsodg@alfonsodg:/$
```

-Finalmente debemos de crear el último script llamado “**existeix.sh**” y que contendrá el siguiente código:

```
Ubuntu 2204 Scripting_Alfonso_Dominguez_Garcia (Finalizadas las ACTIVIDADES 9 y 10 ACABADAS) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 9 de may 22:33
alfonsodg@alfonsodg: ~/scripts
GNU nano 6.2 existeix.sh

#!/bin/bash

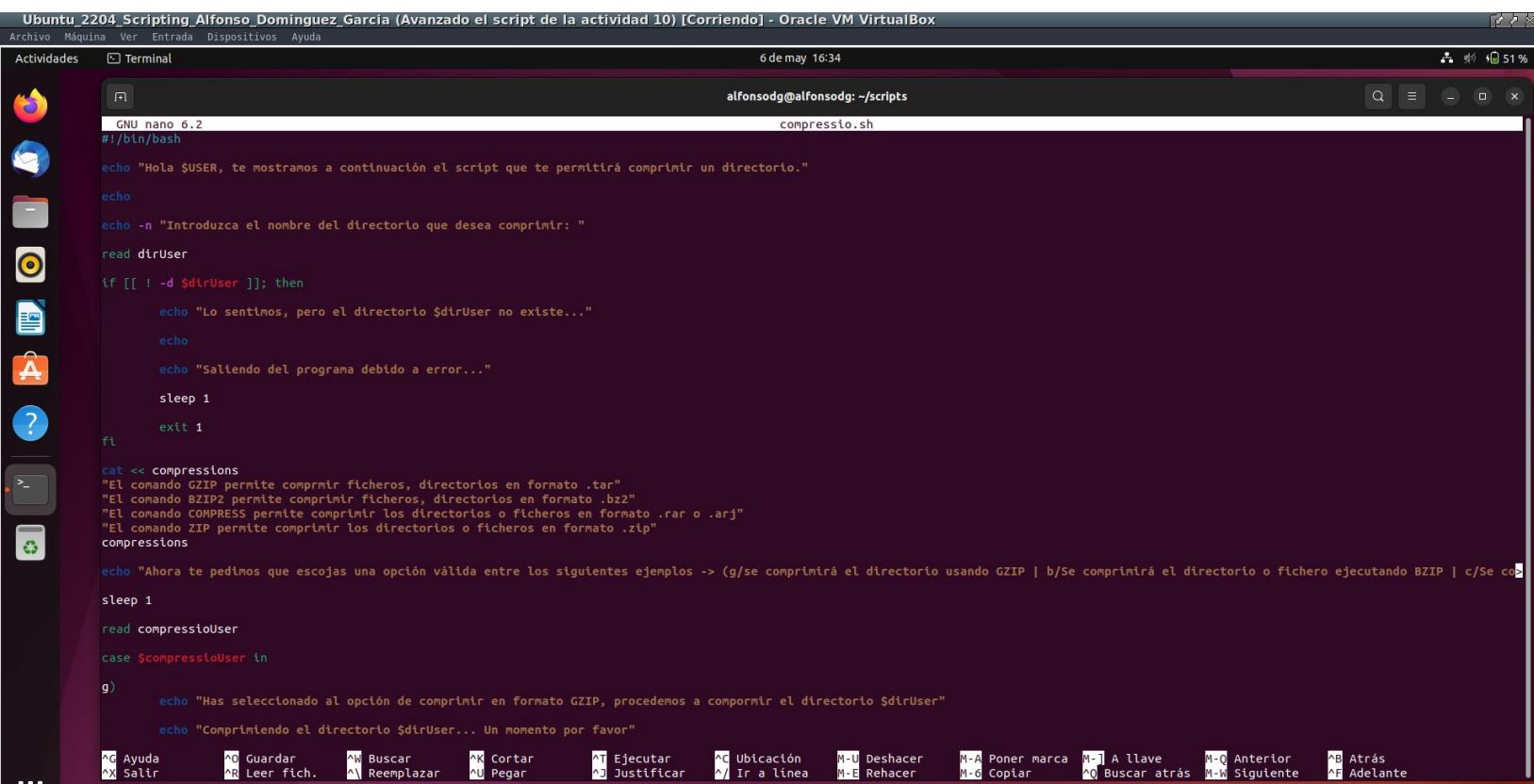
if [[ $# -eq 1 ]]; then
    var="/home/$USER/$1"
    if [[ -e $var ]]; then
        export fichero="$1"
        /home/$USER/scripts/filetype.sh $fichero
    else
        echo "El fichero $1 no existe en este Home Directory"
        echo
        echo "Saliendo del programa..."
        exit 1
    fi
else
    echo "Por favor ha de introducir al menos un parámetro."
    echo
    echo "Saliendo del programa $USER..."
    exit 2
fi
```

ACT. 10

a)

-Primero crearemos el script “compressio.sh” y una vez hecho esto muestro capturas de pantalla con la estructura del mismo script.

-Las cosas más importantes a destacar serían el condicional que permitirá saber si es un directorio y existe dicho directorio que el usuario ha introducido:



```

Ubuntu 22.04, Scripting Alfonso_Dominguez_Garcia (Avanzado el script de la actividad 10) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 6 de may 16:34
alfonsodg@alfonsodg: ~/scripts
GNU nano 6.2
#!/bin/bash

echo "Hola $USER, te mostramos a continuación el script que te permitirá comprimir un directorio."
echo
echo -n "Introduzca el nombre del directorio que desea comprimir: "
read dirUser
if [[ ! -d $dirUser ]]; then
    echo "Lo sentimos, pero el directorio $dirUser no existe..."
    echo
    echo "Saliendo del programa debido a error..."
    sleep 1
    exit 1
fi
cat << compressions
#El comando GZIP permite comprimir ficheros, directorios en formato .tar"
#El comando BZIP2 permite comprimir ficheros, directorios en formato .bz2"
#El comando COMPRESS permite comprimir los directorios o ficheros en formato .rar o .arj"
#El comando ZIP permite comprimir los directorios o ficheros en formato .zip"
compressions
echo "Ahora te pedimos que escojas una opción válida entre los siguientes ejemplos -> (g/se comprimirá el directorio usando GZIP | b/Se comprimirá el directorio o fichero ejecutando BZIP | c/Se co-
sleep 1
read compressioUser
case $compressioUser in
g)
    echo "Has seleccionado al opción de comprimir en formato GZIP, procedemos a compormir el directorio $dirUser"
    echo "Comprimiendo el directorio $dirUser... Un momento por favor"
    ;;
esac

```

Keyboard shortcuts at the bottom:

- Alt-Ayuda
- Alt-Guardar
- Alt-Buscar
- Alt-Cortar
- Alt-Ejecutar
- Alt-Ubicación
- Alt-Deshacer
- Alt-Poner marca
- Alt-A llave
- Alt-Q Anterior
- Alt-B Atrás
- Alt-Salir
- Alt-Leer fich.
- Alt-Reemplazar
- Alt-Pegar
- Alt-Justificar
- Alt-Ir a línea
- Alt-Rehacer
- Alt-Copiar
- Alt-Buscar atrás
- Alt-Siguiente
- Alt-Adelante

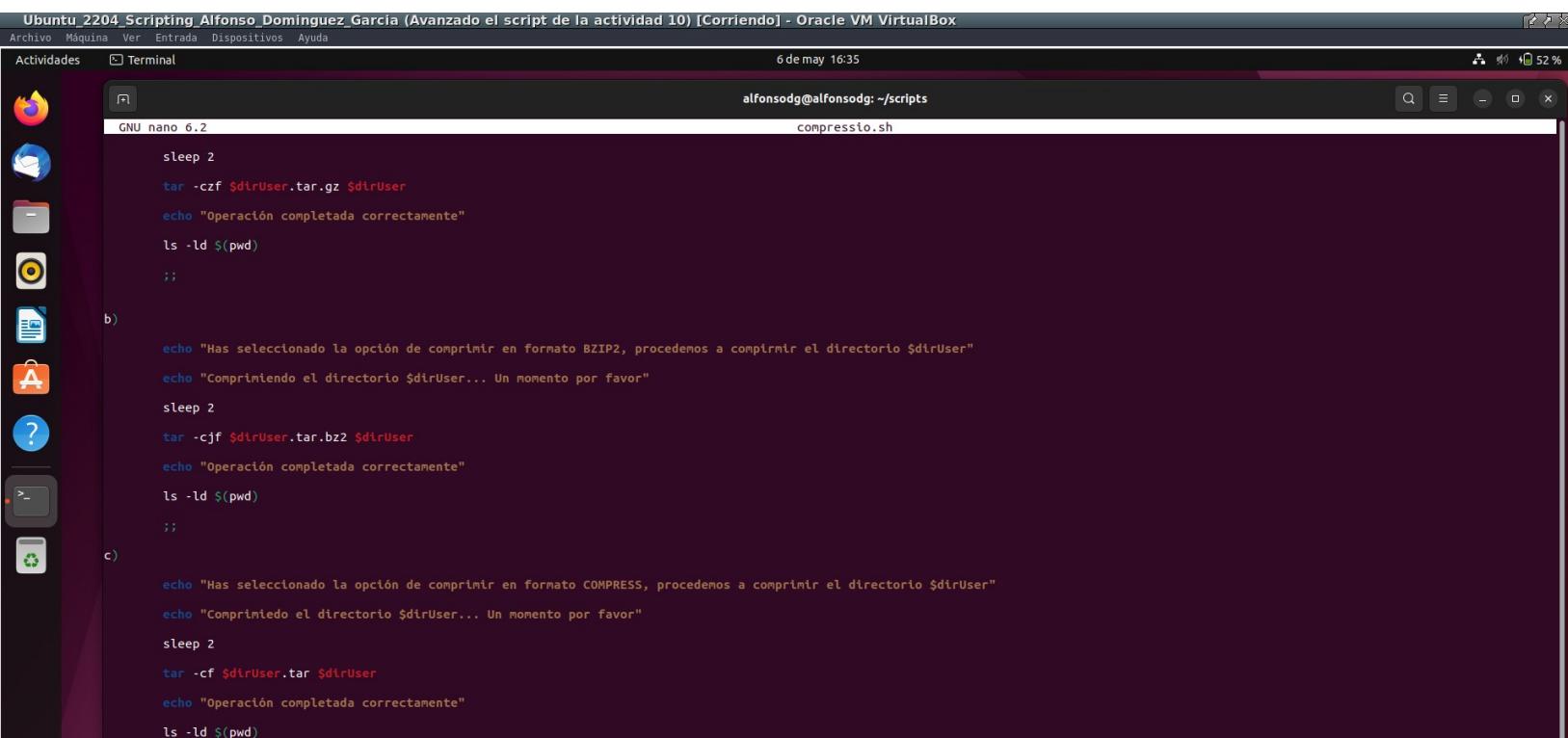
-El siguiente paso a destacar sería el propio case ya que en este caso le pedimos al usuario que seleccione entre las letras, g) b), c), z) que permitirán comprimir el directorio usando las distintas funcionalidades del script. Este permitirá comprimir usando los comandos de compresión siguientes.

tar -czf \$dirUser.tar.bz2 (directorio a comprimir) \$dirUser (de momento guardaremos el fichero en la propia ruta que haya dicho el usuario, más adelante esta variable se cambiará por \$rutaUser que permitirá almacenar la ruta que haya pedido el usuario) → Compresión en BZIP

tar -czf \$dirUser.tar.gz → Compresión en GZIP

tar -cf \$dirUser.tar → Compresión usando COMPRESS

zip -r \$dirUser.zip → Compresión usando ZIP



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "compressio.sh" and it shows a script being run. The script contains code to compress a directory using different methods based on user input. The terminal window is part of a desktop interface with a menu bar at the top and various icons on the left.

```
GNU nano 6.2                                         alfonsgdg@alfonsdg:~/scripts
compressio.sh                                         6 de may 16:35

sleep 2
tar -czf $dirUser.tar.gz $dirUser
echo "Operación completada correctamente"
ls -ld $(pwd)
;;
b)

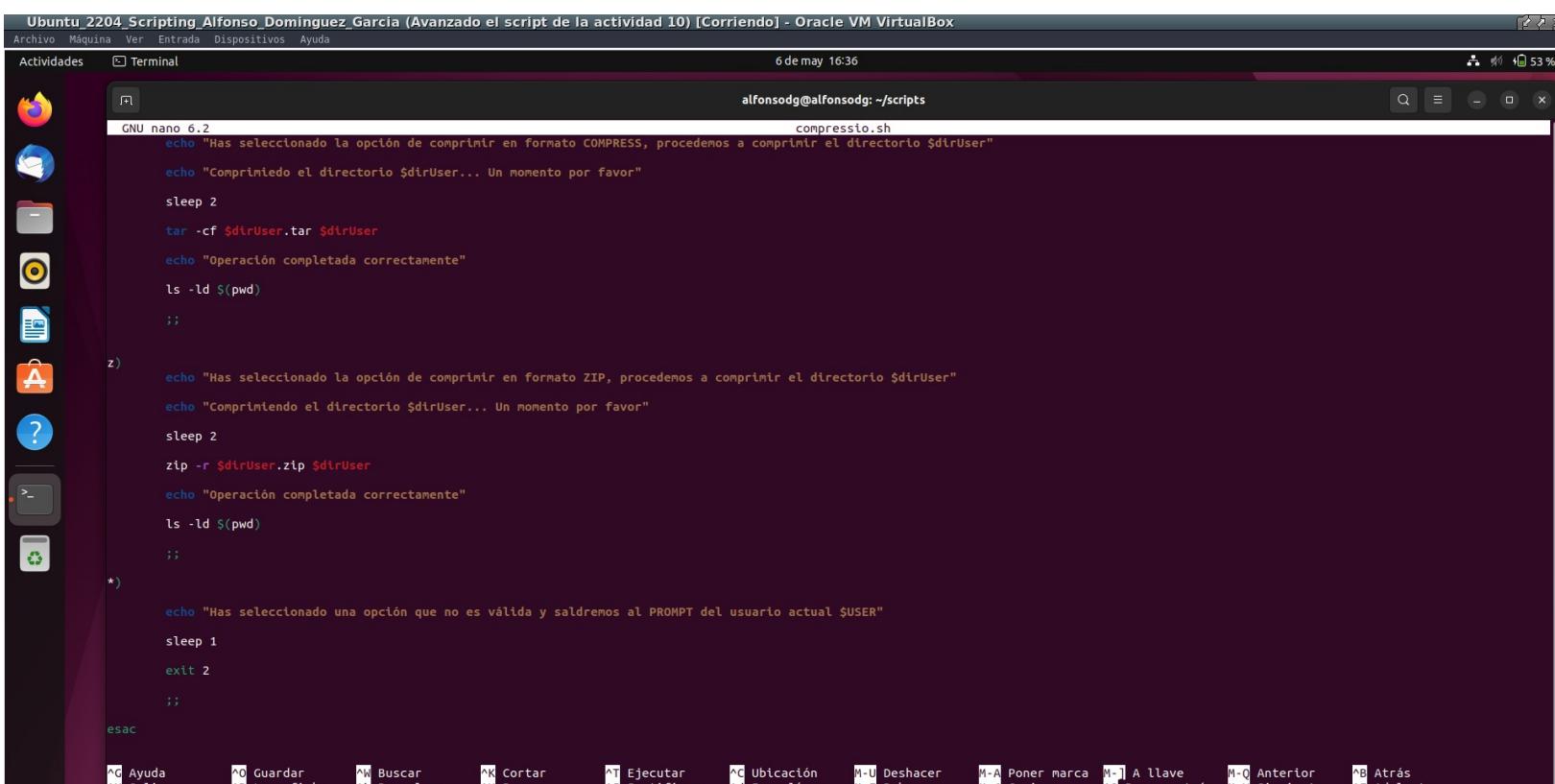
echo "Has seleccionado la opción de comprimir en formato BZIP2, procedemos a comprimir el directorio $dirUser"
echo "Comprimiendo el directorio $dirUser... Un momento por favor"
sleep 2
tar -cjf $dirUser.tar.bz2 $dirUser
echo "Operación completada correctamente"
ls -ld $(pwd)
;;
c)

