



UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE
SEDE SANTO DOMINGO DE LOS TSÁCHILAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - DCCO-SS

CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN



PERIODO : 2024 51 mayo– septiembre 2024
ASIGNATURA : Sistemas Operativos
TEMA : Evaluación 1_U3
ESTUDIANTE : Jiménez Juan
NIVEL-PARALELO - NRC: NRC 15310
DOCENTE : Ing. Javier Cevallos
FECHA DE ENTREGA : 24/08/2024

SANTO DOMINGO – ECUADOR

1. Introducción	3
2. Objetivos	3
Objetivo General:	3
Objetivos Específicos:	3
3. Desarrollo / Marco Teórico/ Práctica +	4
Marco teórico:	4
Marco Practico:	5
4. Conclusiones	43
6. Bibliografía/ Referencias	44
7. Anexos:	44

1. Introducción

En la presente guía se revisará los comandos y scripts que realizaron mis compañeros como actividad individual para la presentación de estas, aquí se revisaran sus funciones y como estas fueron realizadas para su presentación para la actividad de la Evaluación 1 de la 3ra Unidad.

Un terminal es un dispositivo de hardware, ya sea de naturaleza electromecánica o electrónica, que se puede usar tanto para ingresar como para transcribir información. Esas tareas se pueden llevar a cabo bien desde un ordenador o echando mano de un sistema informático.
(Desarrollo, 2022)

2. Objetivos

Objetivo General:

- Realizar la práctica de los comandos proporcionados en la actividad en formato de guía para tener una mejor comprensión sobre los temas tratados.

Objetivos Específicos:

- Realizar la actividad de la creación de un script para la gestión de almacenamiento y la automatización de la misma.
- Realizar un informe sobre los comandos proporcionados mis compañeros sobre sus temas, analizándolos y explicándolos cuál es su función.

3. Desarrollo / Marco Teórico/ Práctica +

Marco teórico:

KERNEL:

El componente central de un sistema operativo, también conocido como núcleo (kernel), sirve como intermediario entre el hardware y el software, lo que permite que interactúen entre sí para que el sistema funcione correctamente. Para comprender su funcionamiento y maximizar su eficiencia y seguridad, es esencial estudiar sus capacidades de hardware soportadas y la programación a nivel de sistema.

Características y componentes de Kernel

- El Kernel Realiza una variedad de funciones que son esenciales para el funcionamiento del sistema operativo:
- Gestión de procesos: El Kernel controla la ejecución de los procesos del sistema, asigna tiempo de CPU, controla la concurrencia y coordina la comunicación entre procesos.
- Gestión de memoria: es responsable de asignar y liberar memoria a los procesos, garantizando un uso efectivo de los recursos y evitando conflictos de acceso a la memoria.
- Gestión de E/S (Entrada/Salida): Mantiene la comunicación fluida y sincronizada entre los dispositivos de E/S y el resto del sistema.
- Sistema de Archivos: Permite la lectura, escritura y manipulación de datos almacenados en disco mediante el acceso a sistemas de archivos.

Capacidades de Hardware Soportadas:

El kernel de un sistema operativo especifica los tipos de hardware que puede administrar y cómo interactuar con ellos. Por ejemplo, los sistemas operativos modernos deben poder reconocer y usar una amplia gama de tarjetas de red, controladores de dispositivos, dispositivos de almacenamiento y otros componentes de hardware. “Investigar las características y funcionalidades del kernel de varios sistemas operativos, incluida su arquitectura y mecanismos de comunicación con el hardware, es esencial para comprender cómo estas capacidades afectan la compatibilidad y el rendimiento del sistema.””Silberschatz, Galvin y Gagne (año desconocido).,

Marco Practico:

Juan Jiménez Script para la gestión del almacenamiento y automatización.

Para la actividad se decidió realizar un Script que realice la actividad de Gestión de almacenamiento para después automatizarlo para que este se realice cada semana los domingos.

Activities Terminal Aug 22 11:36

vboxuser@Debian2: ~

```
root@debian2:/home/vboxuser# ls
Desktop  Downloads  Pictures  ScriptEjemploAlmacenamiento.sh  Trabajos
Documents  Music      Public    Templates                      Videos
```

Primero se crea un archivo en la carpeta “Trabajos” con el uso del comando “mkdir Trabajos” donde ingresamos a la carpeta y creamos un archivo con el comando “touch ScriptEjemploAlmacenamiento.sh” para que este en formato de archivo de script, para siguiente ingresar dentro del archivo mediante el comando “nano ScriptEjemploAlmacenamiento.sh” para así realizar la edición del archivo

```
vboxuser@Debian2: ~
GNU nano 7.2                               ScriptEjemploAlmacenamiento.sh
#!/bin/bash

# Directorio de limpieza
CLEANUP_DIR="/tmp/cleanup"

# Función para mostrar el uso del disco
show_disk_usage() {
    echo "Uso del disco:"
    df -h
    echo "Mostrando con el tipo de archivo:"
    df -T
    echo
    echo "Lista de dispositivos de bloques:"
    lsblk
    echo
}

# Función para limpiar archivos temporales
cleanup_temp_files() {
    ; Read 49 lines
}

[ Help   Write Out  Where Is  Cut   Execute   Location
  Exit   Read File  Replace  Paste  Justify  Go To Line ]
```

Una vez dentro del archivo procedemos a realizar el script que realice la función de gestión del almacenamiento donde se crea un directorio donde estarán los archivos de que se limpian y se crea una función donde este realice la función de mostrar el uso del disco.

En la función se usa el comando “df –h” que muestra el uso del disco.

También se encuentra el comando “df –T” que muestra el uso del disco, pero con el tipo de archivo.

Con el comando “lsblk” se muestra la lista del almacenamiento donde se muestra por bloques.

Y con eso concluiría la función de “show_disk_usage”.



```
GNU nano 7.2           ScriptEjemploAlmacenamiento.sh *
# Función para limpiar archivos temporales
cleanup_temp_files() {

    echo "Mostrando en pantalla el contenido de las carpetas Temporales"
    ls /tmp/*
    echo
    echo "Limpiando archivos temporales..."

    # Limpiar /tmp
    sudo rm -rf /tmp/*
    echo "/tmp limpiado."

    # Limpiar /var/tmp
    sudo rm -rf /var/tmp/*
    echo "/var/tmp limpiado.

    # Muestra un mensaje de que se termino de limpiar
    echo "Limpieza completada."
    echo
}

^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File   ^V Replace    ^U Paste      ^J Justify   ^/ Go To Line
```

siguiente función es la de limpiar los archivos temporales este al empezar mostrara el contenido de las carpetas TMP ósea la carpeta temporal para siguiente mostrar un mensaje que dice que está eliminando el contenido de la capeta y con el uso del comando “rm (dirección que se quiere borrar)” y con el complemento sudo para que elimine todos los archivos que estén dentro.

```

# Función Main
main() {
    #llamada a las funciones
    show_disk_usage
    cleanup_temp_files
}

# Ejecutar la función principal
main

```

Para terminar el script tenemos la función main que se encargará de llamar a las otras funciones para ejecutarlas y finalizando el script.

```

root@Debian2:/home/vboxuser# sh ScriptEjemploAlmacenamiento.sh
Uso del disco:
Filesystem      Size  Used Avail Use% Mounted on
udev            1.5G   0    1.5G  0% /dev
tmpfs           297M  1.3M  296M  1% /run
/dev/sda1        59G  6.5G  50G  12% /
tmpfs           1.5G   0    1.5G  0% /dev/shm
tmpfs           5.0M  8.0K  5.0M  1% /run/lock
tmpfs           297M  132K  297M  1% /run/user/1000

Mostrando con el tipo de archivo:
Filesystem      Type      1K-blocks   Used Available Use% Mounted on
udev          devtmpfs     1485068      0   1485068  0% /dev
tmpfs          tmpfs       303392   1232   302160  1% /run
/dev/sda1        ext4      61328188  6749624  51430840 12% /
tmpfs          tmpfs       1516960      0   1516960  0% /dev/shm
tmpfs          tmpfs        5120       8    5112   1% /run/lock
tmpfs          tmpfs       303392    132   303260  1% /run/user/1000

```

Para ejecutarlo se usó el comando “sh (nombre del archivo)” para así realizar el scripting de la gestión de almacenamiento y realiza las funciones especificadas dentro del script.

```

Lista de dispositivos de bloques:
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda    8:0    0 60.7G  0 disk
└─sda1  8:1    0 59.7G  0 part /
└─sda2  8:2    0   1K  0 part
└─sda5  8:5    0  976M  0 part [SWAP]
sr0   11:0   1 1024M  0 rom

Mostrando en pantalla el contenido de las carpetas Temporales

Limpiando archivos temporales...
/tmp limpiado.
/var/tmp limpiado.
Limpieza completada.