echo "Has seleccionado la opción de comprimir en formato COMPRESS, procedemos a comprimir el directorio $dirUser"
echo "Comprimiendo el directorio $dirUser... Un momento por favor"
sleep 2
tar -cf $dirUser.tar $dirUser
echo "Operación completada correctamente"
ls -ld $(pwd)
```

-El último punto a destacar sería finalmente el poder mostrar la compresión en el fichero en el directorio actual ejecutando el comando:

ls -l \$(pwd)

-Ejecutaremos también un sleep 1 segundos para dar 1 segundos de margen entre texto y texto:

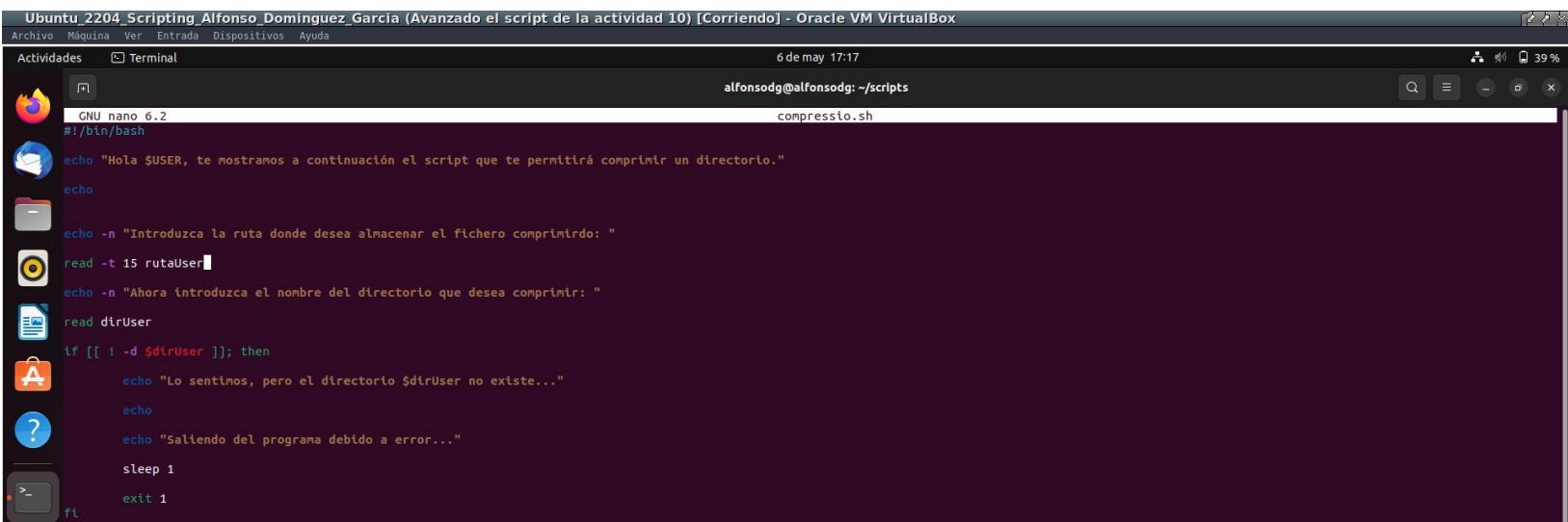


```
GNU nano 6.2
compressio.sh
echo "Has seleccionado la opción de comprimir en formato COMPRESS, procedemos a comprimir el directorio $dirUser"
echo "Comprimiendo el directorio $dirUser... Un momento por favor"
sleep 2
tar -cf $dirUser.tar $dirUser
echo "Operación completada correctamente"
ls -ld $(pwd)
;;
z)
echo "Has seleccionado la opción de comprimir en formato ZIP, procedemos a comprimir el directorio $dirUser"
echo "Comprimiendo el directorio $dirUser... Un momento por favor"
sleep 2
zip -r $dirUser.zip $dirUser
echo "Operación completada correctamente"
ls -ld $(pwd)
;;
*)
echo "Has seleccionado una opción que no es válida y saldremos al PROMPT del usuario actual $USER"
sleep 1
exit 2
;;
esac
```

b)

-Para arreglar este apartado B) debemos de realizar un echo -n y le pediremos al usuario que introduzca la ruta donde desea guardar el fichero comprimido:

-El siguiente paso será crear un variable con “**read -t 15 rutaUser**” \$rutaUser será la variable que almacene la ruta que habrá introducido. **read -t 15** que calculará un total de 15 segundos que tendrá el usuario para establecer la ruta... En caso de que pasen esos 15 segundos no podrá introducir la ruta y saltará a preguntar de insertar el nuevo valor que será el directorio que desean comprimir.



```
GNU nano 6.2
#!/bin/bash
echo "Hola $USER, te mostramos a continuación el script que te permitirá comprimir un directorio."
echo
echo -n "Introduzca la ruta donde desea almacenar el fichero comprimido: "
read -t 15 rutaUser
echo -n "Ahora introduzca el nombre del directorio que desea comprimir: "
read dirUser
if [[ ! -d $dirUser ]]; then
    echo "Lo sentimos, pero el directorio $dirUser no existe..."
    echo
    echo "Saliendo del programa debido a error..."
    sleep 1
    exit 1
fi
```

-En esta captura de pantalla muestro los apartados en los que en el caso guardará la ruta donde se almacenará finalmente mediante el siguiente comando:

sudo tar -cJ \$dirUser.tar.bz2 \$rutaUser

La variable **\$rutaUser** será la ruta donde almacenará el directorio comprimido.

Ubuntu_2204_Scripting_Alfonso_Dominguez_Garcia (Avanzado el script de la actividad 10) [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal 6 de may 17:18

alfonsodg@alfonsodg: ~/scripts

```
GNU nano 6.2
compressio.sh
"El comando BZIP2 permite comprimir ficheros, directorios en formato .bz2"
"El comando COMPRESS permite comprimir los directorios o ficheros en formato .rar o .arj"
"El comando ZIP permite comprimir los directorios o ficheros en formato .zip"
compressions

echo
echo "Ahora te pedimos que escojas una opción válida entre los siguientes ejemplos -> (g/se comprimirá el directorio usando GZIP | b/Se comprimirá el directorio o fichero usando BZIP2 | a/Se comprimirá el directorio o fichero usando ZIP | ?/Ayuda | >/Salida)"

sleep 1
read compressioUser
case $compressioUser in
g)
    echo "Has seleccionado al opción de comprimir en formato GZIP, procedemos a compormir el directorio $dirUser"
    echo "Comprimiendo el directorio $dirUser... Un momento por favor"
    sleep 2
    sudo tar -czf $dirUser.tar.gz $rutaUser
    echo "Operación completada correctamente"
    ls -ld $(pwd)
    ;;
b)
    echo "Has seleccionado la opción de comprimir en formato BZIP2, procedemos a compirmir el directorio $dirUser"
    echo "Comprimiendo el directorio $dirUser... Un momento por favor"
    sleep 2
    sudo tar -cJf $dirUser.tar.bz2 $rutaUser
    echo "Operación completada correctamente"
    ls -ld $(pwd)
    ;;
a)
    echo "Has seleccionado la opción de comprimir en formato ZIP, procedemos a compormir el directorio $dirUser"
    echo "Comprimiendo el directorio $dirUser... Un momento por favor"
    sleep 2
    sudo zip -r $dirUser.zip $dirUser
    echo "Operación completada correctamente"
    ls -ld $(pwd)
    ;;
?)
    echo "Has seleccionado la opción de Ayuda, mostrando las opciones disponibles"
    echo "g/ Se comprimirá el directorio usando GZIP"
    echo "b/ Se comprimirá el directorio o fichero usando BZIP2"
    echo "a/ Se comprimirá el directorio o fichero usando ZIP"
    echo "?/ Ayuda"
    echo ">/ Salida"
    ;;
*) echo "Opción no válida, por favor escoja una opción válida"
esac
```

-Finalmente muestro captura de pantalla conforme al ejecutar el script “**compressio.sh**” le pasamos el directorio donde guardaremos el directorio comprimido en **/home** y comprimiremos el directorio **/home/alfonsodg/scripts**

-En la captura de la parte derecha podemos ver como el fichero “**scripts.zip**” se ha comprimido en formato .zip (todos los formatos están comprobados y funcionan correctamente).

```
Ubuntu_2204.Scripting_Alfonso_Dominguez_Garcia [Avanzado el script de la actividad 10] [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 6 de may 17:20
alfonsodg@alfonsodg: ~/scripts
alfonsodg@alfonsodg: ./scripts.sh
Hola alfonsodg, te mostramos a continuación el script que te permitirá comprimir un directorio.

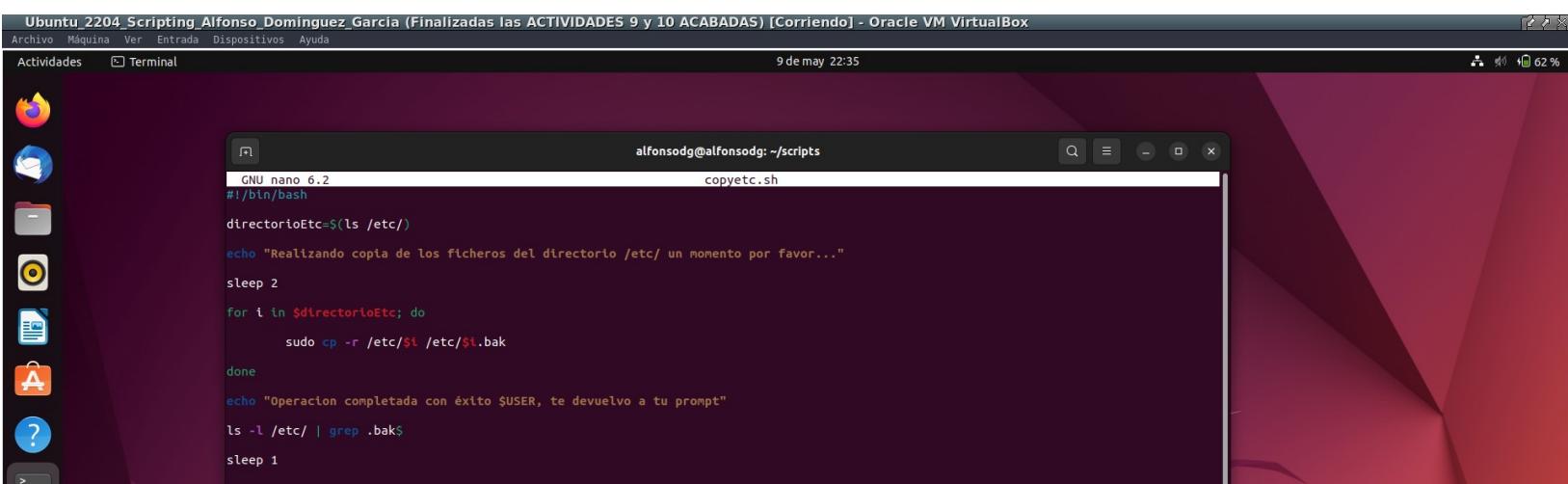
Introduzca la ruta donde desea almacenar el fichero comprimido: /home/alfonsodg/scripts
Ahora introduzca el nombre del directorio que desea comprimir: /home/alfonsodg/scripts
"El comando GZIP permite comprimir ficheros, directorios en formato .tar"
"El comando BZIP2 permite comprimir ficheros, directorios en formato .bz2"
"El comando COMPRESS permite comprimir los directorios o ficheros en formato .rar o .arj"
"El comando ZIP permite comprimir los directorios o ficheros en formato .zip"

Ahora te pedimos que escojas una opción válida entre los siguientes ejemplos -> (g/se comprimirá el d-rw-r--r-- rará el directorio o fichero ejecutando BZIP | c/Se comprimirá en formato COMPRESS y finalmente z/Compr-rw-r--r--)
Has seleccionado la opción de comprimir en formato ZIP, procedemos a comprimir el directorio /home/alfonsodg/scripts... Un momento por favor
[sudo] contraseña para alfonsodg:
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/idrwxr-xr-x 2 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/idrwxr-xr-x 2 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/idrwxr-xr-x 3 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/i-rw----- 1 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/idrwx----- 3 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/idrwxr-xr-x 2 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/idrwxr-xr-x 2 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/l-rw-r--r-- 1 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/ldrwxr-xr-x 2 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/ldrwxr-xr-x 2 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/l-drwxr-xr-x 1 root root
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/l-drwxr-xr-x 3 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/l-drwxr-xr-x 2 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/i-rw----- 1 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/g-rw----- 1 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/g-rw----- 1 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/common/.cache/g-rw----- 2 alfonsodg alfonsodg
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/83/.local/share
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/83/.themes
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/83/.config/gtk-
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/83/.config/gtk-
zip warning: name not matched: /home/alfonsodg/snap/snapd-desktop-integration/83/.config/gtk-
alfonsodg@alfonsodg:~/scripts$ ll
total 3172
drwxr-x--- 18 alfonsodg alfonsodg 4096 may 6 17:18 .
drwxr-xr-x 3 root root 4096 abr 22 21:05 ..
-rw----- 1 alfonsodg alfonsodg 16678 may 5 11:19 .bash_history
-rw----- 1 alfonsodg alfonsodg 220 abr 22 21:05 .bash_logout
drwx----- 13 alfonsodg alfonsodg 3802 abr 24 21:40 .bashrc
drwx----- 13 alfonsodg alfonsodg 4096 abr 24 11:28 .cache/
drwx----- 14 alfonsodg alfonsodg 4096 abr 27 13:15 classes/
drwx----- 2 alfonsodg alfonsodg 4096 abr 27 23:56 .config/
drwx----- 2 alfonsodg alfonsodg 4096 abr 27 22:39 Descargas/
drwx----- 2 alfonsodg alfonsodg 4096 abr 22 22:39 Documentos/
drwx----- 2 alfonsodg alfonsodg 4096 abr 22 22:39 Escritorio/
drwx----- 2 alfonsodg alfonsodg 4096 abr 24 11:36 .gnupg/
drwx----- 2 alfonsodg alfonsodg 4096 abr 24 11:28 Imágenes/
drwx----- 2 alfonsodg alfonsodg 20 may 2 09:32 .lessht
drwx----- 3 alfonsodg alfonsodg 4096 abr 22 22:39 .local/
drwx----- 3 alfonsodg alfonsodg 4096 abr 22 22:39 Música/
drwx----- 2 alfonsodg alfonsodg 4096 abr 22 22:39 Plantillas/
drwx----- 2 alfonsodg alfonsodg 807 abr 22 21:05 .profile
drwx----- 2 alfonsodg alfonsodg 4096 abr 22 22:39 Pública/
drwx----- 2 alfonsodg alfonsodg 4096 abr 5 17:18 scripts/
drwx----- 2 alfonsodg alfonsodg 4096 abr 22 22:39 snar/
drwx----- 2 alfonsodg alfonsodg 4096 abr 24 11:36 .ssh/
drwx----- 1 alfonsodg alfonsodg 0 abr 22 22:47 .sudo_as_admin_successful
drwx----- 1 alfonsodg alfonsodg 5 may 6 16:59 .vboxclient-clipboard.pid
drwx----- 1 alfonsodg alfonsodg 5 may 6 16:59 .vboxclient-draganddrop.pid
drwx----- 1 alfonsodg alfonsodg 5 may 6 16:59 .vboxclient-seamless.pid
drwx----- 1 alfonsodg alfonsodg 5 may 6 16:59 .vboxclient-vmsvga-session-tty2.pid
drwx----- 1 alfonsodg alfonsodg 4096 abr 22 22:39 Videos/
alfonsodg@alfonsodg:~/scripts$
```

ACT. 11

a)

-Lo primero que haremos será mostrar el contenido de /etc/ y lo guardamos en la variable directorioEtc y al realizar esto lo que haremos será hacer la copia de seguridad recorriendo todos los ficheros de la variable “\$directorioEtc” mediante sudo cp -r /etc/\$i /etc/\$i.bak y luego demostraremos que se ha realizado la copia de seguridad ejecutando finalmente el comando “ls -l /etc/ | grep .bak\$” (*ls -l /etc/ podría substituirse por la variable \$directorioEtc {he tenido un pequeño descuido en esta sentencia}*) para encontrar los ficheros que finalizan en .bak



```
GNU nano 6.2                               copyetc.sh
#!/bin/bash

directorioEtc=$(ls /etc/)

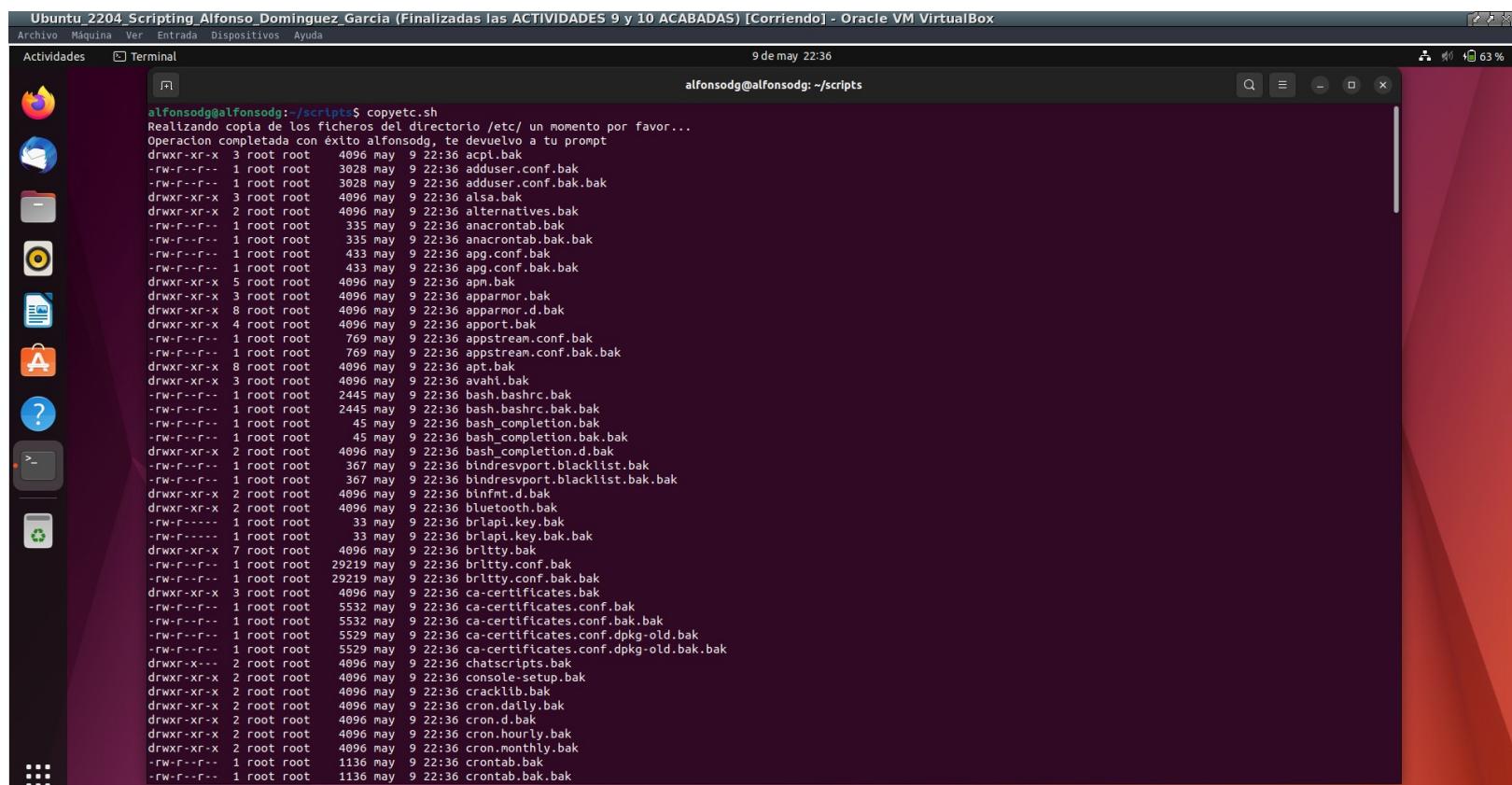
echo "Realizando copia de los ficheros del directorio /etc/ un momento por favor..."

sleep 2

for i in $directorioEtc; do
    sudo cp -r /etc/$i /etc/$i.bak
done

echo "Operacion completada con éxito $USER, te devuelvo a tu prompt"
ls -l /etc/ | grep .bak$ 
sleep 1
```

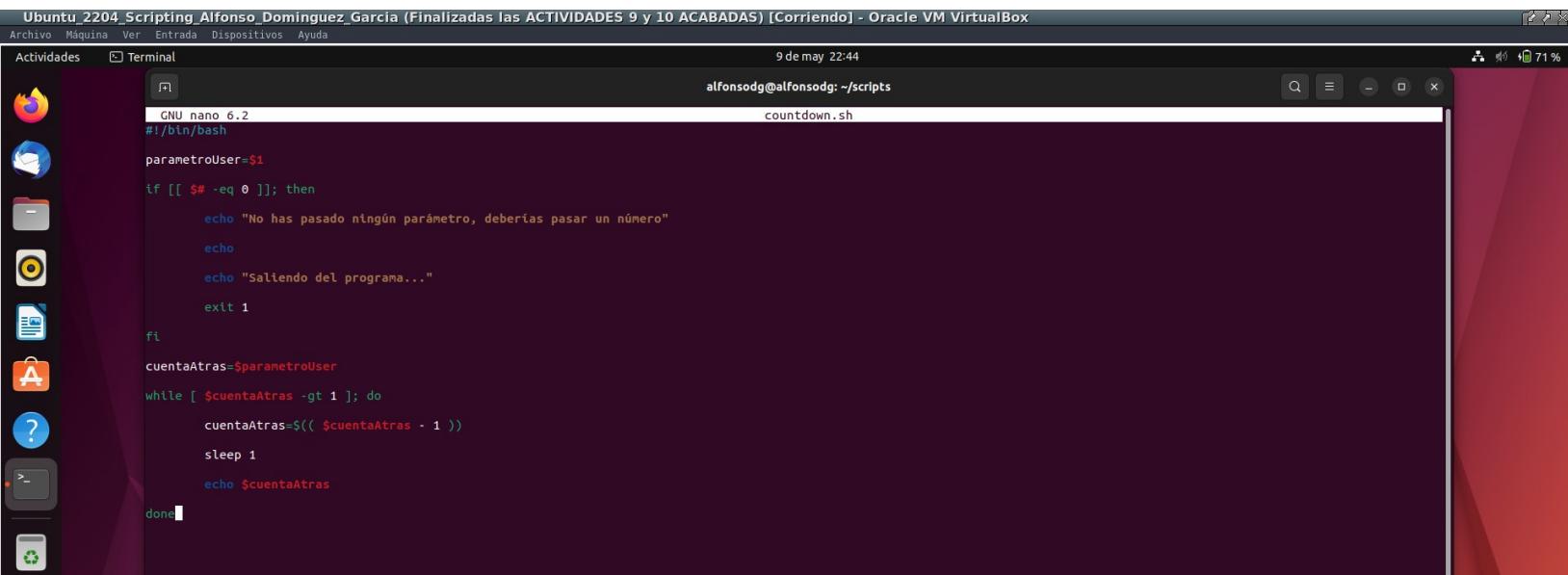
-La siguiente captura de pantalla mostrará el resultado conforme se ha realizado la copia de seguridad de todos los ficheros que acaben en .bak



```
alfonsodg@alfonsodg:~/scripts$ copyetc.sh
Realizando copia de los ficheros del directorio /etc/ un momento por favor...
Operacion completada con éxito alfonsodg, te devuelvo a tu prompt
drwxr-xr-x 3 root root 4096 may 9 22:36 acpi.bak
-rw-r--r-- 1 root root 3028 may 9 22:36 adduser.conf.bak
-rw-r--r-- 1 root root 3028 may 9 22:36 adduser.conf.bak.bak
drwxr-xr-x 3 root root 4096 may 9 22:36 alsa.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 alternatives.bak
-rw-r--r-- 1 root root 335 may 9 22:36 anacrontab.bak
-rw-r--r-- 1 root root 335 may 9 22:36 anacrontab.bak.bak
-rw-r--r-- 1 root root 433 may 9 22:36 apg.conf.bak
-rw-r--r-- 1 root root 433 may 9 22:36 apg.conf.bak.bak
drwxr-xr-x 5 root root 4096 may 9 22:36 apt.bak
drwxr-xr-x 3 root root 4096 may 9 22:36 apparmor.bak
drwxr-xr-x 8 root root 4096 may 9 22:36 apparmor.d.bak
drwxr-xr-x 4 root root 4096 may 9 22:36 apport.bak
-rw-r--r-- 1 root root 769 may 9 22:36 appstream.conf.bak
-rw-r--r-- 1 root root 769 may 9 22:36 appstream.conf.bak.bak
drwxr-xr-x 8 root root 4096 may 9 22:36 apt.bak
drwxr-xr-x 3 root root 4096 may 9 22:36 avahi.bak
drwxr-xr-x 1 root root 2445 may 9 22:36 bash.bashrc.bak
-rw-r--r-- 1 root root 2445 may 9 22:36 bash.bashrc.bak.bak
-rw-r--r-- 1 root root 45 may 9 22:36 bash_completion.bak
-rw-r--r-- 1 root root 45 may 9 22:36 bash_completion.bak.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 bash_completion.d.bak
-rw-r--r-- 1 root root 367 may 9 22:36 bindresvport.blacklist.bak
-rw-r--r-- 1 root root 367 may 9 22:36 bindresvport.blacklist.bak.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 binfmt.d.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 bluetooth.bak
-rw-r----- 1 root root 33 may 9 22:36 brlapi.key.bak
-rw-r--r-- 1 root root 33 may 9 22:36 brlapi.key.bak.bak
drwxr-xr-x 7 root root 4096 may 9 22:36 brltty.bak
-rw-r--r-- 1 root root 29219 may 9 22:36 brltty.conf.bak
-rw-r--r-- 1 root root 29219 may 9 22:36 brltty.conf.bak.bak
drwxr-xr-x 3 root root 4096 may 9 22:36 ca-certificates.bak
-rw-r--r-- 1 root root 5532 may 9 22:36 ca-certificates.conf.bak
-rw-r--r-- 1 root root 5532 may 9 22:36 ca-certificates.conf.bak.bak
-rw-r--r-- 1 root root 5529 may 9 22:36 ca-certificates.conf.dpkg-old.bak
-rw-r--r-- 1 root root 5529 may 9 22:36 ca-certificates.conf.dpkg-old.bak.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 chatscripts.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 console-setup.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 cracklib.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 cron.daily.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 cron.d.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 cron.hourly.bak
drwxr-xr-x 2 root root 4096 may 9 22:36 cron.monthly.bak
-rw-r--r-- 1 root root 1136 may 9 22:36 crontab.bak
-rw-r--r-- 1 root root 1136 may 9 22:36 crontab.bak.bak
```

b)

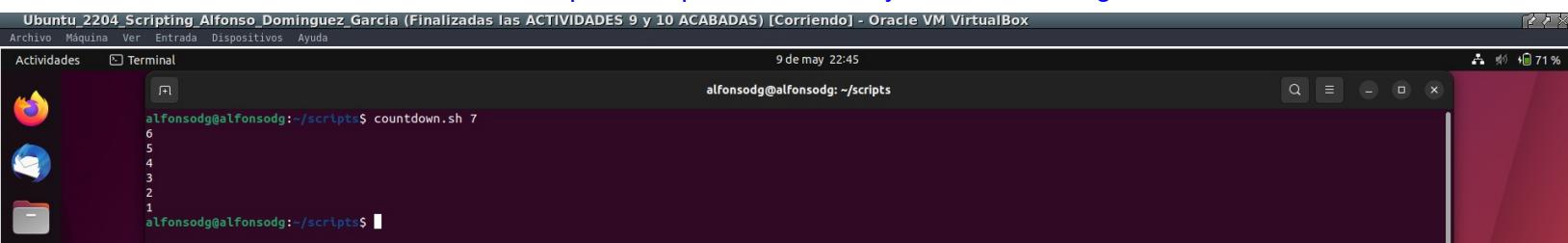
-Ahora crearé el script “**countdown.sh**” que nos permitirá pasar como parámetro \$1 y se almacenará en la variable “parametroUser” de esta forma comprobará que si el usuario no ha pasado ningún argumentos no seguiremos con el script y salga.