```

```
vboxuser@Debian2:~
```

```
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 3 * * 0 /home/vboxuser/ScriptEjemploAlmacenamiento.sh
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^V Replace ^U Paste ^J Justify ^L Go To Line

En el apartado de automatización tenemos que con el comando “crontab –e” que se encarga de poder editar el archivo crontab que es el encargado de realizar las actividades de manera automática según el tiempo especificado en este caso con el comando ingresado se realiza la función del Script de gestión de almacenamiento cada domingo a las 3 am.

Anderson Lara Script para el ordenamiento de carpetas:

The screenshot shows a terminal window titled "Terminal" with the command "debianjuanj@10: ~/trabajos". The script "AndersonLara.sh" is being edited in nano 7.2. The script content is as follows:

```
GNU nano 7.2          AndersonLara.sh
#!/bin/bash

# Directorio donde se encuentran los archivos a ordenar
DIRECTORIO_ORIGEN="$HOME/Documents/archives" # Cambia esto a la ruta de tu directorio

# Cambiar al directorio de origen
cd "$DIRECTORIO_ORIGEN" || { echo "No se pudo cambiar al directorio $DIRECTORIO_ORIGEN"

# Iterar sobre todos los archivos en el directorio
for archivo in *; do
    # Verificar si es un archivo regular
    if [ -f "$archivo" ]; then
        # Obtener la extensión del archivo
        extension="${archivo##*.}"
        # Crear la carpeta con el nombre de la extensión si no existe
        [ ! -d "$extension" ] && mkdir "$extension"
        # Mover el archivo a la carpeta correspondiente
        mv "$archivo" "$extension/"
        echo "Moviendo archivo a la carpeta $extension/"
    fi
done
echo "Tarea completada. Todos los archivos han sido ordenados por su extensión."

```

The terminal shows the script being run, with the message "[Read 23 lines]" indicating the number of lines read. At the bottom, standard nano key bindings are listed.

done
echo "Tarea completada. Todos los archivos han sido ordenados por su extensión."
[
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

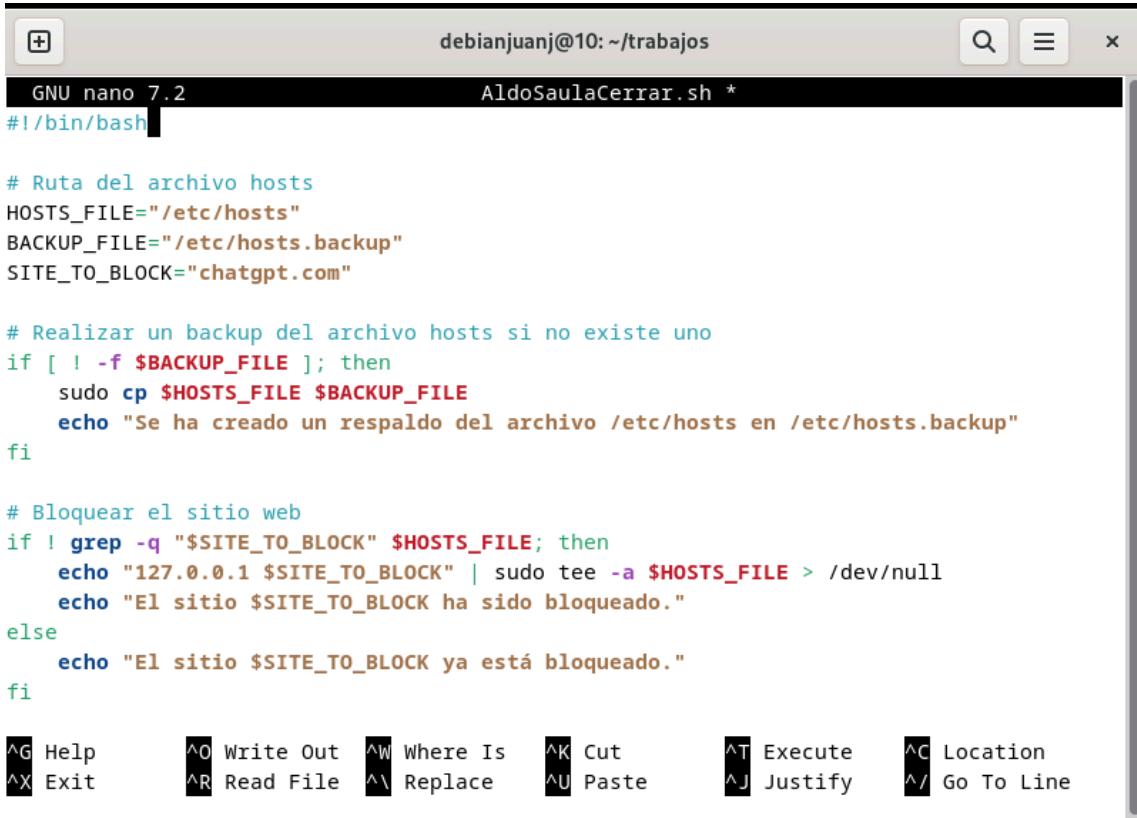
El Script del compañero realiza la función de ordenar archivos dependiendo la extensión de la misma donde crea varias carpetas con el nombre de la extensión, en la imagen podemos ver que para empezar añade en un valor el directorio de archivos que se encuentra dentro de la carpeta documentos para siguiente cambiar el directorio de origen en caso que no pueda mostrará un mensaje que no pudo.

Siguiente crea un bucle for para que este ordene los archivos donde primero verifica el archivo, su extensión y lo mueve a la carpeta correspondiente, en caso de no existir la carpeta esté la crea, al finalizar el bucle este mostrará un mensaje diciendo que a terminado de ordenar.

```
debianjuanj@10:~/Documents/archives$ cd
debianjuanj@10:~$ cd trabajos/
debianjuanj@10:~/trabajos$ sh AndersonLara.sh
Moviendo SistemasOperativos.sh a la carpeta sh/
Moviendo hola.txt a la carpeta txt/
Moviendo papu.txt a la carpeta txt/
Tarea completada. Todos los archivos han sido ordenados por su extensi n.
debianjuanj@10:~/trabajos$ cd
debianjuanj@10:~$ ls
bad-apple Documents Music Public trabajos
Desktop Downloads Pictures Templates Videos
debianjuanj@10:~$ cd Documents/archives/
sh/ txt/
debianjuanj@10:~$ cd Documents/
debianjuanj@10:~/Documents$ ls
archives
debianjuanj@10:~/Documents$ cd archives/
debianjuanj@10:~/Documents/archives$ ls
sh txt
debianjuanj@10:~/Documents/archives$ cd txt/
debianjuanj@10:~/Documents/archives/txt$ ls
holo.txt papu.txt
```

Ejecutando el script podemos ver que muestra que los archivos fueron xvideos dentro de la carpeta con el nombre de la extensi n en este caso se crearon 2 carpetas 1 de archivos txt y otra para archivos sh donde si ingresamos dentro de ellas podemos ver que si se encuentran los archivos hay.

Aldo Saula Script para el bloqueo a páginas Web:



The screenshot shows a terminal window titled 'AldoSaulaCerrar.sh *'. The script content is as follows:

```
GNU nano 7.2
#!/bin/bash

# Ruta del archivo hosts
HOSTS_FILE="/etc/hosts"
BACKUP_FILE="/etc/hosts.backup"
SITE_TO_BLOCK="chatgpt.com"

# Realizar un backup del archivo hosts si no existe uno
if [ ! -f $BACKUP_FILE ]; then
    sudo cp $HOSTS_FILE $BACKUP_FILE
    echo "Se ha creado un respaldo del archivo /etc/hosts en /etc/hosts.backup"
fi

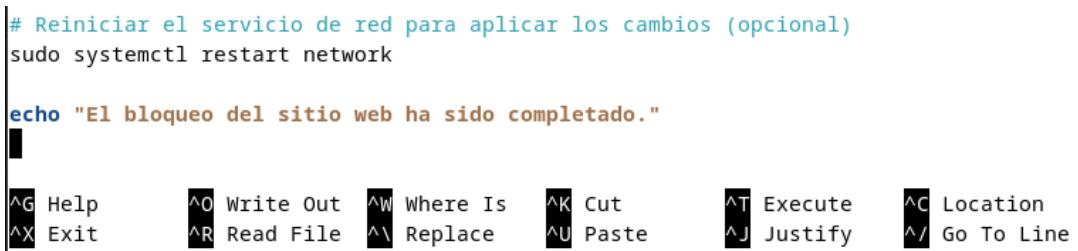
# Bloquear el sitio web
if ! grep -q "$SITE_TO_BLOCK" $HOSTS_FILE; then
    echo "127.0.0.1 $SITE_TO_BLOCK" | sudo tee -a $HOSTS_FILE > /dev/null
    echo "El sitio $SITE_TO_BLOCK ha sido bloqueado."
else
    echo "El sitio $SITE_TO_BLOCK ya está bloqueado."
fi
```

At the bottom of the terminal window, there is a key mapping legend:

^G	Help	^O	Write Out	^W	Where Is	^K	Cut	^T	Execute	^C	Location
^X	Exit	^R	Read File	^V	Replace	^U	Paste	^J	Justify	^/	Go To Line

El siguiente script es el presentado por el compañero Aldo Saula el cual consiste en el bloqueo de páginas web, en la imagen podemos ver como el script al inicio crea variables con los archivos del host donde esté obtiene las direcciones de los archivos y la página que desea bloquear.

Se crea un if que sirve para crea un respaldo de los archivos host en caso de no existir alguno, el otro if es el que se encarga de bloquear el sitio por lo cual obtiene la dirección ip de usuario y llama a la variable del sitio donde mediante sudo bloquea la conexión de la misma, en caso de intentarlo de nuevo dirá que este sitio ya está bloqueado.