```
GNU nano 6.2
#!/bin/bash
parametroUser=$1
if [[ $# -eq 0 ]]; then
    echo "No has pasado ningún parámetro, deberías pasar un número"
    echo
    echo "Saliendo del programa..."
    exit 1
fi
cuentaAtras=$parametroUser
while [ $cuentaAtras -gt 1 ]; do
    cuentaAtras=$(( $cuentaAtras - 1 ))
    sleep 1
    echo $cuentaAtras
done
```

-De lo contrario lo que hará será con un while desde un contador que será la anterior variable **\$parametroUser** o \$1 hasta que no sea mayor a 1 el bucle no parará de ejecutar la cuenta atrás restando el valor de la variable anterior hasta que sea menor que 1.

A continuación mostramos una captura de pantalla con la ejecución del código:



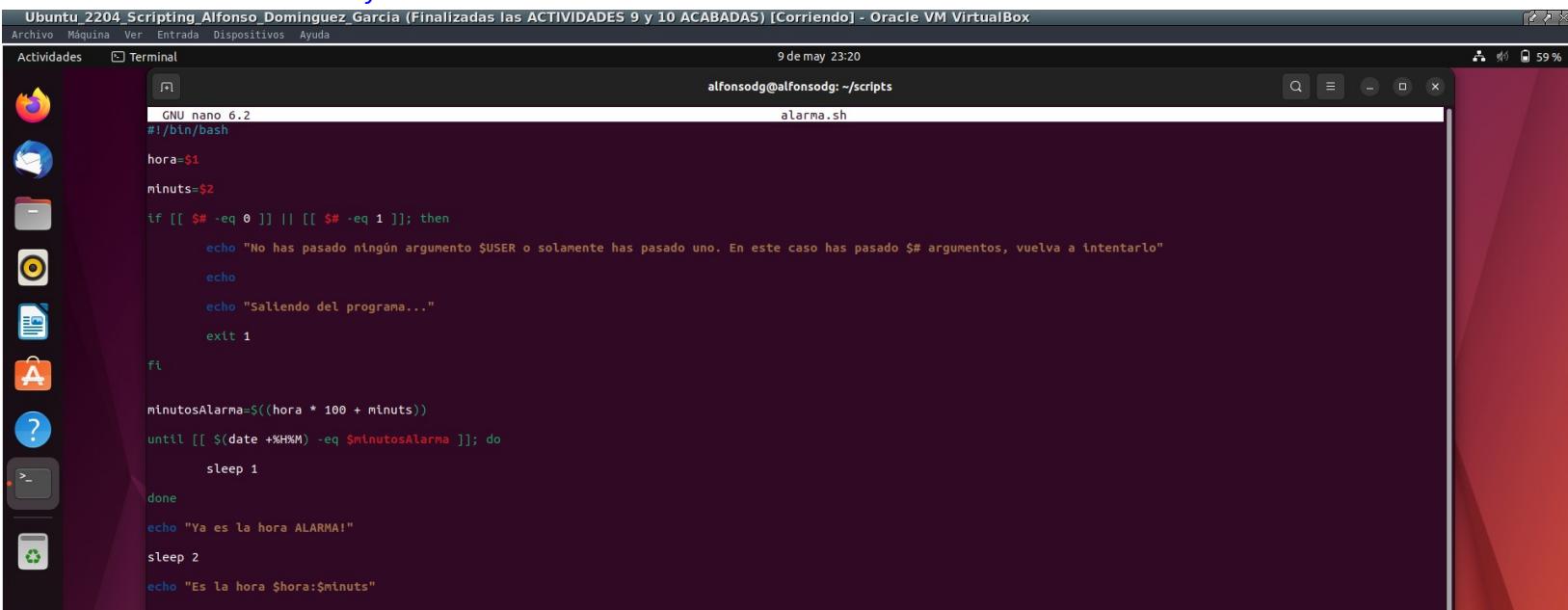
```
alfonsodg@alfonsodg:~/scripts$ countdown.sh 7
6
5
4
3
2
1
alfonsodg@alfonsodg:~/scripts$
```

c)

-En este script **alarma.sh** lo que haremos será pasar como argumentos la hora y los minutos **\$1 y \$2** y los almacenaremos en las variables **\$hora \$minuts** y de esta forma lo que hacemos es un condicional que si ambos argumentos son iguales a 0 o a 1 saltará el mensaje conforme NO se han pasado dos argumentos que son necesarios para la ejecución de este script.

-Lo siguiente que hacemos es el cálculo que nos permita multiplicar la hora * 100 y aumentamos a los minutos con esto por ejemplo tendríamos **23 * 100 + 19 = 2319** que separado será 23 19 que son la hora y los minutos exactos. De esta forma usando la comanda **\$(date +%H%M)** devolverá el resultado de 23 * 100 + 19 que será 2319 y en caso de que no sea igual al resultado obtenido usando Until no parará el bucle de estar en ejecución.

-Y una vez haya acabado el bucle lo que haremos será mostrar el resultado final mediante dos salidas con echo y mostraremos el texto ALRAMA!



The screenshot shows a terminal window titled "alarma.sh" running on an Ubuntu 22.04 LTS system. The terminal content is as follows:

```
GNU nano 6.2
#!/bin/bash

hora=$1
minuts=$2

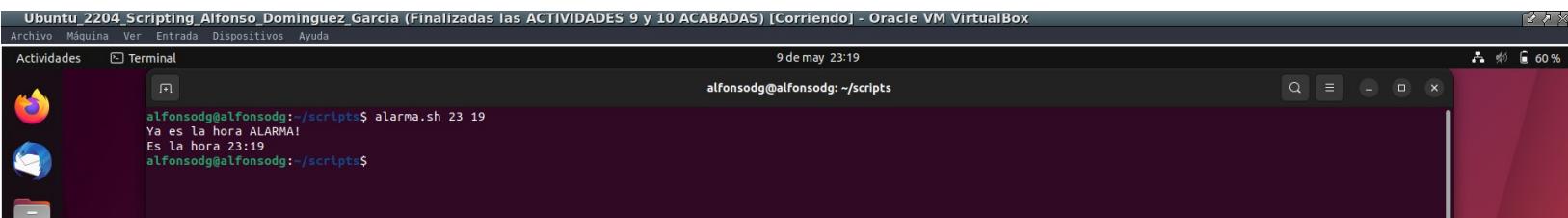
if [[ $# -eq 0 ]] || [[ $# -eq 1 ]]; then
    echo "No has pasado ningún argumento $USER o solamente has pasado uno. En este caso has pasado $# argumentos, vuelva a intentarlo"
    echo
    echo "Saliendo del programa..."
    exit 1
fi

minutosAlarma=$((hora * 100 + minuts))

until [[ $(date +%H%M) -eq $minutosAlarma ]]; do
    sleep 1
done

echo "Ya es la hora ALRAMA!"
sleep 2
echo "Es la hora $hora:$minuts"
```

-Aquí mostramos la ejecución del script **alarma.sh** adjuntando los argumentos y vemos el resultado que muestra:



The screenshot shows a terminal window titled "Terminal" in an Ubuntu 22.04 desktop environment. The window title bar also includes the text "Scripting Alfonso_Dominguez_Garcia (Finalizadas las ACTIVIDADES 9 y 10 ACABADAS) [Corriendo] - Oracle VM VirtualBox". The terminal shows the following command and output:

```
alfonsodg@alfonsodg:~/scripts$ alarma.sh 23 19
Ya es la hora ALARMA!
Es la hora 23:19
alfonsodg@alfonsodg:~/scripts$
```

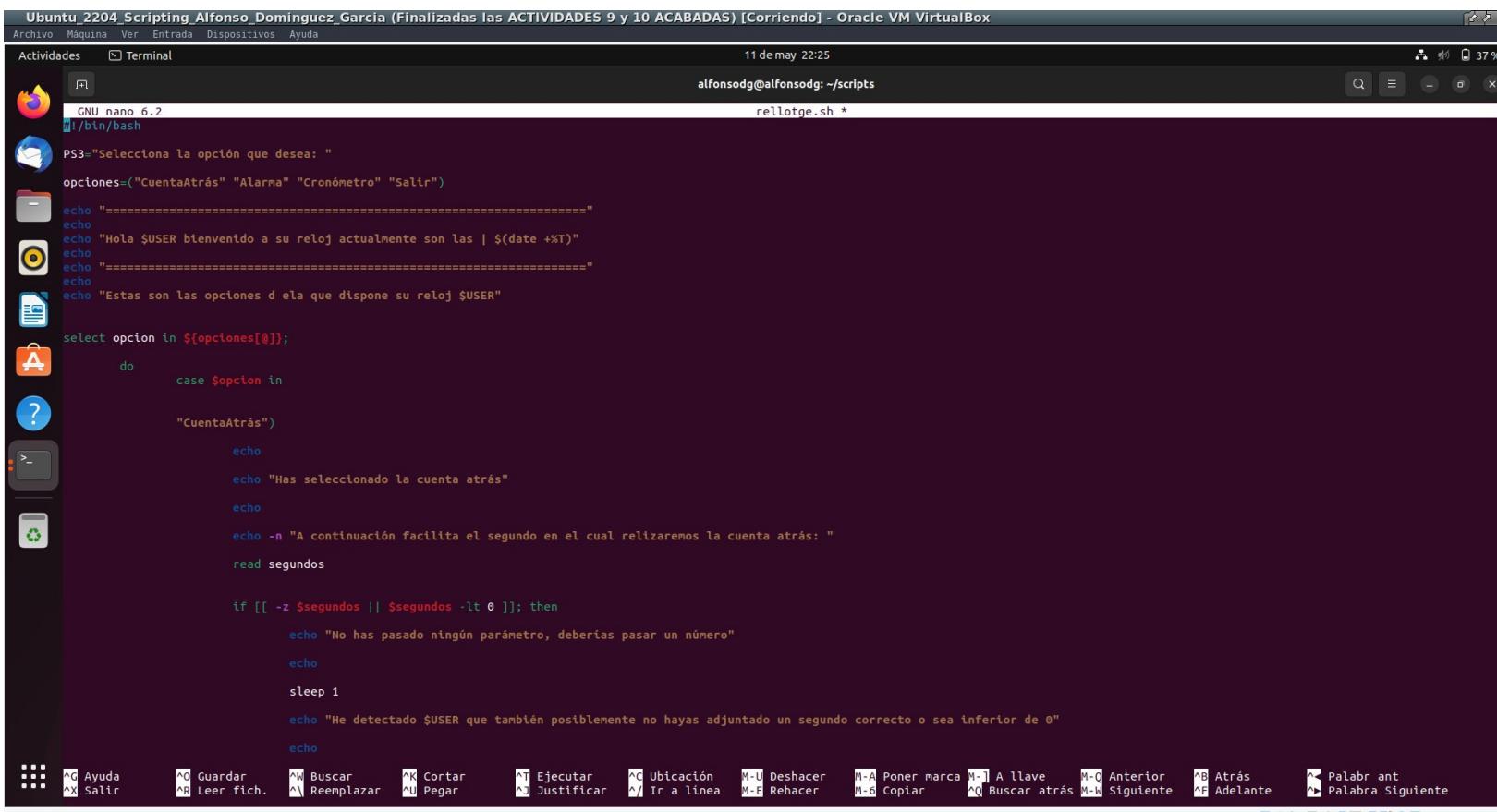
d)

-En este apartado D) mostraré capturas de pantalla de todo el código Bash script rellotge.sh:

-Las cosas principales a destacar serían que habrá un apartado usando “**select**” y un menú principal mostrando un texto que indicará automáticamente la hora actual en formato HH:MM:SS y a continuación la variable PS3 y una tupla de opciones que serían “**CuentaAtrás**” “**Alarma**” “**Cronómetro**” “**Salir**” y cada una de estas opciones estarán en la función **CASE** que estará a continuación:

-Recorreremos la tupla de opciones anterior haciendo un select que diga select opcion in \${opciones[@]} [Recordamos que la variable @ servirá para recorrer todos los elementos de la tupla].

-Ahora muestro captura de pantalla con la opción “CuentaAtrás”. Esta opción pedirá a la usuario que pase los segundos desde donde quiera que parte y vaya descostando hasta el final de la cuenta atrás.



```

Ubuntu_2204 Scripting_Alfonso_Dominguez_Garcia (Finalizadas las ACTIVIDADES 9 y 10 ACABADAS) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 11 de may 22:25
alfonsodg@alfonsodg: ~/scripts
rellotge.sh *

GNU nano 6.2
#!/bin/bash

PS3="Selecciona la opción que deseas: "
opciones=("CuentaAtrás" "Alarma" "Cronómetro" "Salir")

echo =====
echo "Hola $USER bienvenido a su reloj actualmente son las | $(date +%T)"
echo =====
echo "Estas son las opciones de la que dispone su reloj $USER"

select opcion in ${opciones[@]};
do
    case $opcion in
        "CuentaAtrás")
            echo
            echo "Has seleccionado la cuenta atrás"
            echo
            echo -n "A continuación facilita el segundo en el cual realizaremos la cuenta atrás: "
            read segundos