```
# Reiniciar el servicio de red para aplicar los cambios (opcional)
sudo systemctl restart network

echo "El bloqueo del sitio web ha sido completado."
```

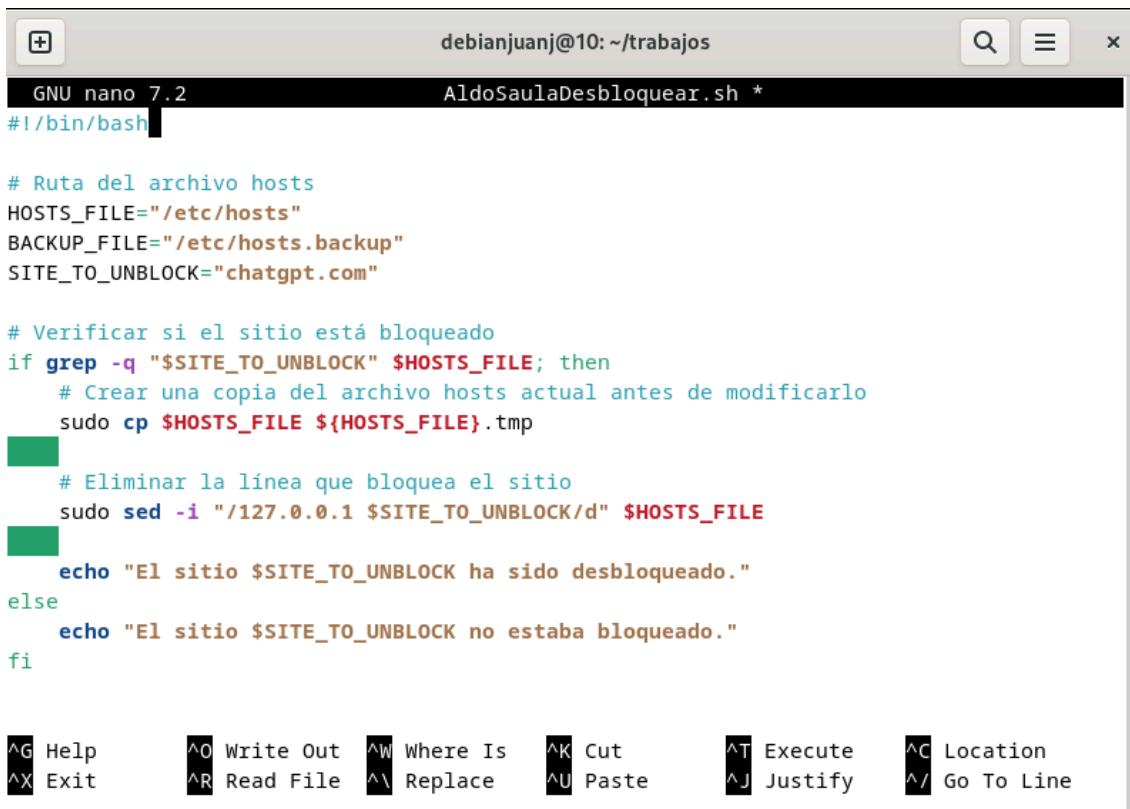
At the bottom of the terminal window, there is a key mapping legend:

^G	Help	^O	Write Out	^W	Where Is	^K	Cut	^T	Execute	^C	Location
^X	Exit	^R	Read File	^V	Replace	^U	Paste	^J	Justify	^/	Go To Line

La siguiente imagen es la conclusión del script que se encarga de cerrar la conexión con la página web deseada donde con un comando este reinicia las conexiones para aplicar los cambios y para finalizar muestra un mensaje que dice que el script realizó su función.

```
root@10:~# cd /home/debianjuanj/trabajos/
root@10:/home/debianjuanj/trabajos# sh AldoSaulaCerrar.sh
Se ha creado un respaldo del archivo /etc/hosts en /etc/hosts.backup
El sitio chatgpt.com ha sido bloqueado.
Failed to restart network.service: Unit network.service not found.
El bloqueo del sitio web ha sido completado.
root@10:/home/debianjuanj/trabajos#
```

En la ejecución podemos ver que crea el respaldo debido a que en el equipo al parecer no existían los archivos de Host y a siguiente bloquear la pagina “chatgpt.com” y realizando su función.



```
GNU nano 7.2          debianjuanj@10: ~/trabajos
AldoSaulaDesbloquear.sh *

#!/bin/bash

# Ruta del archivo hosts
HOSTS_FILE="/etc/hosts"
BACKUP_FILE="/etc/hosts.backup"
SITE_TO_UNBLOCK="chatgpt.com"

# Verificar si el sitio está bloqueado
if grep -q "$SITE_TO_UNBLOCK" $HOSTS_FILE; then
    # Crear una copia del archivo hosts actual antes de modificarlo
    sudo cp $HOSTS_FILE ${HOSTS_FILE}.tmp
    # Eliminar la línea que bloquea el sitio
    sudo sed -i "/127.0.0.1 $SITE_TO_UNBLOCK/d" $HOSTS_FILE
    echo "El sitio $SITE_TO_UNBLOCK ha sido desbloqueado."
else
    echo "El sitio $SITE_TO_UNBLOCK no estaba bloqueado."
fi

^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line
```

Aquí tenemos el script para desbloquear páginas web su codificación es bastante similar por ende vamos a destacar sus funciones nuevas.

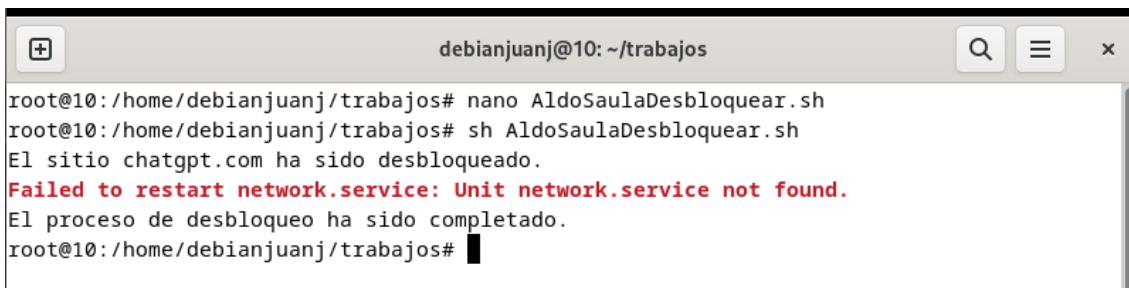
En este script hay un if que verifica que la pagina este bloqueada donde en caso de no estar bloqueada la pagina dira que la pagina no se encontraba bloqueada y tambien crea un respaldo antes de modificar los archivos Host.

```
# Reiniciar el servicio de red para aplicar los cambios (opcional)
sudo systemctl restart network

echo "El proceso de desbloqueo ha sido completado."
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Asi mismo tiene al final el comando para reiniciar los servicios de red para aplicar los cambios realizados y mostrará un mensaje de que se ha completado el desbloqueo.



```
debianjuanj@10: ~/trabajos
root@10:/home/debianjuanj/trabajos# nano AldoSaulaDesbloquear.sh
root@10:/home/debianjuanj/trabajos# sh AldoSaulaDesbloquear.sh
El sitio chatgpt.com ha sido desbloqueado.
Failed to restart network.service: Unit network.service not found.
El proceso de desbloqueo ha sido completado.
root@10:/home/debianjuanj/trabajos#
```

En este script al ejecutarlo este mostrará los mensajes previamente explicados y dirá que el proceso a sido completado lo que significa que ya se puede ingresar a la página web de nuevo.

Angel Rodriguez Script para la sincronización automática entre archivos y automatización:



```
Activities Terminal Aug 23 13:37
debianjuanj@10: ~/trabajos
GNU nano 7.2 AngelRodriguez.sh *
#Directorios de origen y destino
DIR_ORIGEN="/home/debianjuanj/source" #modificar de acuerdo al uso
DIR_DESTINO="/home/debianjuanj/backup" #modificar de acuerdo al uso

#Comando rsync para sincronizar los directorios
rsync -av --delete $DIR_ORIGEN/ $DIR_DESTINO

#Registro de la sincronizacion
echo "$(date): Sincronizacion completa entre $DIR_ORIGEN y $DIR_DESTINO" >> ~/sincro
```

El compañero Angel Rodriguez nos presento un script para la sincronización automática entre archivos el cual consiste que entre directorios los archivos se van a copiar al mismo tiempo, en el script podemos ver como este coje los 2 directorios que se van a usar y mediante el comando “rsync -av” que es el encargado de sincronizar los archivos entre las carpetas para finalizar con un mensaje diciendo que se a realizado la sincronización.



```
Activities Terminal Aug 23 13:41 debianjuanj@10: ~/trabajos
root@10:/home/debianjuanj/trabajos# touch AngelRodriguez.sh
root@10:/home/debianjuanj/trabajos# nano AngelRodriguez.sh
root@10:/home/debianjuanj/trabajos# sh AngelRodriguez.sh
AngelRodriguez.sh: 6: rsync: not found
root@10:/home/debianjuanj/trabajos#
```

En mi caso sale un error con el rsync que no se encuentra, incluso intentando instalarlo manualmente no me ha permitido pero en teoría el script funcionaria y deberá realizar la función de sincronizar los archivos de las 2 carpetas seleccionadas.



```
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
30 15 * * * /home/usuario/hola.sh
23 30 * * * apt-get update -y > /tmp/cron.txt
0 0 * * * /home/debianjuanj/trabajos/AngelRodriguez.sh
```

Para terminar en la parte de automatización se agregan las 2 últimas líneas al archivo crontab que se encarga de realizar scripts que se encuentre dependiendo la hora que se especifique en este caso sincronizará todos los días a las 4 de la madrugada.

Joseph Franco, crear una automatización y registro de actividades del sistema:



The screenshot shows a terminal window titled "Terminal" with the command "GNU nano 7.2" at the top. The file path is "/root/system_log.sh". The script content is as follows:

```
#!/bin/bash

#! Archivo de log
LOG_FILE="/var/log/system_activity.log"

#! Registrar la fecha y hora actuales
echo "----- Registro de actividades: $(date + '%Y-%m-%d %H:%M:%S') -----" >> $LOG_FILE

#! Registrar los procesos en ejecucion

ps aux >> $LOG_FILE

#! Anadir una linea en blanco para separar entradas
echo "" >> $LOG_FILE
```

At the bottom of the terminal window, there is a status message "[Wrote 14 lines]" and a set of keyboard shortcuts:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^V Replace	^U Paste	^J Justify	^/ Go To Line

El compañero propuso el configurar el archivo System.log.sh para que este nos muestre la actividad del sistema con fecha, y hora incluidas, además que lo guarde en un archivo log

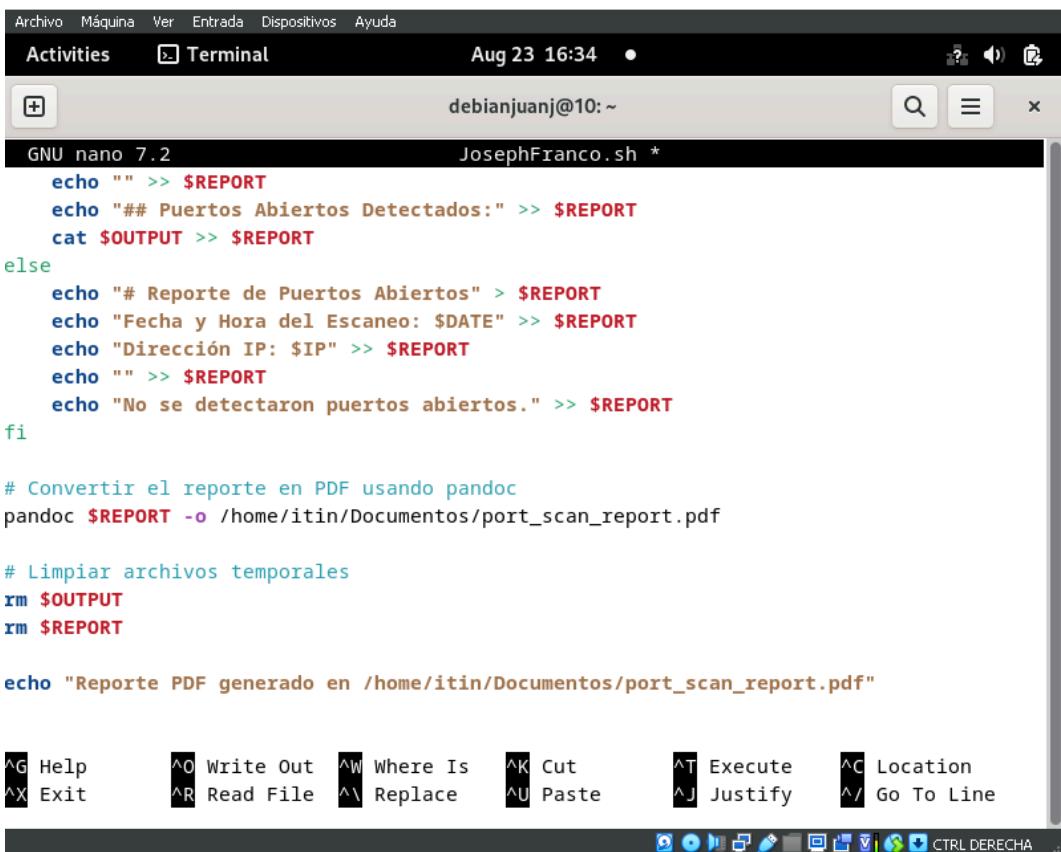


The screenshot shows a terminal window titled "Terminal" with the command "GNU nano 7.2". The window title bar also displays "Aug 23 16:31" and the user "debianjuanj@10: ~". The terminal content is a bash script named "JosephFranco.sh". The script defines variables for IP ("IP='127.0.0.1'"), date ("DATE=\$(date '+%Y-%m-%d %H:%M:%S')"), output file ("OUTPUT='/home/itin/Documentos/port_scan_result.txt'"), and report file ("REPORT='/home/itin/Documentos/port_scan_report.md'"). It then runs an nmap scan on ports 1-65535 and checks for open ports using grep. A header "# Reporte de Puertos Abiertos" is added to the report file. The bottom of the terminal shows nano key bindings:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^V Replace	^U Paste	^J Justify	^/ Go To Line

En el siguiente script vemos que para guardar en un archivo pdf se necesita se guarda un valor de la ip, la fecha, el lugar donde se almacenan los archivos del escaneo, el archivo donde se guarda el reporte en formato Markdown y realiza el escaneo de los puertos del equipo que se encuentre abiertos.

con un if se verifica que los puertos esten abiertos



The screenshot shows a terminal window titled "Terminal" with the command "JosephFranco.sh *". The script content is as follows:

```
GNU nano 7.2
echo "" >> $REPORT
echo "## Puertos Abiertos Detectados:" >> $REPORT
cat $OUTPUT >> $REPORT
else
    echo "# Reporte de Puertos Abiertos" > $REPORT
    echo "Fecha y Hora del Escaneo: $DATE" >> $REPORT
    echo "Dirección IP: $IP" >> $REPORT
    echo "" >> $REPORT
    echo "No se detectaron puertos abiertos." >> $REPORT
fi

# Convertir el reporte en PDF usando pandoc
pandoc $REPORT -o /home/itin/Documentos/port_scan_report.pdf

# Limpiar archivos temporales
rm $OUTPUT
rm $REPORT

echo "Reporte PDF generado en /home/itin/Documentos/port_scan_report.pdf"
```

At the bottom of the terminal window, there is a menu bar with keyboard shortcuts:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^V Replace	^U Paste	^J Justify	^/ Go To Line

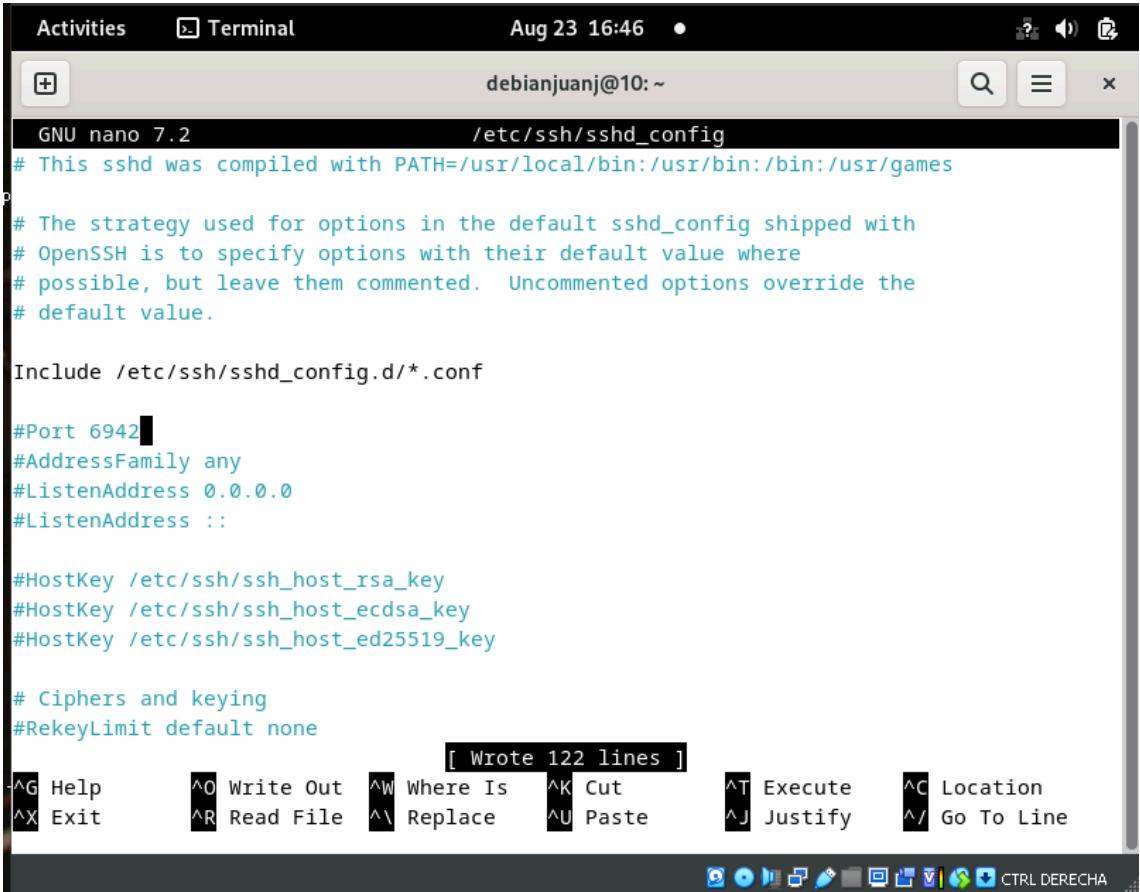
Below the menu bar, there is a toolbar with various icons.

En caso de no estarlos este dira que no se detectaron puertos abiertos

Otra función que es con el comando “pandoc” que se encarga de convertir los reportes en formato PDF y para terminar eliminar los archivos donde se guardaban temporalmente los informes.

Al final mostrar un texto diciendo que el pdf se a generado en una carpeta especificada.

Jennifer Guerra Cambió de puertos en servidor SSH:



The screenshot shows a terminal window titled "Terminal" with the command "debianjuanj@10: ~". The file being edited is "/etc/ssh/sshd_config". The terminal displays the configuration file's contents, including the line "#Port 6942" which is highlighted. The bottom of the terminal shows the nano editor's command-line interface with various keyboard shortcuts and a status bar indicating "[Wrote 122 lines]".

```
GNU nano 7.2          /etc/ssh/sshd_config
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 6942
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none
[ Wrote 122 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File   ^\ Replace     ^U Paste      ^J Justify    ^/ Go To Line
CTRL DERECHA
```

Para empezar la compañera propone realizar configuracion en los puertos SSH para las conexiones entre equipos, en la imagen se ve que se realiza el cambio del puerto por defecto “22” por el “6942”.

```
GNU nano 7.2          /etc/ssh/sshd_config
#SyslogFacility AUTH
LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
[ Wrote 122 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File   ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

```
GNU nano 7.2          /etc/ssh/sshd_config
# Logging
#SyslogFacility AUTH
LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin no
#StrictModes yes
#MaxAuthTries 69
#MaxSessions 42

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none
[ Wrote 122 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File   ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line
```

Con el ahora procedemos a cambiar la autentificación para que no permita el ingreso mediante root, así mismo cambiamos el máximo de intento de autenticación de 6 a 69 y el número de secciones máximos de 10 a 42.