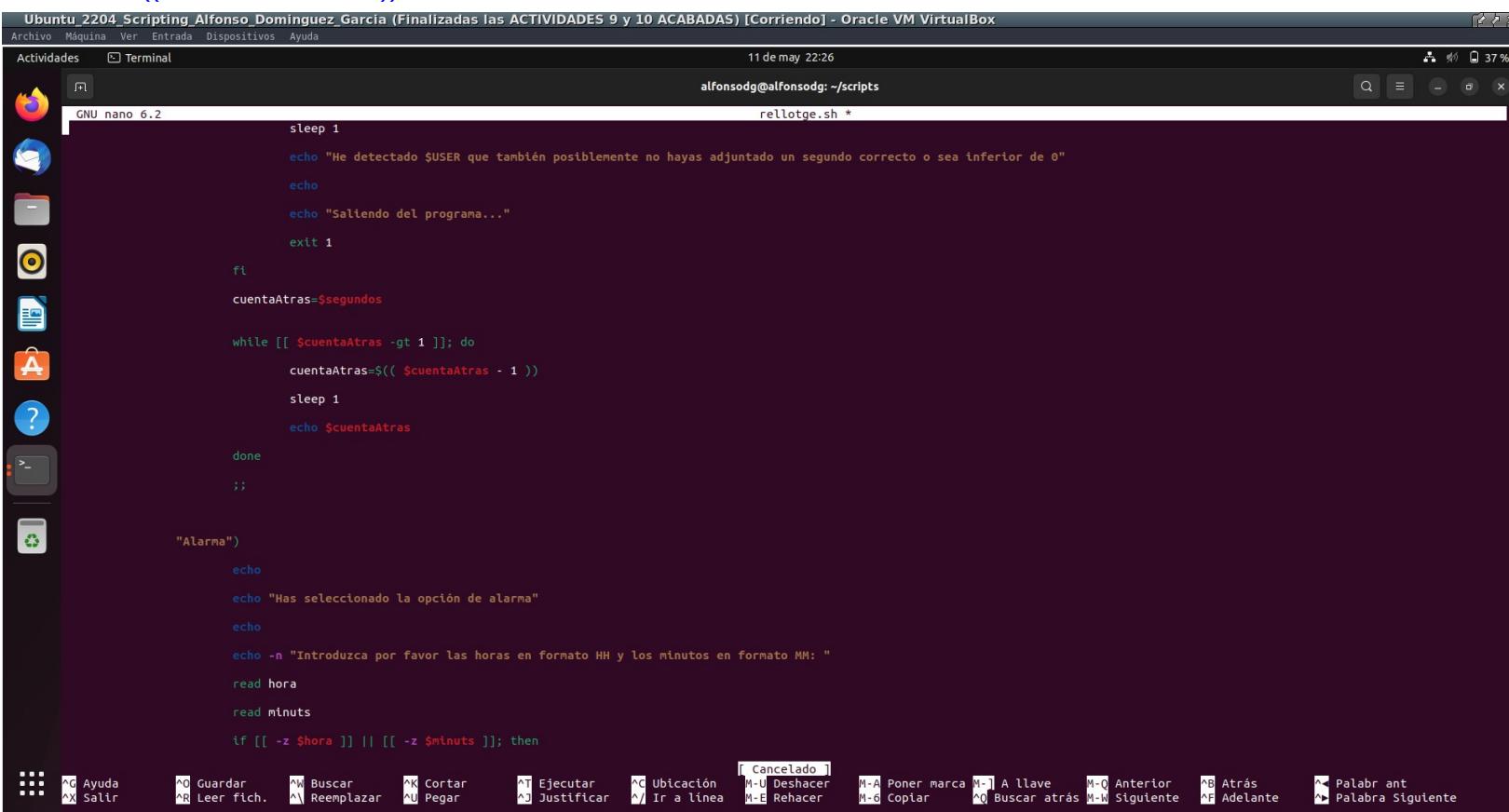
            if [[ -z $segundos || $segundos -lt 0 ]]; then
                echo "No has pasado ningún parámetro, deberías pasar un número"
                echo
                sleep 1
                echo "He detectado $USER que también posiblemente no hayas adjuntado un segundo correcto o sea inferior de 0"
                echo
            fi
        ;;
    esac
done

```

Terminal menu:

- Ayuda Salir
- Guardar Leer fich.
- Buscar Reemplazar
- Cortar Pegar
- Ejecutar Justificar
- Ubicación Ir a línea
- Deshacer Rehacer
- Poner marca A llave
- Buscar atrás Buscar adelante
- Anterior Siguiente
- Atrás Adelante
- Palabra anterior Palabra siguiente

-Para ello he usado un While que mientras que la variable de segundos \$segundos sea mayor que 1 no parará de recorrer el bucle descontando ya que tendrá un cálculo que haga cuentaAtras=\$((\$cuentaAtrás - 1)) e iremos mostrando esta variable dentro de la función WHILE.



The screenshot shows a terminal window titled 'rellotge.sh *' running on an Ubuntu 22.04 LTS system. The script content is as follows:

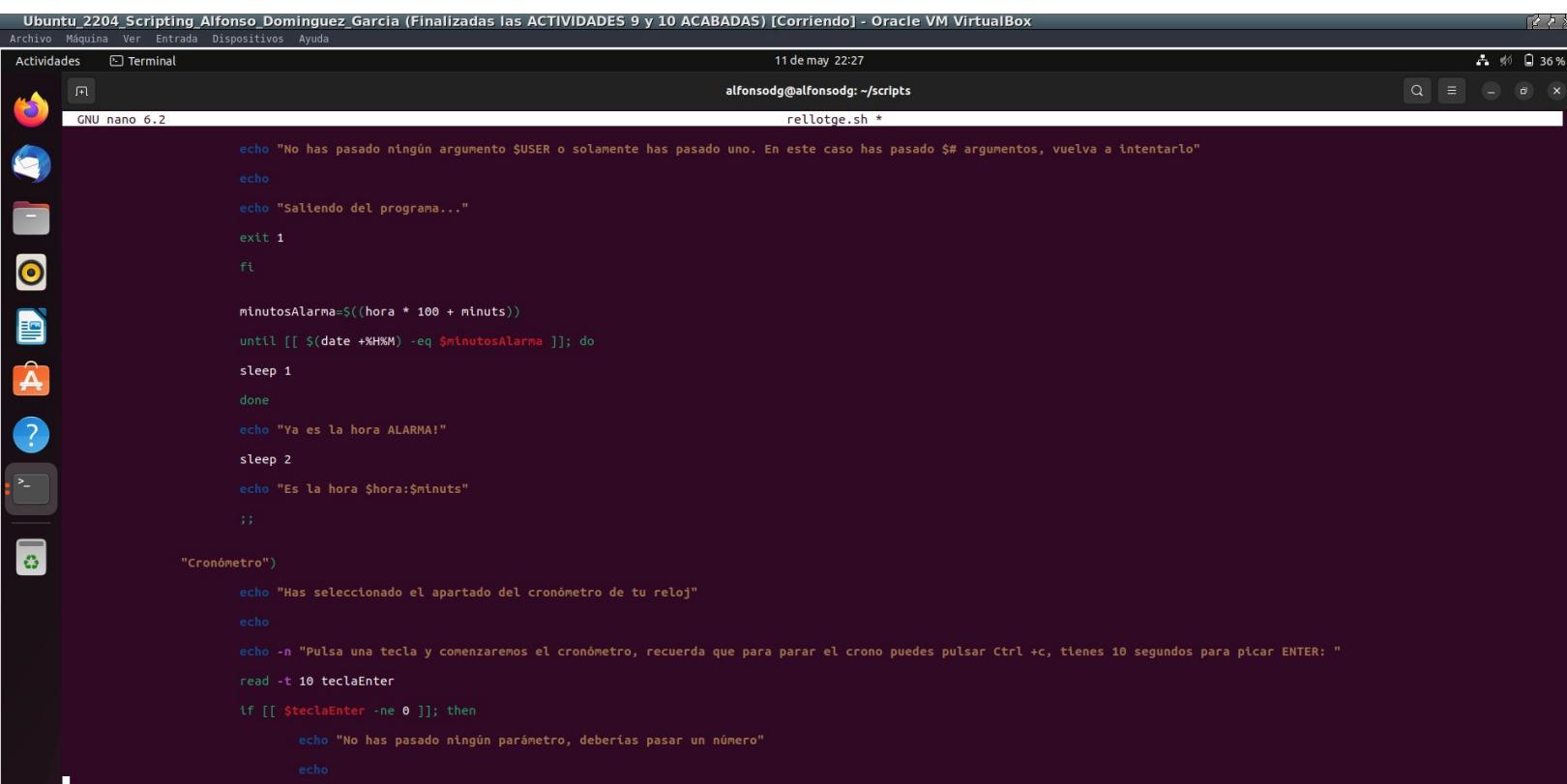
```
GNU nano 6.2
sleep 1
echo "He detectado $USER que también posiblemente no hayas adjuntado un segundo correcto o sea inferior de 0"
echo
echo "Saliendo del programa..."
exit 1
fi
cuentaAtras=$segundos

while [[ $cuentaAtras -gt 1 ]]; do
    cuentaAtras=$(( $cuentaAtras - 1 ))
    sleep 1
    echo $cuentaAtras
done
;;
"Alarma")
echo
echo "Has seleccionado la opción de alarma"
echo
echo -n "Introduzca por favor las horas en formato HH y los minutos en formato MM: "
read hora
read minutus
if [[ -z $hora ]] || [[ -z $minutus ]]; then
    exit 1
fi
echo "La hora es $hora:$minutus"
```

The terminal window also displays a status bar with various keyboard shortcuts for file operations like 'Guardar' (Ctrl-S), 'Copiar' (Ctrl-C), and 'Buscar' (Ctrl-F).

-En este apartado muestro a continuación el apartado de “**Cronómetro**” que leerá el principio de pedirle al usuario que pulse una tecla que será la tecla ENTER (simplemente será un input que estará esperando a la respuesta del usuario) y al pulsar enter lo que hará será

-También comprobará que si el usuario no pulsa la tecla o en este caso no inserta almeno 1 valor no funcionará y saldrá del programa:



```
Ubuntu_2204 Scripting_Alfonso_Dominguez_Garcia (Finalizadas las ACTIVIDADES 9 y 10 ACABADAS) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 11 de may 22:27
alfonsodg@alfonsodg:~/scripts
rellotge.sh *

GNU nano 6.2
echo "No has pasado ningún argumento $USER o solamente has pasado uno. En este caso has pasado $# argumentos, vuelve a intentarlo"
echo
echo "Saliendo del programa..."
exit 1
fi

minutosAlarma=$((hora * 100 + minutos))
until [[ $(date +%H%M) -eq $minutosAlarma ]]; do
sleep 1
done
echo "Ya es la hora ALARMA!"
sleep 2
echo "Es la hora $hora:$minuts"
;;

"Cronómetro")
echo "Has seleccionado el apartado del cronómetro de tu reloj"
echo
echo -n "Pulsa una tecla y comenzaremos el cronómetro, recuerda que para parar el crono puedes pulsar Ctrl +c, tienes 10 segundos para picar ENTER: "
read -t 10 teclaEnter
if [[ $teclaEnter -ne 0 ]]; then
echo "No has pasado ningún parámetro, deberías pasar un número"
echo
```

-Finalmente a nivel de código muestro captura de pantalla conforme la opción “**SALIR**” permitirá mostrar un mensaje de despedida y que en caso de intentar seleccionar cualquier otra cosa mostrará un mensaje que diga que no ha seleccionado ninguna de las cosas anteriores.

```
Ubuntu 22.04 Scripting Alfonso_Domínguez_García (Finalizadas las ACTIVIDADES 9 y 10 ACABADAS) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 11 de may 22:28
alfonsodg@alfonsodg: ~/scripts
GNU nano 6.2
        echo "Salido del programa..."
        exit 1
    fi
    cuentaAtras=$parametroUser
    while [[ $teclaEnter -eq "" ]]; do
        avanzar=$(( $avanzar + 1 ))
        sleep 1
        echo $avanzar
    done
    ;;
    "Salir")
        echo "Salido del programa, gracias por usar el script RELOJ.SH *$USER*"
        echo
        exit 1
    ;;
    *)
        echo "No has seleccionado una opción válida de las anteriores."
        exit 2
    ;;
done
esac
```

^C Ayuda ^O Guardar ^R Leer fich. ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación ^U Deshacer ^A Poner marca ^L A llave ^O Anterior ^B Atrás
^X Salir ^R Reemplazar ^U Pegar ^J Justificar ^I Ir a linea ^E Rehacer ^G Copiar ^Q Buscar atrás ^W Siguiente ^F Adelante ^P Palabra anterior ^N Palabra siguiente

-Muestro a continuación la ejecución del código **rellotge.sh** y conforme los diferentes menús funcionan correctamente y se mostrarán el conjunto de opciones y conforme el usuario va seleccionando las opciones 1) y 3) d ella parte 2) mostraré una segunda captura de pantalla:

```
Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (Finalizadas las ACTIVIDADES 9 y 10 ACABADAS) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 11 de may 22:44
alfonsodg@alfonsodg:~/scripts$ ./rellotge.sh
=====
Hola alfonsodg bienvenido a su reloj actualmente son las | 22:43:06
=====
Estas son las opciones d ela que dispone su reloj alfonsodg
1) CuentaAtrás
2) Alarma
3) Cronómetro
4) Salir
Selecciona la opción que deseas: 1
Has seleccionado la cuenta atrás
A continuación facilita el segundo en el cual realizaremos la cuenta atrás: 5
4
3
2
1
Selección la opción que deseas: 3
Has seleccionado el apartado del cronómetro de tu reloj
Pulsa una tecla y comenzaremos el cronómetro, recuerda que para parar el crono puedes pulsar Ctrl +c, tienes 10 segundos para picar ENTER:
1
2
3
4
5
6
^C
alfonsodg@alfonsodg:~/scripts$ ./rellotge.sh
=====
Hola alfonsodg bienvenido a su reloj actualmente son las | 22:43:49
=====
Estas son las opciones d ela que dispone su reloj alfonsodg
1) CuentaAtrás
2) Alarma
3) Cronómetro
4) Salir
Selecciona la opción que deseas: 2
Has seleccionado la opción de alarma
Introduzca por favor las horas en formato HH y los minutos en formato MM: 22 44
```

-Esta captura de pantalla muestra conforme la opción número 2 de la alarma funciona correctamente y muestra la salida que se desea justamente a la hora introducida por el usuario:

```
Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (Todas las actividades finalizadas) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 14 de may 11:31
alfonsodg@alfonsodg: ~
alfonsodg@alfonsodg: $ rellotge.sh
=====
Hola alfonsodg bienvenido a su reloj actualmente son las | 11:30:16
=====
Estas son las opciones d ela que dispone su reloj alfonsodg
1) CuentaAtrás
2) Alarma
3) Cronómetro
4) Salir
Selecciona la opción que deseas: 2
Has seleccionado la opción de alarma
Introduzca por favor las horas en formato HH: 11
Introduzca los minutos en formato MM: 31
Ya es la hora ALARMA!
Es la hora 11:31
Selecciona la opción que deseas: 4
Saliendo del programa, gracias por usar el script RELOJ.SH *alfonsodg*
alfonsodg@alfonsodg: ~$
```

ACT. 12

-En este apartado de la ACT 12 lo que haremos será crear una lista llamada “*llistanum*” que contendrá todos los parámetros que pase el usuario que será \$@ y haremos que si la lista se encuentra vacía entonces salga del programa ya que el usuario no habrá pasado argumentos a la lista.

-Lo siguiente que haremos será crear una variable contador que sea igual a 0 | Y que cuente cada elemento de la lista y en caso de ser diferente a “x” continue con el bucle WHILE y no salga.

El siguiente paso será comprobar si es par o impar el número por lo que establecemos que si la variable *num=\$((\${llistanum[count]} %2))* serán pares en caso de que sean iguales a 0:

Pintaremos el valor del argumento que había en la lista y diremos que es par. Para printarlo lo que haremos será hacer lo siguiente → ***echo "\${llistanum[count]} es par o impar"***:

-Finalmente para conseguir parar el bucle WHILE sumaremos a la variable ***count o contador + 1*** al final de cada condicional:

```

Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (Finalizadas las ACTIVIDADES 9 y 10 ACABADAS) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 11 de may 22:57
alfonsodg@alfonsodg: ~/scripts
senarsparells.sh
GNU nano 6.2
#!/bin/bash

# Este script permite saber si un valor es par o impar usando una lista

llistanum=$@

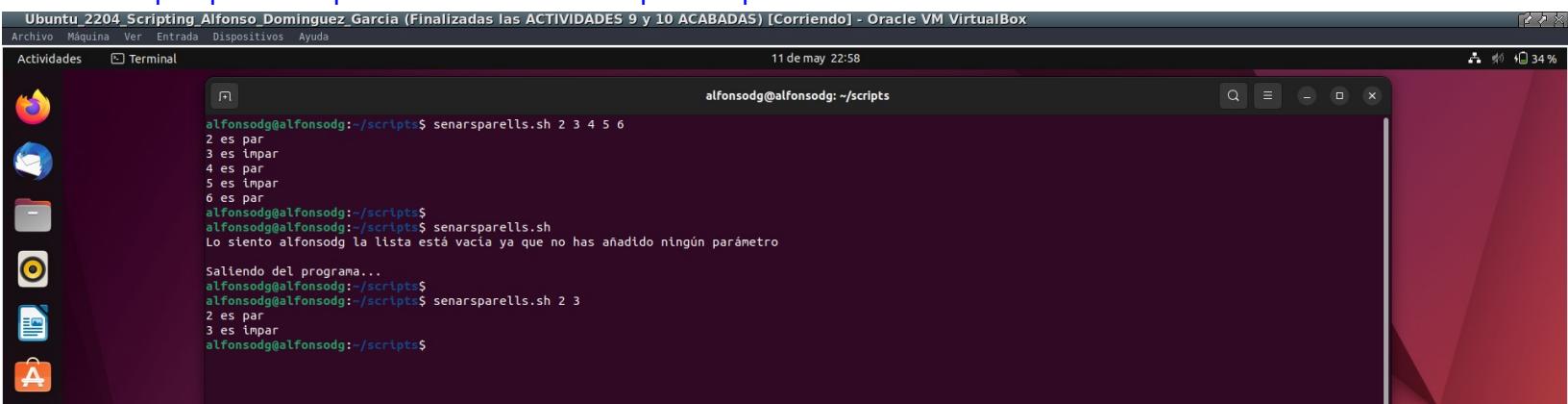
if [[ $llistanum -eq 0 ]]; then
    echo "Lo siento $USER la lista está vacía ya que no has añadido ningún parámetro"
    echo
    echo "Saliendo del programa..."
    exit 1
fi