```
Activities Terminal Aug 23 16:57 • debianjuanj@10: ~
root@10:/home/debianjuanj/trabajos# sudo chage -l debianjuanj
Last password change : Jul 23, 2024
Password expires : never
Password inactive : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
root@10:/home/debianjuanj/trabajos#
```

Para comprobar lo último realizado se usa el comando “chage -l (nombreusuario)” donde nos muestra valores sobre la seguridad del equipo y la cantidad de veces que ha expirado la sección en caso de expirar o de expirar la contraseña, entre otros.

Miguel Molina: Instalacion y configuracion de Docker Network:

```
Activities Terminal Aug 23 17:04 • debianjuanj@10: ~
GNU nano 7.2 MiguelMolina.sh
#!/bin/bash

# Actualizar el sistema
echo "Actualizando el sistema..."
sudo apt-get update

# Instalar paquetes necesarios
echo "Instalando paquetes necesarios..."
sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common

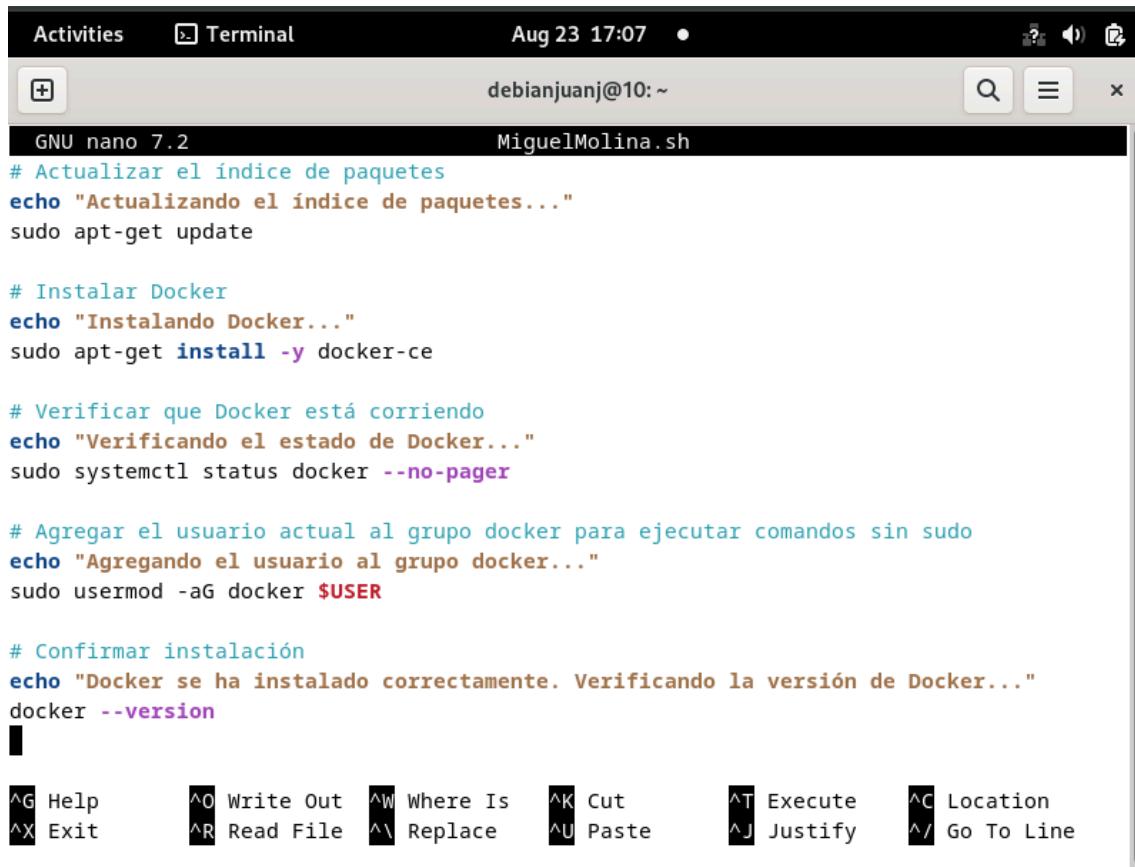
# Agregar la clave GPG de Docker
echo "Agregando la clave GPG de Docker..."
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -

# Agregar el repositorio de Docker
echo "Agregando el repositorio de Docker..."
echo "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >> /dev/null

# Actualizar el índice de paquetes
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^V Replace ^U Paste ^J Justify ^/ Go To Line
```

Para empezar con el Script se actualizan la librería de apt para siguiente instalar las librerías restantes necesarias para la actividad como lo son apt-transports-https entre otros

Luego se agrega la clave GPG de Docker que es la clave nuestra de docker para terminar agregando el repositorio del docker al equipo.



The screenshot shows a terminal window titled "Terminal" with the date and time "Aug 23 17:07". The user is "debianjuanj@10: ~" and the file being edited is "MiguelMolina.sh". The script content is as follows:

```
GNU nano 7.2 MiguelMolina.sh
# Actualizar el indice de paquetes
echo "Actualizando el índice de paquetes..."
sudo apt-get update

# Instalar Docker
echo "Instalando Docker..."
sudo apt-get install -y docker-ce

# Verificar que Docker está corriendo
echo "Verificando el estado de Docker..."
sudo systemctl status docker --no-pager

# Agregar el usuario actual al grupo docker para ejecutar comandos sin sudo
echo "Agregando el usuario al grupo docker..."
sudo usermod -aG docker $USER

# Confirmar instalación
echo "Docker se ha instalado correctamente. Verificando la versión de Docker..."
docker --version
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^V Replace	^U Paste	^J Justify	^/ Go To Line

En la siguiente parte del script se realiza la función de actualizar la lista de paquetes de apt, para consiguiente instalar la librería para los dockers denominada como “docker-ce”, se verifica que el docker este ejecutándose después de la instalación con el comando “systemctl status docker –no-pager”.

Se agrega al usuario actual al grupo de los que puedan ejecutar comandos de docker sin root o sudo para finalizar con la confirmación de la instalación de la librería de docker.

```
Activities Terminal Aug 23 17:13 • debianjuanj@10:~  
root@10:/home/debianjuanj/trabajos# nano MiguelMolina.sh  
root@10:/home/debianjuanj/trabajos# sh MiguelMolina.sh  
Actualizando el sistema...  
Ign:1 https://apt.fury.io/ascii-image-converter InRelease  
Ign:2 https://apt.fury.io/ascii-image-converter Release  
Ign:3 https://apt.fury.io/ascii-image-converter Packages  
Ign:4 https://apt.fury.io/ascii-image-converter Translation-en_US  
Ign:5 https://apt.fury.io/ascii-image-converter Translation-en  
Get:3 https://apt.fury.io/ascii-image-converter Packages  
Ign:4 https://apt.fury.io/ascii-image-converter Translation-en_US  
Ign:5 https://apt.fury.io/ascii-image-converter Translation-en  
Ign:4 https://apt.fury.io/ascii-image-converter Translation-en  
Ign:5 https://apt.fury.io/ascii-image-converter Translation-en  
Ign:4 https://apt.fury.io/ascii-image-converter Translation-en_US  
Ign:5 https://apt.fury.io/ascii-image-converter Translation-en  
Ign:4 https://apt.fury.io/ascii-image-converter Translation-en  
Ign:5 https://apt.fury.io/ascii-image-converter Translation-en  
Ign:4 https://apt.fury.io/ascii-image-converter Translation-en_US  
Ign:5 https://apt.fury.io/ascii-image-converter Translation-en  
Fetched 15.4 kB in 2s (6,916 B/s)  
Reading package lists... Done  
Instalando paquetes necesarios...  
Reading package lists... Done
```

```
Activities Terminal Aug 23 17:13 • debianjuanj@10:~  
Ign:6 https://apt.fury.io/ascii-image-converter Translation-en  
Reading package lists... Done  
W: GPG error: https://download.docker.com/linux/debian bookworm InRelease: The following signatures couldn't be verified because the public key is not available: NO_PUBKEY 7E A0A9C3F273FCDB8  
E: The repository 'https://download.docker.com/linux/debian bookworm InRelease' is not signed.  
N: Updating from such a repository can't be done securely, and is therefore disabled by default.  
N: See apt-secure(8) manpage for repository creation and user configuration details.  
Instalando Docker...  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
E: Unable to locate package docker-ce  
Verificando el estado de Docker...  
Unit docker.service could not be found.  
Agregando el usuario al grupo docker...  
usermod: group 'docker' does not exist  
Docker se ha instalado correctamente. Verificando la versión de Docker...  
MiguelMolina.sh: 38: docker: not found  
Instalación completa. Por favor, cierre la sesión y vuelva a iniciarla para aplicar los cambios del grupo docker.  
root@10:/home/debianjuanj/trabajos#
```

Al ejecutar el script realizará todas las operaciones que se encontraban dentro del script y que se especificó anteriormente en la revisión del mismo y cuál era su utilidad.

```
Activities Text Editor Aug 23 17:19
Open comandosMiguelMolina.txt ~/trabajos
Comandos para crear red docker segun el driver-----
docker network create --driver bridge mi_red_personalizada

docker network create --driver overlay mi_red_overlay

docker network create -d macvlan \
--subnet=192.168.1.0/24 \
--gateway=192.168.1.1 \
-o parent=eth0 \
mi_red_macvlan

comando de creacion de red con configuracion avanzada-----
docker network create --driver bridge \
--subnet 192.168.1.0/24 \
--ip-range 192.168.1.128/25 \
mi_red_avanzada

comando para inspeccionar la configuracion de la red creada -----
docker network inspect mi_red_personalizada

comandos que ejecutan un contenedor en una red-----

```

El archivo contiene los comandos para la creacion de una red docker donde al inicio se crea 2 redes, una red puente con puerto personalizado y otro de una interfaz, luego se crea otra red docker tipo lan.

Otro comando para la creación de un driver de puerto puente con datos personalizados y con el comando “docker network inspect (dirección del puerto personalizado)” se realiza una revisión del puerto creado aunque cambiando los valores se puede verificar de cualquier puerto.

```
comandos que ejecutan un contenedor en una red-----
(si no hay contenedores añadidos no se pueden añadir)
docker run -d --name mi_contenedor --network mi_red_personalizada nginx

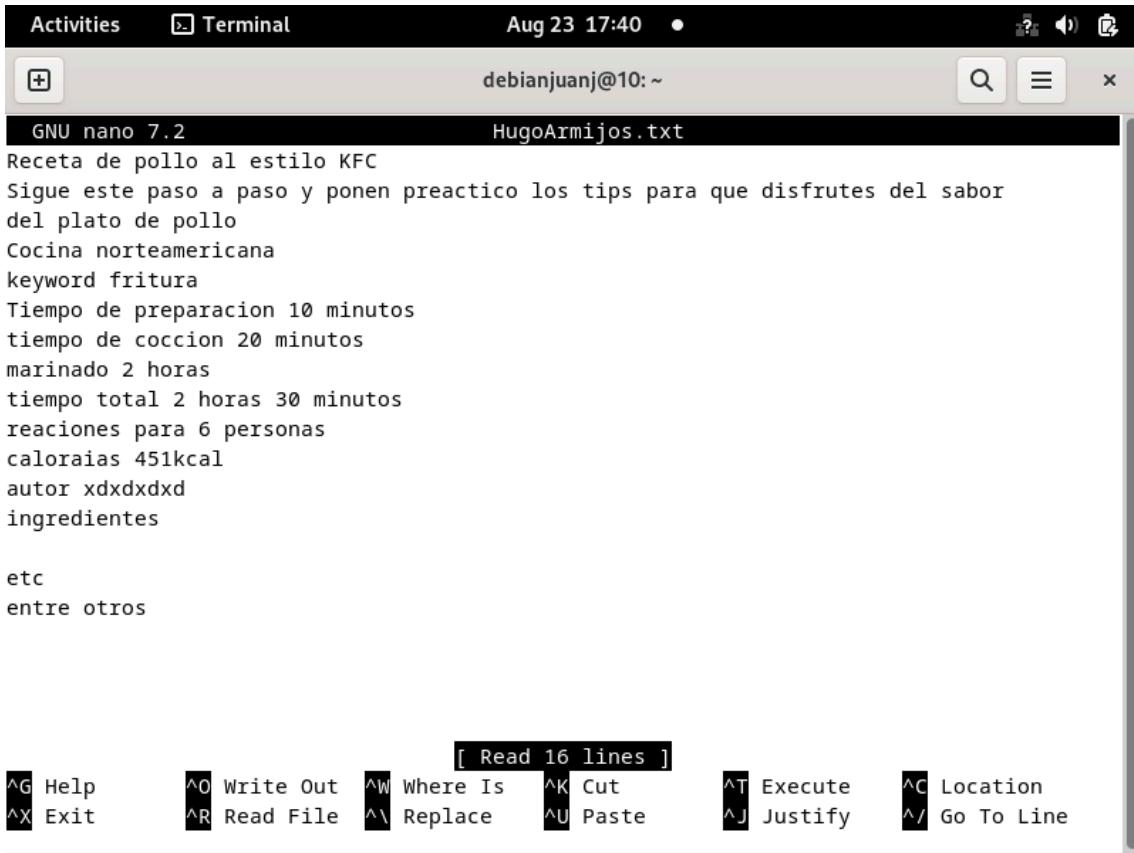
docker network connect mi_red_personalizada mi_contenedor

docker network disconnect mi_red_personalizada mi_contenedor
```

Estos son los comandos para ejecutar una red docker:

El primer comando sirve para agregar dockers en caso de no estar agregados o que se quieran agregar a la red docker.

El segundo comando sirve para realizar la conexión entre la red personalizada con el docker y el último comando sirve para realizar lo contrario al anterior, osea la desconexión de la red al docker.



The screenshot shows a terminal window titled "Terminal" with the date and time "Aug 23 17:40". The working directory is "debianjuanj@10: ~". The file being edited is "HugoArmijos.txt". The content of the file is:

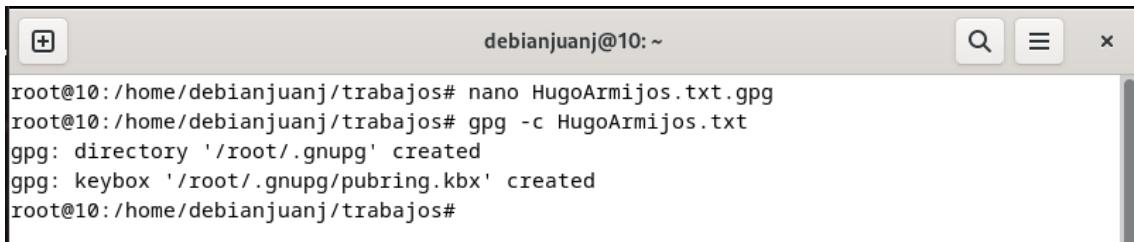
```
GNU nano 7.2          HugoArmijos.txt
Receta de pollo al estilo KFC
Sigue este paso a paso y ponen preactico los tips para que disfrutes del sabor
del plato de pollo
Cocina norteamericana
keyword fritura
Tiempo de preparacion 10 minutos
tiempo de coccion 20 minutos
marinado 2 horas
tiempo total 2 horas 30 minutos
reaciones para 6 personas
caloraias 451kcal
autor xxdxdxd
ingredientes

etc
entre otros
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts:

- ^G Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^T Execute
- ^C Location
- ^X Exit
- ^R Read File
- ^V Replace
- ^U Paste
- ^J Justify
- ^L Go To Line

[Read 16 lines]



The screenshot shows a terminal window titled "Terminal" with the date and time "Aug 23 17:40". The working directory is "debianjuanj@10: ~". The command entered is:

```
root@10:/home/debianjuanj/trabajos# nano HugoArmijos.txt.gpg
root@10:/home/debianjuanj/trabajos# gpg -c HugoArmijos.txt
gpg: directory '/root/.gnupg' created
gpg: keybox '/root/.gnupg/pubring.kbx' created
root@10:/home/debianjuanj/trabajos#
```

Una vez creado el archivo mediante el uso del comando “gpg -c (nombrearchivo)” encriptamos el archivo para así solo poder ingresar nosotros mediante la contraseña ingresada y resguardar nuestros datos, entre otros.

A screenshot of a terminal window titled "Terminal" at "Aug 23 17:44". The window shows a file named "HugoArmijos.txt.gpg" with the following content:

```
GNU nano 7.2                                     HugoArmijos.txt.gpg
?
^D      ^C^B f ??, ???:^AT^T??$ ??^?k^[ ?]EQw??P18K ??7??
? ???:^K ??b^_??^T ??Yb ??W ??$w, ??2 ??N?^?EQ"5> ?? ??| ?
???:^XO ??A ??D ?? V ??Z _?L! ??> O ?? a$G?1>{ ??A ??
```

Esto es lo que se vera cuando se quiera ingresar al archivo lo cual es ilegible para las personas debido a que no se pueden ver la mayoría de caracteres y no está en un lenguaje normal.

A screenshot of a terminal window titled "Terminal" at "Aug 23 17:46". The window shows the command being run:

```
root@10:/home/debianjuanj/trabajos# gpg -o HugoArmijos2.txt -d HugoArmijos.txt.gpg
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
root@10:/home/debianjuanj/trabajos# nano HugoArmijos2.txt
root@10:/home/debianjuanj/trabajos#
```

A screenshot of a terminal window titled "Terminal" at "Aug 23 17:47". The window shows the decrypted file "HugoArmijos2.txt" with the following content:

```
GNU nano 7.2                                     HugoArmijos2.txt
Receta de pollo al estilo KFC
Sigue este paso a paso y ponen preactico los tips para que disfrutes del sabor
del plato de pollo
Cocina norteamericana
keyword fritura
Tiempo de preparacion 10 minutos
tiempo de coccion 20 minutos
marinado 2 horas
tiempo total 2 horas 30 minutos
reaciones para 6 personas
caloraias 451kcal
autor xddxdxd
ingredientes