count=0

while [ "x${llistanum[count]}" != "x" ]; do
    num=$(( ${llistanum[count]} % 2 ))
    if [[ $num -eq 0 ]]; then
        echo "${llistanum[count]} es par"
    else
        echo "${llistanum[count]} es impar"
    fi
    count=$(( $count + 1 ))
done

```

-Aquí muestro que al ejecutar el script “**senarparells.sh**” y pasamos como argumentos los números que queramos pues nos mostrará si es par o impar en la salida:

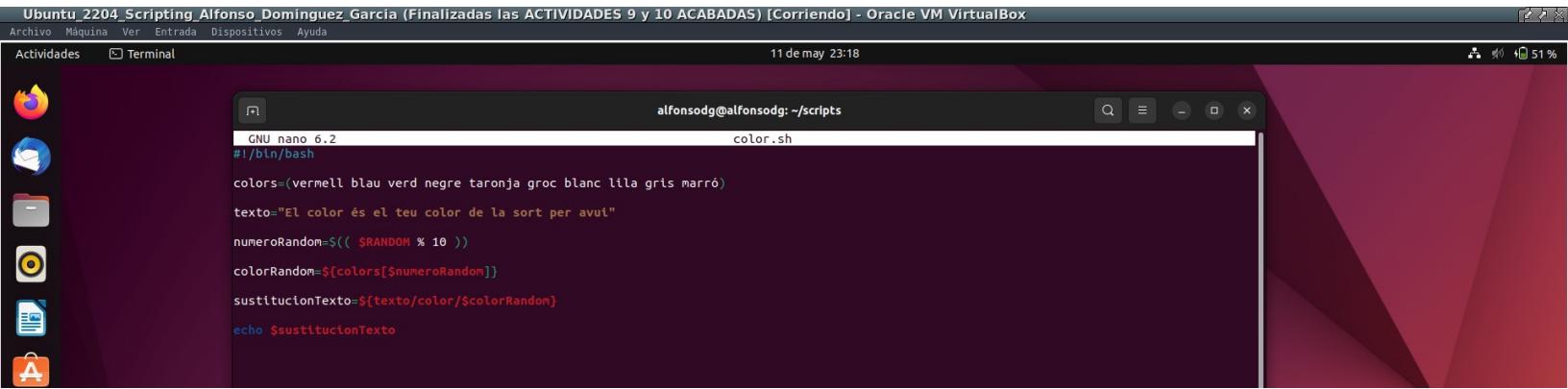


The screenshot shows a terminal window titled "Terminal" running on an Ubuntu 22.04 LTS system. The window title bar includes the text "Ubuntu 22.04 Scripting Alfonso_Dominguez_Garcia (Finalizadas las ACTIVIDADES 9 y 10 ACABADAS) [Corriendo] - Oracle VM VirtualBox". The terminal content is as follows:

```
alfonsodg@alfonsodg:~/scripts$ senarparells.sh 2 3 4 5 6
2 es par
3 es impar
4 es par
5 es impar
6 es par
alfonsodg@alfonsodg:~/scripts$ alfonsodg@alfonsodg:~/scripts$ senarparells.sh
Lo siento alfonsodg la lista está vacía ya que no has añadido ningún parámetro
Saliendo del programa...
alfonsodg@alfonsodg:~/scripts$ alfonsodg@alfonsodg:~/scripts$ senarparells.sh 2 3
2 es par
3 es impar
alfonsodg@alfonsodg:~/scripts$
```

ACT. 13

-El objetivo de la ACT 13 era crear una variable que permita substituir el texto dado de una tupla de valores llamada colors y que contendrá un listado de colores “**vermell blau verd negre taronja groc blanc lila gris i marró**”. De forma aleatoria tendrá que cambiar el texto

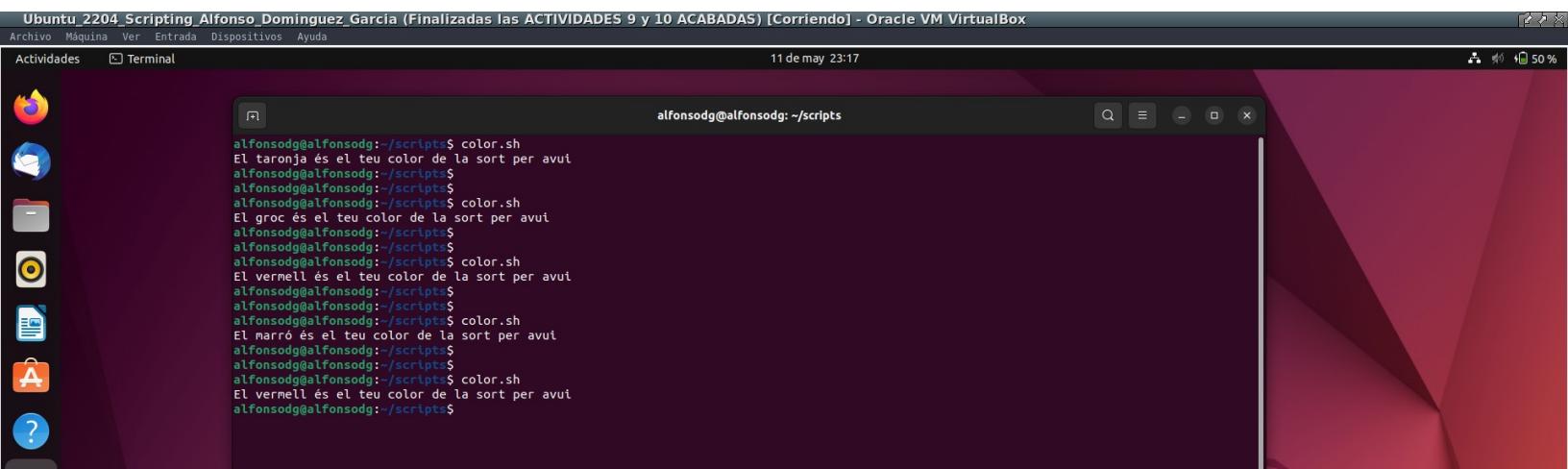


```
GNU nano 6.2
#!/bin/bash

colors=(vermell blau verd negre taronja groc blanc lila gris marró)
texto="El color és el teu color de la sort per avui"
numeroRandom=$(( $RANDOM % 10 ))
colorRandom=${colors[$numeroRandom]}
sustitucionTexto=${texto/color/$colorRandom}
echo $sustitucionTexto
```

-Lo que he hecho ha sido primero, almacenar en una variable llamada `numeroRandom=$(($RANDOM % 10))` y otra variable que se llamará `sustitucionTexto=${texto/color/colorRandom}` y a su vez para poder realizar la sustitución del texto será `colorRandom=${colors[$numeroRandom]}` y printaremos el texto final de `$sustitucionTexto`

-Ahora muestro captura de pantalla conforme al ejecutar el script “color.sh” veremos que a cada pasada de la tupla anterior aleatoriamente seleccionará el color que sea de forma aleatoria ya que de el texto siguiente → “**El color és el teu color de la sort per avui**” de forma **RANDOM del 1 AL 10** escogerá un color que se encuentre dentro del rango cada vez que ejecute este script.



```
alfonsodg@alfonsodg:~/scripts$ color.sh
El taronja és el teu color de la sort per avui
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ color.sh
El groc és el teu color de la sort per avui
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ color.sh
El vermell és el teu color de la sort per avui
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ color.sh
El marró és el teu color de la sort per avui
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ 
alfonsodg@alfonsodg:~/scripts$ color.sh
El vermell és el teu color de la sort per avui
alfonsodg@alfonsodg:~/scripts$ 
```

ACT. 14

-En este apartado de la Actividad 14 lo que haré será crear primero en `/etc/` el fichero funciona con permisos de “`sudo`” y dentro del fichero `/etc/funcions` lo que haremos será crear una función llamada “`noparam()`” y que printará solamente un echo que permita mostrar un echo que avise al usuario de que no ha establecido ningún valor:

-En la captura siguiente (a mano derecha) lo que haré será crear el script “`sumaarg.sh`” llamaremos a la función usando el comando “`source /etc/funcions`” que contará que si los parámetros sean igual a 0 pues llamará a la función “`noparam`” y saldrá del programa en caso de que no pase ningún argumento:

-De lo contrario sumará los argumentos de `$1 + $2` de la siguiente forma → `sumaArgumentos=$(($1+$2))` y mostraré el resultado de la suma de `N` parámetros:

```

Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (Falta repasar ACT 11 el script de reloj.sh la ACT 12 y 13 están acabadas) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 13 de may 17:29
alfonsodg@alfonsodg: ~
GNU nano 6.2
/etc/funcions
noparam() {
    echo "Lo siento $USER, no has facilitado ningún parámetro. Necesita facilitar al menos un argumento"
}

alfonsodg@alfonsodg: ~/scripts
GNU nano 6.2
sumaarg.sh
#!/bin/bash

echo "Hola $USER bienvenido al script/función que te permitirá sumar dos elementos pasados como argumentos"

echo

source /etc/funcions

if [ $# -eq 0 ]; then
    noparam
    exit 1
fi

sumaArgumentos=$(( $1+$2 ))
echo "El resultado de la suma de $# argumentos pasados y los parámetros $1 y $2 será igual a $sumaArgumentos"

```

-En esta captura de pantalla muestro el resultado de la ejecución y conforme podemos pasar los argumentos necesarios:

```

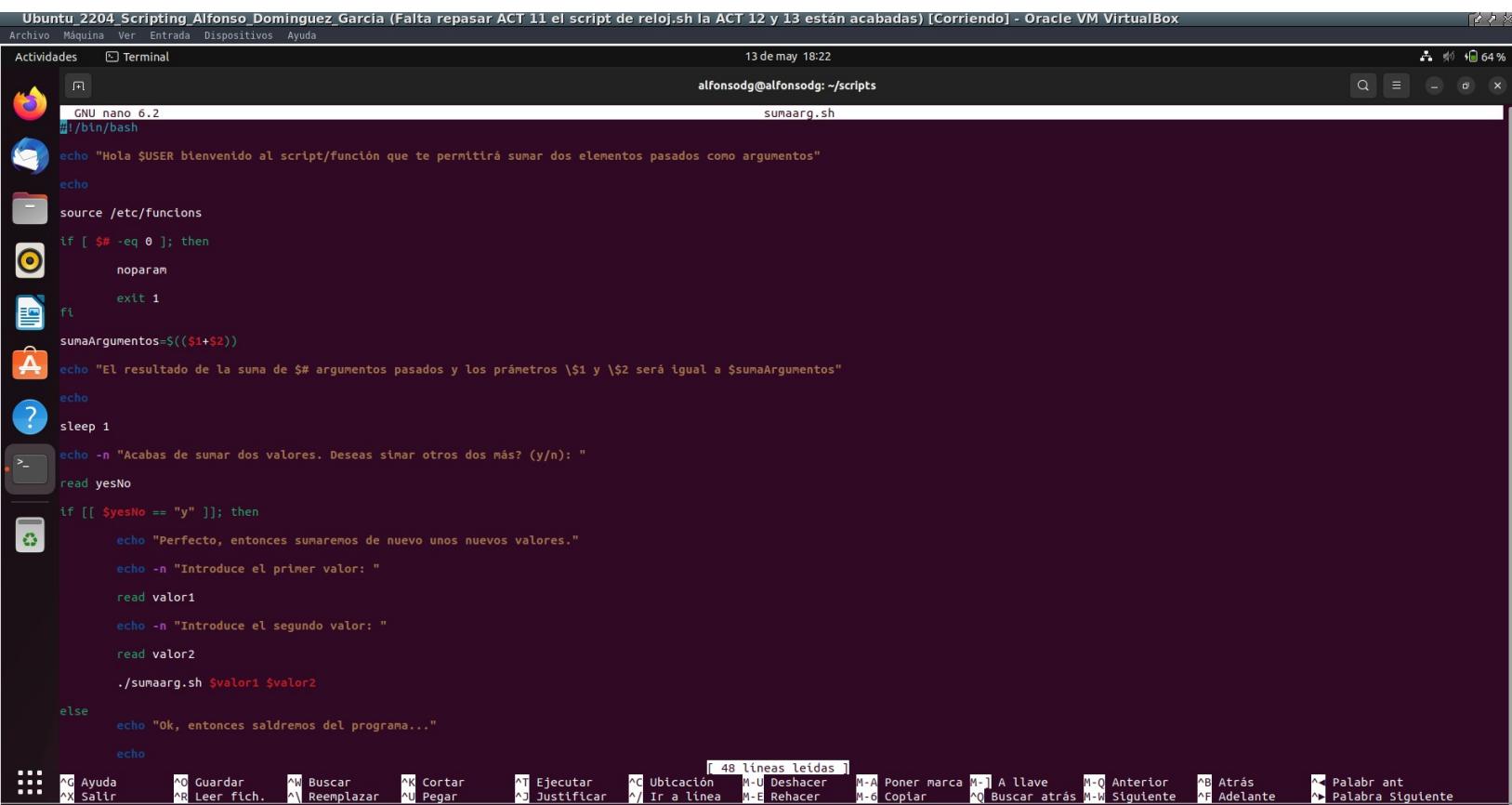
Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (Todas las actividades finalizadas) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 14 de may 12:40
alfonsodg@alfonsodg: ~
alfonsodg@alfonsodg: $ sumaarg.sh 3 5
Hola alfonsodg bienvenido al script/función que te permitirá sumar dos elementos pasados como argumentos
El resultado de la suma de 2 argumentos pasados y los parámetros $1 y $2 será igual a 8
Acabas de sumar dos valores. Deseas sumar otros dos más? (y/n): n
Ok, entonces saldremos del programa...
alfonsodg@alfonsodg: $

```

-Ahora mejoraremos el script "sumaarg.sh" conforme le pediremos al usuario si desea introducir más argumentos, con lo que al acabar el resultado de la suma de los parámetros lo que haré será pedirle al usuario mediante un **echo -n** si desea introducir más elementos "**yesNo**" que se almacenará en esta variable.

-Luego un condicional que indique que si **yesNo** es igual a "y" entonces le pediremos al usuario dos valores que se almacenarán en dos variables "**valor 1**" y "**valor 2**" y llamaremos al script de nuevo utilizando o bien **source sumaarg.sh** o bien **./sumaarg.sh \$valor1 \$valor2** pasados como valores de parámetro:

-En caso de que el usuario haya picado cualquier otra cosa que sea "**n**" pues saldremos del programa:



```

Ubuntu_2204.Scripting_Alfonso_Dominguez_Garcia [Falta repasar ACT 11 el script de reloj.sh la ACT 12 y 13 están acabadas] [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 13 de may 18:22
alfonsodg@alfonsodg: ~/scripts
sumaarg.sh
GNU nano 6.2
#!/bin/bash
echo "Hola $USER bienvenido al script/función que te permitirá sumar dos elementos pasados como argumentos"
echo
source /etc/funcions
if [ $# -eq 0 ]; then
    nparam
    exit 1
fi
sumaArgumentos=$((1+$2))
echo "El resultado de la suma de $# argumentos pasados y los prámetros \$1 y \$2 será igual a $sumaArgumentos"
echo
sleep 1
echo -n "Acabas de sumar dos valores. Deseas sumar otros dos más? (y/n): "
read yesNo
if [[ $yesNo == "y" ]]; then
    echo "Perfecto, entonces sumaremos de nuevo unos nuevos valores."
    echo -n "Introduce el primer valor: "
    read valor1
    echo -n "Introduce el segundo valor: "
    read valor2
    ./sumaarg.sh $valor1 $valor2
else
    echo "Ok, entonces saldremos del programa..."
    echo
fi
[ 48 líneas leidas ]
^G Ayuda      ^O Guardar      ^W Buscar      ^K Cortar      ^T Ejecutar      ^C Ubicación      M-U Deshacer      M-A Poner marca      M-[ A llave      M-Q Anterior      ^B Atrás      ^P Palabri ant
^X Salir      ^R Leer fich.  ^\ Reemplazar  ^U Pegar       ^J Justificar   ^I Ir a linea     M-E Rehacer      M-C Copiar       M-[ Buscar atrás M-W Siguiente    ^F Adelante    ^N Palabra Siguiente

```

-Esta captura de pantalla muestra el final del script “**sumaarg.sh**”:

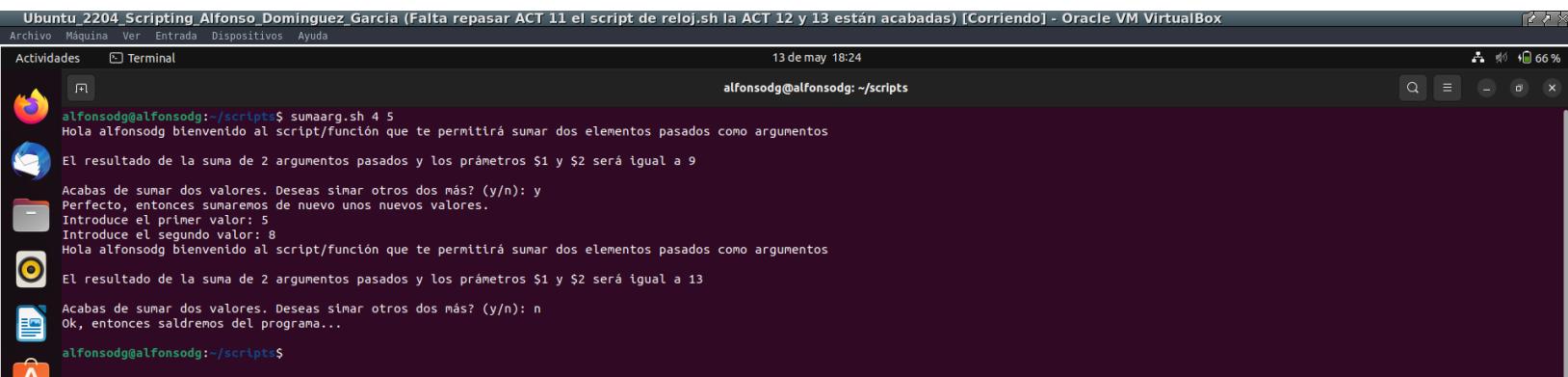
```

else
    echo "Ok, entonces saldremos del programa..."
    echo
    exit 2
fi

^G Ayuda      ^O Guardar     ^W Buscar      ^K Cortar      ^T Ejecutar
^X Salir      ^R Leer fich.  ^\ Reemplazar   ^U Pegar       ^J Justificar
^C

```

-Finalmente muestro captura de pantalla del resultado final ejecutando este script modificado y conforme pasamos dos argumentos primero de todo y al acabar de realizar la suma de los elementos le pediremos al usuario si desea volver a ejecutar el programa de nuevo y esta vez que introduzca nuevos parámetros y sumarlos para obtener el total de la suma:



The screenshot shows a terminal window titled "Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (Falta repasar ACT 11 el script de reloj.sh la ACT 12 y 13 están acabadas) [Corriendo] - Oracle VM VirtualBox". The terminal content is as follows:

```

alfonsodg@alfonsodg:~/scripts$ sumaarg.sh 4 5
Hola alfonsodg bienvenido al script/funció n que te permitirá sumar dos elementos pasados como argumentos
El resultado de la suma de 2 argumentos pasados y los prámetros $1 y $2 será igual a 9
Acabas de sumar dos valores. Deseas sumar otros dos más? (y/n): y
Perfecto, entonces sumaremos de nuevo unos nuevos valores.
Introduce el primer valor: 5
Introduce el segundo valor: 8
Hola alfonsodg bienvenido al script/funció n que te permitirá sumar dos elementos pasados como argumentos
El resultado de la suma de 2 argumentos pasados y los prámetros $1 y $2 será igual a 13
Acabas de sumar dos valores. Deseas sumar otros dos más? (y/n): n
Ok, entonces saldremos del programa...
alfonsodg@alfonsodg:~/scripts$
```

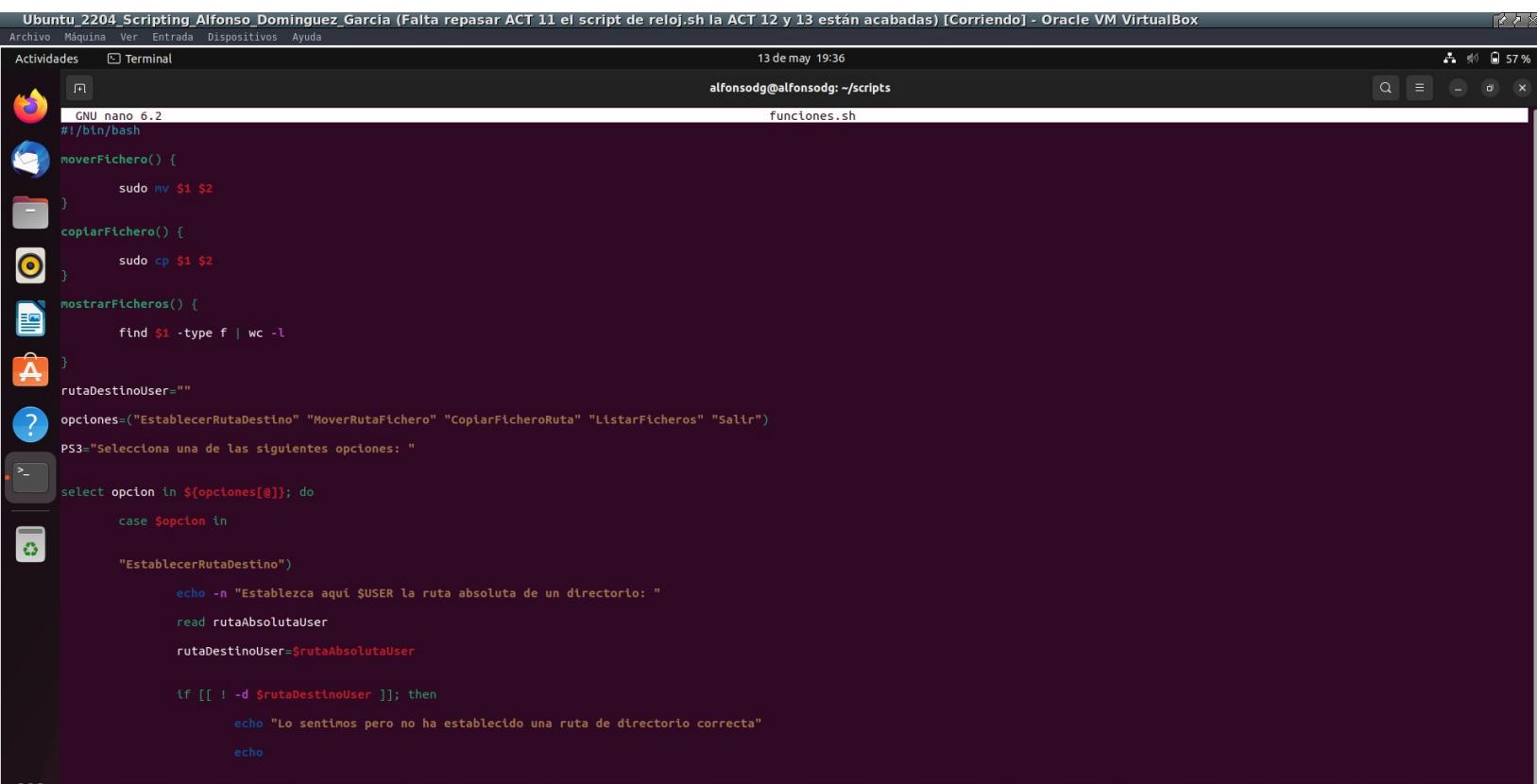
ACT. 14 | EXTRA

-En este script que he denominado “**funciones.sh**” que contendrá tres funciones “**moverFichero()**” | “**copiarFichero()**” | “**mostrarFicheros()**” que la primera función permitirá mover los parámetros \$1 a \$2. Má sedante comentaré que es **\$1 y \$2** que será **\$1** el valor del fichero y **\$2** será la ruta absoluta donde desea almacenar los valores:

-La primera función **moverFichero()** que permitirá mover un fichero a la ruta que deseé el usuario → **sudo mv \$1 \$2**. La segunda función **copiarFichero()** permitirá copiar el fichero mediante el comando → **sudo cp \$1 \$2** y finalmente la función “**mostrarFicheros()**” nos permitirá encontrar y contar los ficheros que hay dentro del directorio que haya decidido pasar el usuario.

-Crearemos una variable llamada “**rutaDestinoUser**” que contendrá almacenada la ruta en formato string:

-Mediante un select recorrerá las opciones que he creado en una tupla llamada opciones, y luego recorreremos todos los valores que hayan en opciones que se realizará d ella siguiente forma → **\$ {opciones[@]}** y luego crearé un case con las funciones de la tupla opciones.



The screenshot shows a terminal window titled "Ubuntu 22.04 Scripting Alfonso_Dominguez_Garcia (Falta repasar ACT 11 el script de reloj.sh la ACT 12 y 13 están acabadas) [Corriendo] - Oracle VM VirtualBox". The terminal displays the following content of the funciones.sh script:

```
GNU nano 6.2
#!/bin/bash

moverFichero() {
    sudo mv $1 $2
}

copiarFichero() {
    sudo cp $1 $2
}

mostrarFicheros() {
    find $1 -type f | wc -l
}

rutaDestinoUser=""

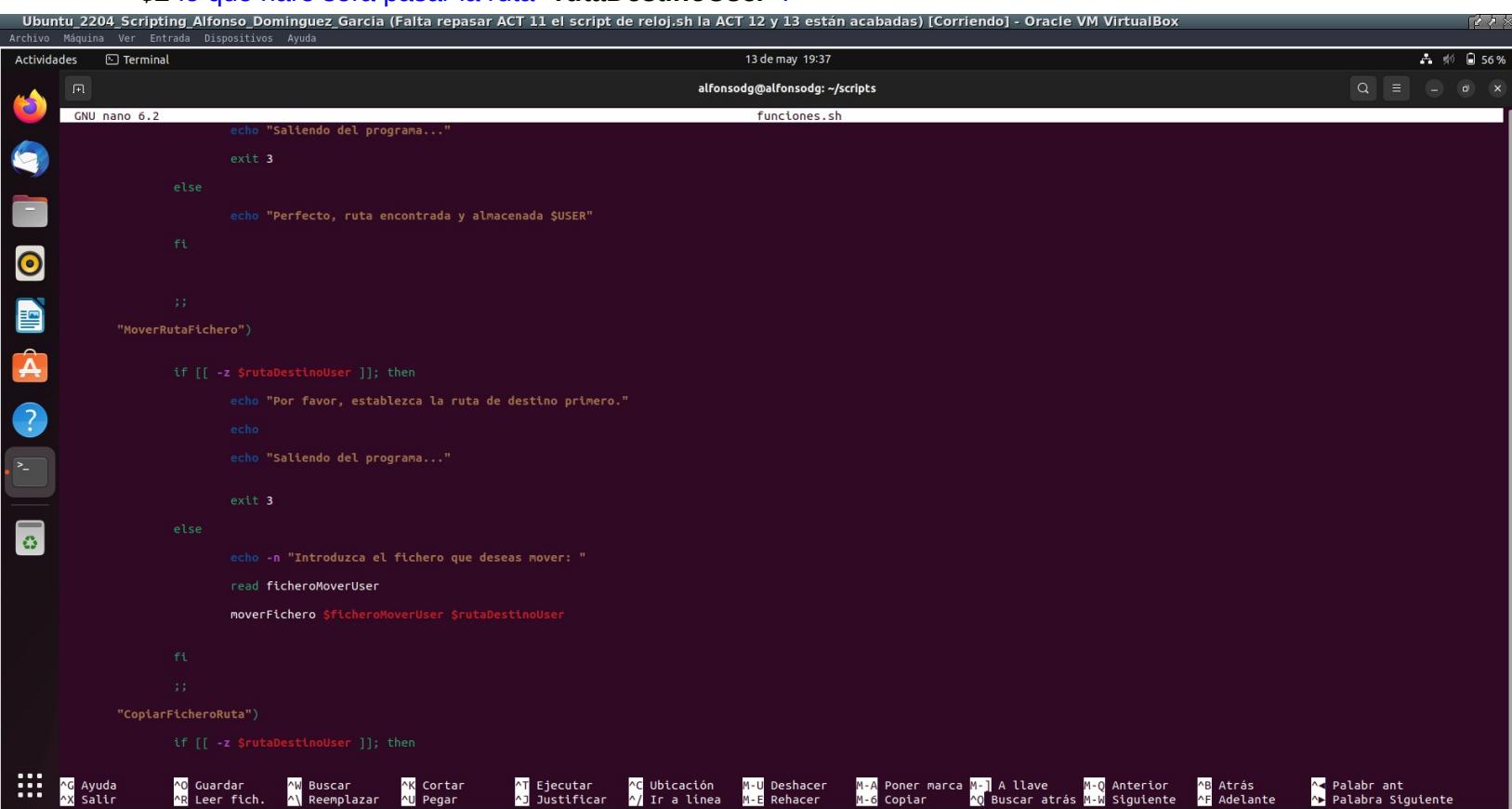
opciones=("EstablecerRutaDestino" "MoverRutaFichero" "CopiarFicheroRuta" "ListarFicheros" "Salir")

PS3="Selecciona una de las siguientes opciones: "

select opcion in ${opciones[@]}; do
    case $opcion in
        "EstablecerRutaDestino")
            echo -n "Establezca aquí $USER la ruta absoluta de un directorio: "
            read rutaAbsolutaUser
            rutaDestinoUser=$rutaAbsolutaUser
            if [[ ! -d $rutaDestinoUser ]]; then
                echo "Lo sentimos pero no ha establecido una ruta de directorio correcta"
                echo
            fi
        ;;
    esac
done
```

-Ahora muestro captura del código conforme se muestra el apartado “**MoverRutaFichero**” que permitirá comprobar que si el usuario no ha introducido la ruta de destino previamente que vuelva a insertar la ruta o que pase por la opción número **1** obligatoriamente:

-En caso de que ya haya establecido la ruta lo que hará será mostrar el resultado y a su vez introducirá el usuario el nombre del fichero que desea mover y lo que hará será llamar a la función con ambos parámetros, el parámetro **\$1** será la variable “**ficheroMoverUsuario**” y en el parámetro **\$2** lo que haré será pasar la ruta “**rutaDestinoUser**”.



The screenshot shows a Linux desktop environment with a terminal window open. The terminal title is "Ubuntu_2204 Scripting_Alfonso_Dominguez_Garcia (Falta repasar ACT 11 el script de reloj.sh la ACT 12 y 13 están acabadas) [Corriendo] - Oracle VM VirtualBox". The terminal window contains a script named "funciones.sh" with the following content:

```
GNU nano 6.2
echo "Saliendo del programa..."

exit 3

else
    echo "Perfecto, ruta encontrada y almacenada $USER"
fi

;;
"MoverRutaFichero")

if [[ -z $rutaDestinoUser ]]; then
    echo "Por favor, estableza la ruta de destino primero."
    echo
    echo "Saliendo del programa..."

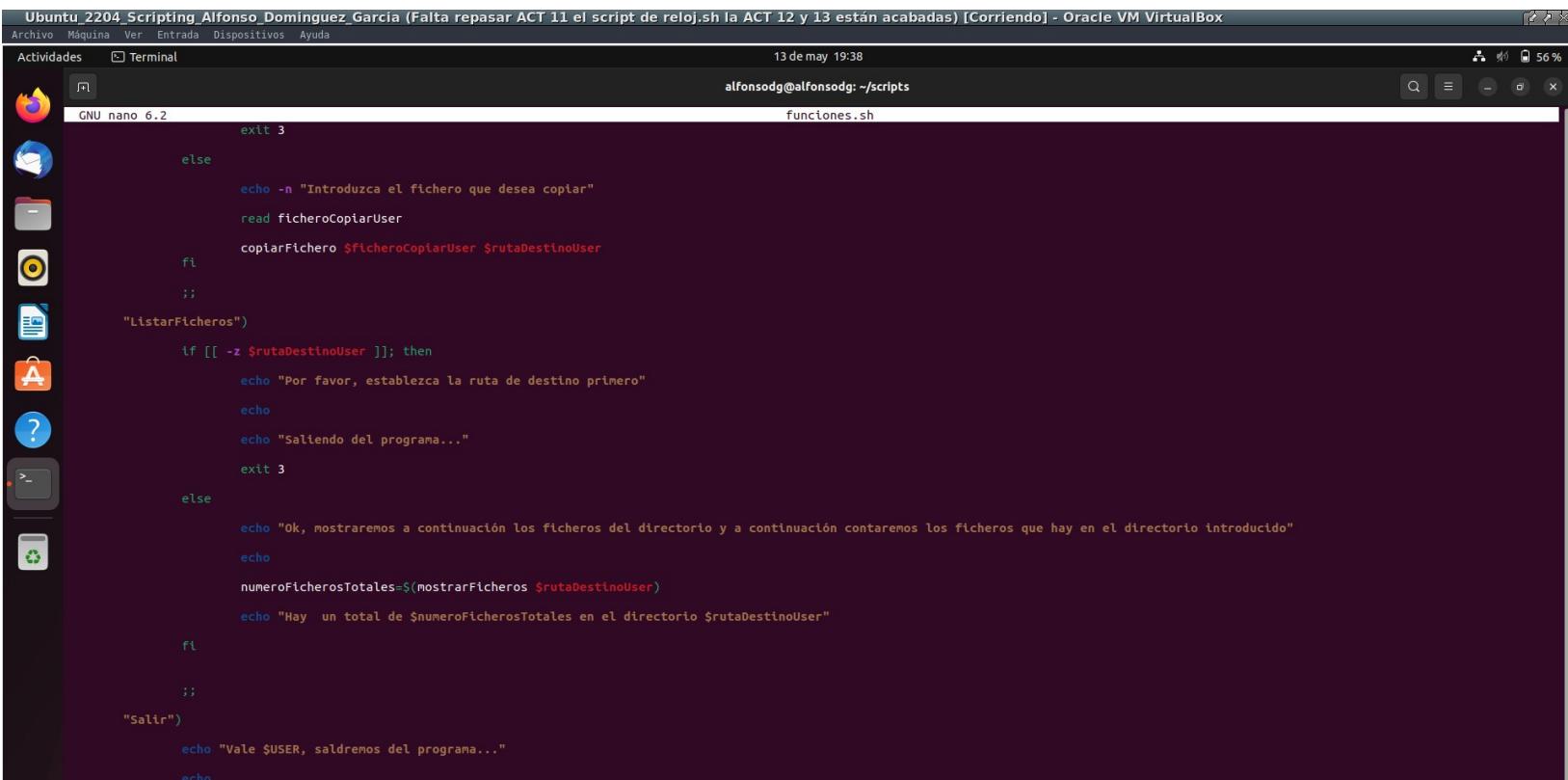
    exit 3
else
    echo -n "Introduzca el fichero que deseas mover: "
    read ficheroMoverUser
    moverFichero $ficheroMoverUser $rutaDestinoUser
fi
;;
"CopiarFicheroRuta")

if [[ -z $rutaDestinoUser ]]; then
```

The terminal window also shows a file manager sidebar on the left and a dock at the bottom with various icons and keyboard shortcuts.

-En la opción “**ListarFicheros**” lo que hará será comprobara que el usuario haya establecido la ruta de forma correcta, de lo contrario pues saldrá del programa y en caso de que ya haya establecido una ruta previamente.

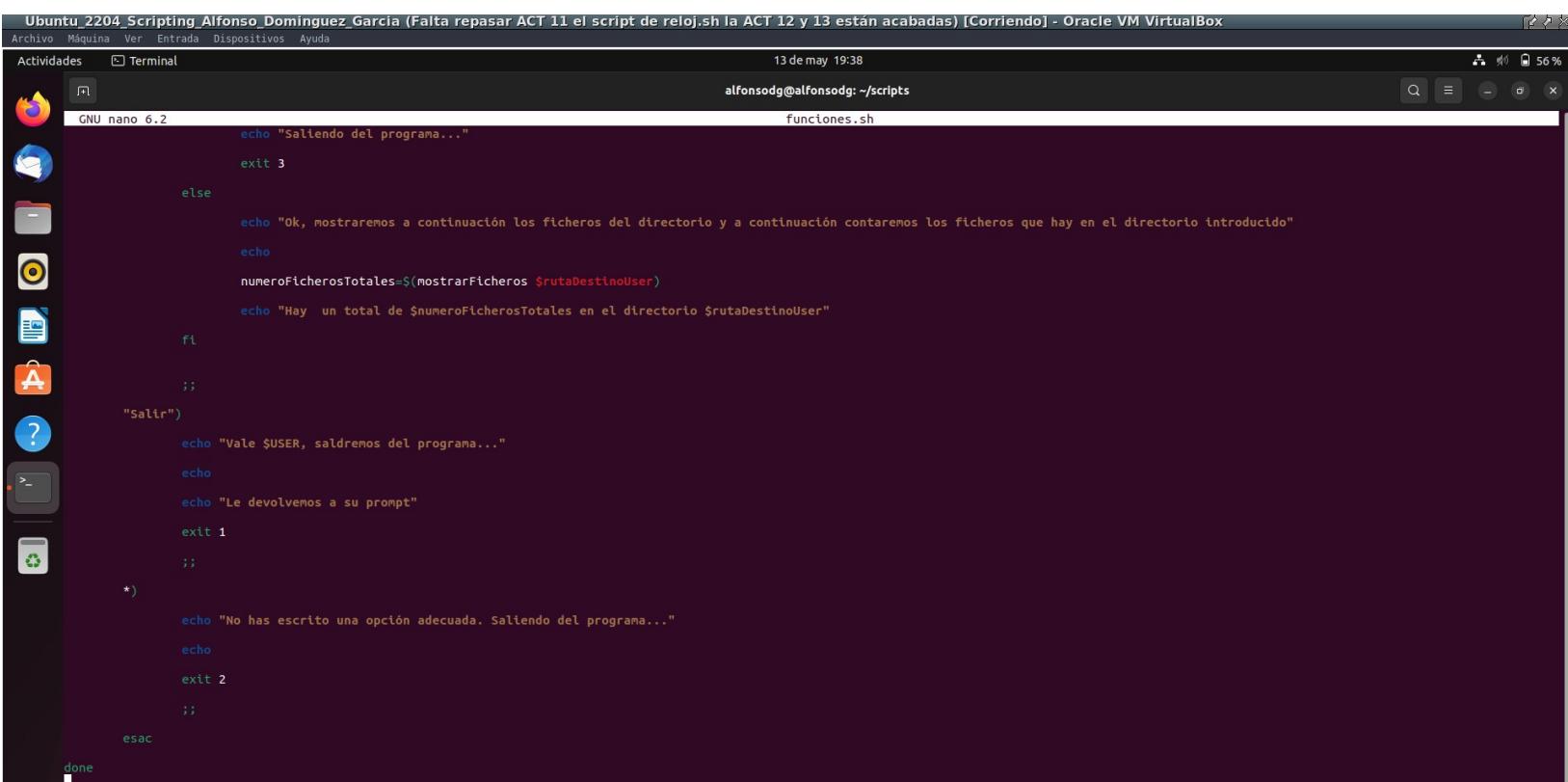
Pues almacenará en una variable llamada “**numeroFicherosTotales**” la función **mostrarFicheros** y \$1 que será la **rutaDestinoUser** de esta forma mostrará los valores ejecutando el siguiente echo → “Hay un total de \$numeroFicherosTotales”:



```
Ubuntu 22.04 Scripting_Alfonso_Dominguez_Garcia (Falta repasar ACT 11 el script de reloj.sh la ACT 12 y 13 están acabadas) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 13 de may 19:38
alfonsodg@alfonsodg: ~/scripts
funciones.sh
GNU nano 6.2
exit 3

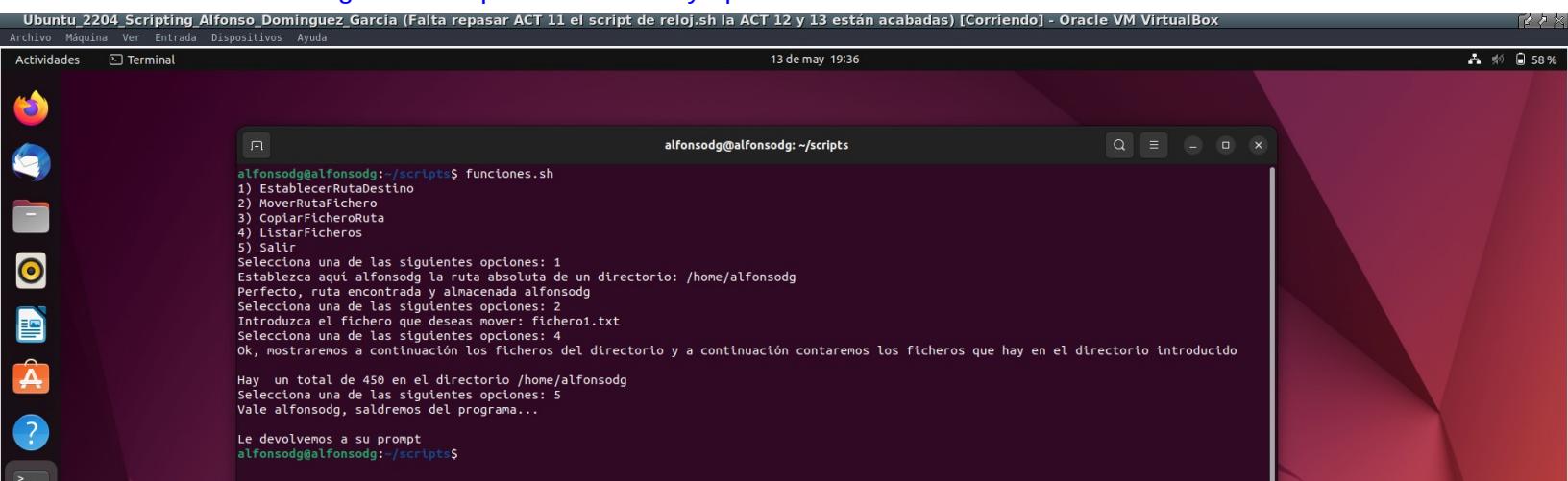
else
    echo -n "Introduzca el fichero que desea copiar"
    read ficheroCopiarUser
    copiarFichero $ficheroCopiarUser $rutaDestinoUser
fi
;;
"ListarFicheros")
if [[ -z $rutaDestinoUser ]]; then
    echo "Por favor, establezca la ruta de destino primero"
    echo
    echo "Saliendo del programa..."
    exit 3
else
    echo "Ok, mostraremos a continuación los ficheros del directorio y a continuación contaremos los ficheros que hay en el directorio introducido"
    echo
    numeroFicherosTotales=$(mostrarFicheros $rutaDestinoUser)
    echo "Hay un total de $numeroFicherosTotales en el directorio $rutaDestinoUser"
fi
;;
"Salir")
echo "Vale $USER, saldremos del programa..."
echo
```

-Ahora muestro captura del código que contiene la opción del case que será "Salir" en el cual le mostraremos al usuario el final de ejecución y le devolvemos a su prompt. Para cualquier otra cosa que el usuario haya establecido como es por ejemplo la opción número 5 pues no funcionará y saldrá del programa.



```
GNU nano 6.2
echo "Saliendo del programa..."
exit 3
else
    echo "Ok, mostraremos a continuación los ficheros del directorio y a continuación contaremos los ficheros que hay en el directorio introducido"
    echo
    numeroFicherosTotales=$(mostrarFicheros $rutaDestinoUser)
    echo "Hay un total de $numeroFicherosTotales en el directorio $rutaDestinoUser"
fi
;;
"Salir")
    echo "Vale $USER, saldremos del programa..."
    echo
    echo "Le devolvemos a su prompt"
    exit 1
;;
*)
    echo "No has escrito una opción adecuada. Saliendo del programa..."
    echo
    exit 2
;;
esac
done
```

-Finalmente muestro captura de pantalla conforme le pasamos el script “funciones.sh” que saldrá el menú principal usando un select y en este caso he marcado primero la opción número 1 que he puesto la ruta absoluta y luego he seleccionado la opción número 2 que permitirá mover cualquier fichero y conforme lo ha movido y si quisiera saber el valor total de ficheros que hay en el directorio he usado el comando **fin type -f | wc -l** y muestra el número de ficheros que hay en el directorio /home/alfonsodg o la ruta que el usuario haya puesto.



The screenshot shows a terminal window titled "alfonsodg@alfonsodg: ~/scripts". The terminal displays the following output:

```
alfonsodg@alfonsodg:~/scripts$ funciones.sh
1) EstablecerRutaDestino
2) MoverRutaFichero
3) CopiarFicheroRuta
4) ListarFicheros
5) Salir
Selecciona una de las siguientes opciones: 1
Establezca aquí alfonsodg la ruta absoluta de un directorio: /home/alfonsodg
Perfecto, ruta encontrada y almacenada alfonsodg
Selecciona una de las siguientes opciones: 2
Introduzca el fichero que deseas mover: fichero1.txt
Selecciona una de las siguientes opciones: 4
Ok, mostraremos a continuación los ficheros del directorio y a continuación contaremos los ficheros que hay en el directorio introducido
Hay un total de 450 en el directorio /home/alfonsodg
Selecciona una de las siguientes opciones: 5
Vale alfonsodg, saldremos del programa...
Le devolvemos a su prompt
alfonsodg@alfonsodg:~/scripts$
```

Webgrafía

<https://www.digitalocean.com/community/tutorials/how-to-read-and-set-environmental-and-shell-variables-on-linux-es>