etc
entre otros
```

[Read 16 lines]

Keyboard shortcuts displayed at the bottom:

- ^G Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^T Execute
- ^C Location
- ^X Exit
- ^R Read File
- ^V Replace
- ^U Paste
- ^J Justify
- ^L Go To Line

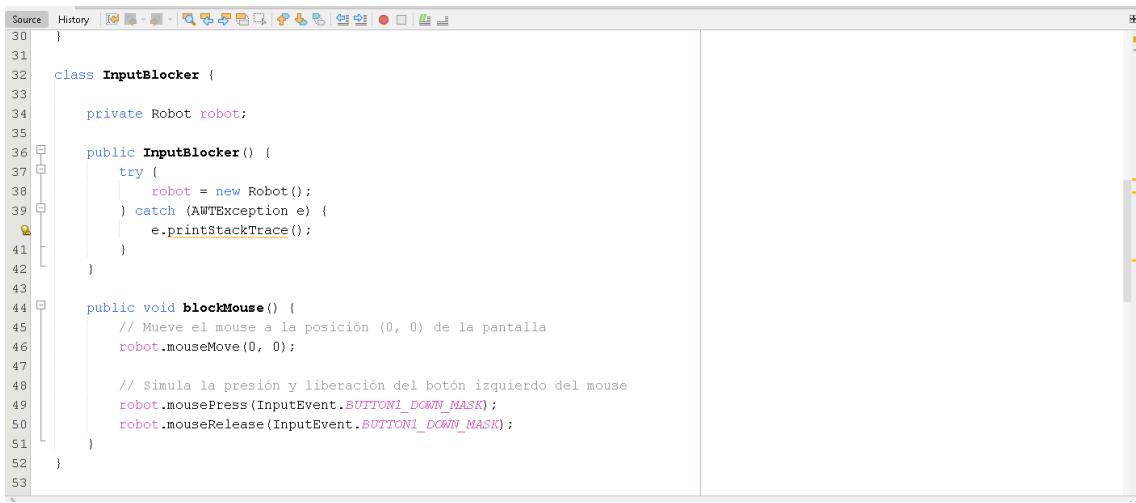
Para desencriptar un archivo se usa el comando “gpg -o (archivo que tendra el desencriptado) -d (archivo encriptado)” donde este copiara el contenido del archivo encriptado dentro de otro que no estará encriptado.

Adonis Alegria: Vulnerabilidades y malware:

El compañero desarrolló un “malware” más como vulnerabilidad que inhabilita al afectado la capacidad de tener control del equipo obligando al puntero del ratón a quedarse en la parte superior izquierda y evitando moverlo a cualquier dirección además de inhabilitar el poder escribir algún tipo de texto.

En el código podemos ver que llama varias librerías en específico librerías para la creación de ventanas y movimiento de equipo automático como sería el “awt.Robot”

En la clase main crea los atributos que llaman a las librerías y llama a las funciones que realizará el malware

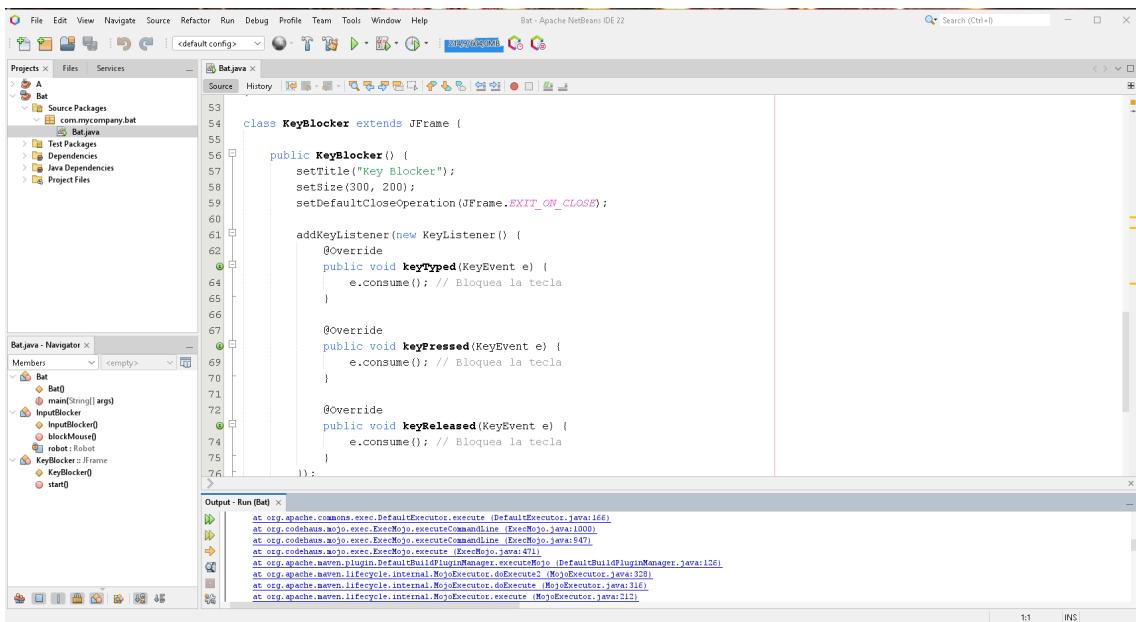


```

30 }
31
32 class InputBlocker {
33
34     private Robot robot;
35
36     public InputBlocker() {
37         try {
38             robot = new Robot();
39         } catch (AWTException e) {
40             e.printStackTrace();
41         }
42
43     }
44
45     public void blockMouse() {
46         // Mueve el mouse a la posición (0, 0) de la pantalla
47         robot.mouseMove(0, 0);
48
49         // Simula la presión y liberación del botón izquierdo del mouse
50         robot.mousePress(InputEvent.BUTTON1_DOWN_MASK);
51         robot.mouseRelease(InputEvent.BUTTON1_DOWN_MASK);
52     }
53 }

```

Primero se crea una clase denominada `inputBlocker` el cual se encarga de bloquear los botones de usuario y el movimiento del mouse, creando una función dentro de la clase denominado “`inputblocker`” que se encarga de bloquear los botones del teclado y otra función llamada “`blockmouse`” que como su nombre indica se encarga de bloquear el movimiento del mouse en este caso obliga al mouse a dirigirse a la posición 0 0 de la pantalla por eso se va a la parte superior izquierda de la pantalla.

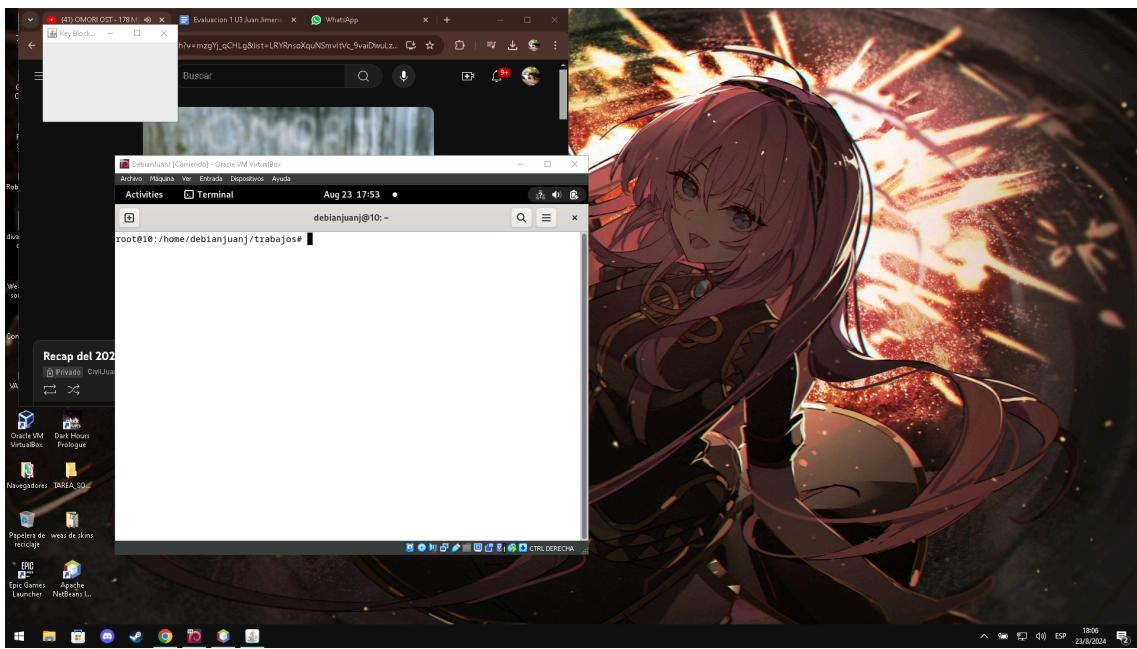


```

53
54 class KeyBlocker extends JFrame {
55
56     public KeyBlocker() {
57         setTitle("Key Blocker");
58         setSize(300, 200);
59         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
60
61         addKeyListener(new KeyListener() {
62             @Override
63             public void keyTyped(KeyEvent e) {
64                 e.consume(); // Bloquea la tecla
65             }
66
67             @Override
68             public void keyPressed(KeyEvent e) {
69                 e.consume(); // Bloquea la tecla
70             }
71
72             @Override
73             public void keyReleased(KeyEvent e) {
74                 e.consume(); // Bloquea la tecla
75             }
76         });
77     }
78 }

```

Por otro lado se crea otra clase llamada “`inputBlocker`” donde se tiene una función llamada `keyblocker` que se encarga de bloquear el colocado de letras por el usuario afectado el cual crea una ventana que va bloqueando las letras que se ingresen, esto funcionando durante 1 min.



Al momento de ejecutar el programa el mouse se me movio rápidamente a la parte superior izquierda de la pantalla donde se quedo y volvía en caso de querer mover el mouse, por otro lado también no dejaba mover bien entre las pestañas debido a que estas se cerraban y para tomar captura se tuvo que usar un botón especial para capturar la pantalla.

Freddy Jiménez: Almacenamiento, Automatización y Script

El ejemplo del compañero inicia usando los comandos individualmente esta parte la saltaremos debido a que es exactamente lo mismo que se realizará cuando se cree el script.



```
GNU nano 7.2          FreddyJimenez.sh *
#!/bin/bash
# Define el directorio de montaje para las comprobaciones
DIRECTORIO_MONTADO="/mnt"
# Mostrar uso de disco
echo "Uso del espacio en disco:"
df -h
echo
# Identificar los directorios que más espacio ocupan
echo "Directorios que más espacio ocupan:"
sudo du -sh /* | sort -h
echo
# Limpiar archivos temporales
echo "Limpiando archivos temporales y cachés..."
sudo apt-get clean
sudo apt-get autoremove -y
echo
# Buscar y eliminar archivos grandes
echo "Buscando archivos grandes (>1024MB):"
sudo find / -type f -size +1024M -exec ls -lh {} \;
echo

^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File    ^\ Replace     ^U Paste      ^J Justify    ^/ Go To Line
```

El script inicia con la creacion de una variable que apunta al directorio donde se monta las comprobaciones, para siguiente mostrar en la pantalla del terminal un mensaje que indica que se vera el uso del espacio del disco con el comando “df -h” y se coloca un echo para hacer el espaciado.

Otra función que tiene es la de mostrar los directorios y el espacio que esos usan en este caso en orden del que mas peso tiene al menor y otro echo para hacer un salto de linea.

Otro comando es el de limpiar los archivos como lo son los temporales y los que se encuentran en la memoria caché mediante el uso del comando “sudo apt-get clean” y el comando “sudo apt-get autoremove -y” donde el primero es para eliminar la caché y el otro los archivos temporales.

para terminar la función de buscar y eliminar los archivos de gran tamaño que ocupen el espacio del almacenamiento, esto es posible con el comando “sudo find / -type f -size +1024M -exec ls -lh {} \;”

```
# Verificar el uso de inodos
echo "Uso de inodos:"
df -i
echo
# Eliminar archivos antiguos de logs
echo "Limpiando logs antiguos..."
sudo journalctl --disk-usage
sudo journalctl --vacuum-time=7d
echo
# Comprobar y reparar el sistema de archivos
# Reemplaza /dev/sda1 con la partición correcta
echo "Verificando sistema de archivos en /dev/sda1..."
sudo fsck /dev/sdx1
echo
echo "Mantenimiento del espacio en disco completado."
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

También tiene otra función de verificar el uso de los inodos que son el índice de los archivos que maneja los SO como Linux, también tiene la función de eliminar los archivos viejos de logs esto con los comandos “journalctl --disk-usage” y el comando “sudo journalctl --vacuum-time=7d”

Para terminar el script se realiza la comprobación de reparar el sistema de los archivos y se reemplaza con la partición que sea la correcta del sistema esto se realiza mediante el uso del comando “sudo fsck /dev/sdx1”.

```
Activities Terminal Aug 23 18:51 •
debianjuanj@10:~
```

```
root@10:/home/debianjuanj/trabajos# touch FreddyJimenez.sh
root@10:/home/debianjuanj/trabajos# nano FreddyJimenez.sh
root@10:/home/debianjuanj/trabajos# sh FreddyJimenez.sh
Uso del espacio en disco:
Filesystem      Size   Used  Avail Use% Mounted on
udev            2.5G    0     2.5G  0% /dev
tmpfs           508M   2.7M  505M  1% /run
/dev/sda1        19G   5.9G   12G  34% /
tmpfs           2.5G    0     2.5G  0% /dev/shm
tmpfs           5.0M   8.0K   5.0M  1% /run/lock
tmpfs           508M  128K  508M  1% /run/user/1000

Directarios que más espacio ocupan:
du: cannot access '/proc/13475/task/13475/fd/4': No such file or directory
du: cannot access '/proc/13475/task/13475/fdinfo/4': No such file or directory
du: cannot access '/proc/13475/fd/3': No such file or directory
du: cannot access '/proc/13475/fdinfo/3': No such file or directory
du: cannot access '/run/user/1000/doc': Permission denied
du: cannot access '/run/user/1000/gvfs': Permission denied
0      /bin
0      /dev
0      /initrd.img
0      /initrd.img.old
0      /lib
0      /lib64
```

```
Activities Terminal Aug 23 18:52 •
debianjuanj@10:~
```

```
e6d483b47cc-77d0cdd3515d6822.journal~ (8.0M).
Deleted archived journal /var/log/journal/e994332c239344a7bca69af4410d101e/user-1000@00
061e6d4dc4bdbf-1805c999725b3d69.journal~ (8.0M).
Deleted archived journal /var/log/journal/e994332c239344a7bca69af4410d101e/system@00061
fae63f47600-d2d2be2d5c6adc07.journal~ (8.0M).
Deleted archived journal /var/log/journal/e994332c239344a7bca69af4410d101e/user-1000@00
061fae66ddfebd-94f40718de384cc2.journal~ (8.0M).
Deleted archived journal /var/log/journal/e994332c239344a7bca69af4410d101e/system@00062
0100c57a2cb-0a97fe0f77992df8.journal~ (8.0M).
Deleted archived journal /var/log/journal/e994332c239344a7bca69af4410d101e/user-1000@00
06201010e2c828-7037795f87172e52.journal~ (8.0M).
Vacuuming done, freed 48.0M of archived journals from /var/log/journal/e994332c239344a7
bca69af4410d101e.
Vacuuming done, freed 0B of archived journals from /run/log/journal.
Vacuuming done, freed 0B of archived journals from /var/log/journal.

Verificando sistema de archivos en /dev/sda1...
fsck from util-linux 2.38.1
e2fsck 1.47.0 (5-Feb-2023)
fsck.ext2: No such file or directory while trying to open /dev/sdX1
Possibly non-existent device?

Mantenimiento del espacio en disco completado.
root@10:/home/debianjuanj/trabajos# █
```

Al momento de ejecutar el script este realizo todas las funciones de forma correcta y se tardo un buen rato en realizar todo por lo que yo creo que era más por buscar todos los archivos que pesan y el mostrar toda esa información aunque no funcionó la parte del cambio del directorio sda posiblemente porque en el equipo que se realizó no es la misma dirección o no existe.

Edith Chuico: Script para ingreso de usuarios con cédula en un archivo.txt

The screenshot shows a terminal window titled "EdithChuico.sh *". The script content is as follows:

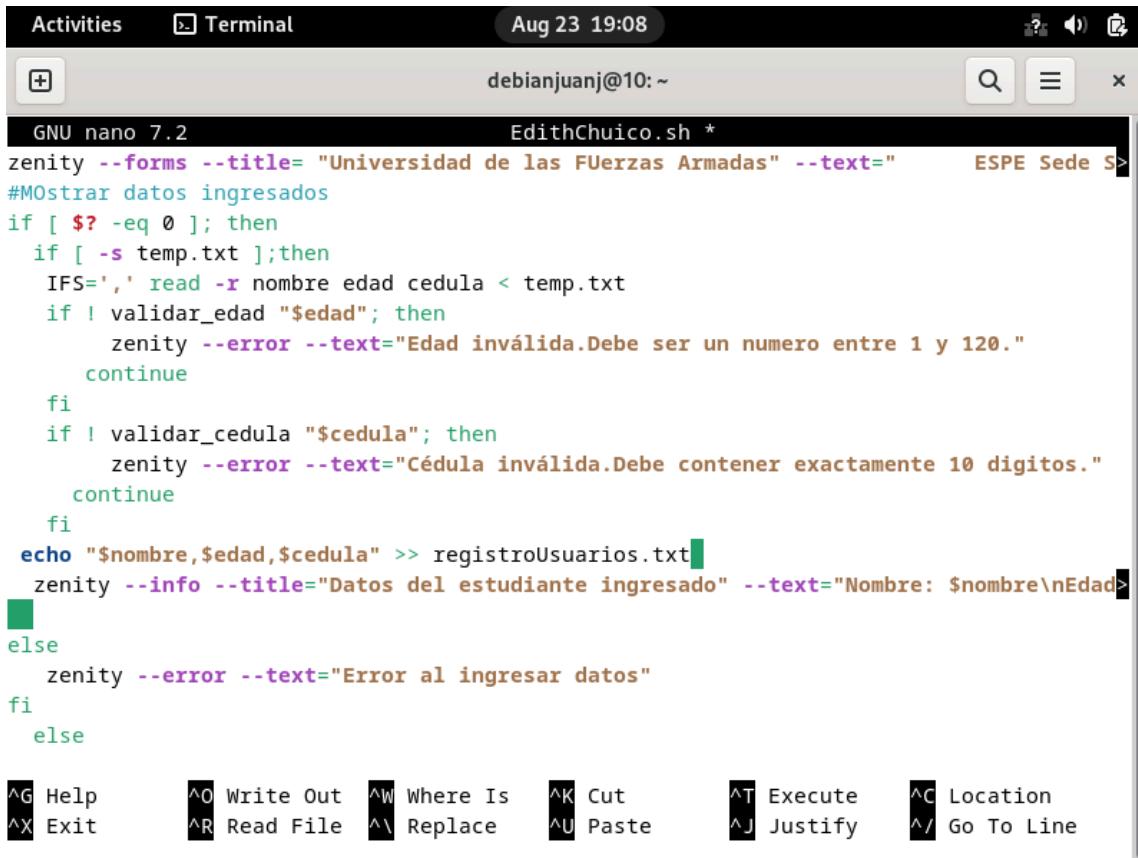
```
GNU nano 7.2                               EdithChuico.sh *
#!/bin/bash
validar_edad(){
    if [ "$1" -ge 1 ] 2>/dev/null && [ "$1" -le 120 ] 2>/dev/null; then
        return 0
    else
        return 1
    fi
}
validar_cedula(){
    if [ ${#1} -ne 10 ]; then
        return 1
    fi
    if ! [[ "$1" =~ ^[0-9]{10}+$ ]]; then
        return 1
    fi
    return 0
}
zenity --forms --title= "Universidad de las FUerzas Armadas" --text= ESPE Sede S>
#Mostrar datos ingresados
```

At the bottom of the terminal window, there is a keymap legend:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^V Replace	^U Paste	^J Justify	^/ Go To Line

El trabajo individual de mi compañera consiste en el desarrollo de un script que permita el ingreso de usuarios dentro de un archivo.txt mediante una ventana generada por terminal.

El script podemos ver que tiene unas funciones que sirven para la validación de los datos ingresados en caso de las edad esta no debe de ser menor a 1 o mayor a 120 y este regresa un valor de tipo boolean, mientras la otra función es la de validar la cédula donde válida que ingrese valores del 0 al 9 y que no supere los 10 dígitos, igualmente devuelve valores boolean.

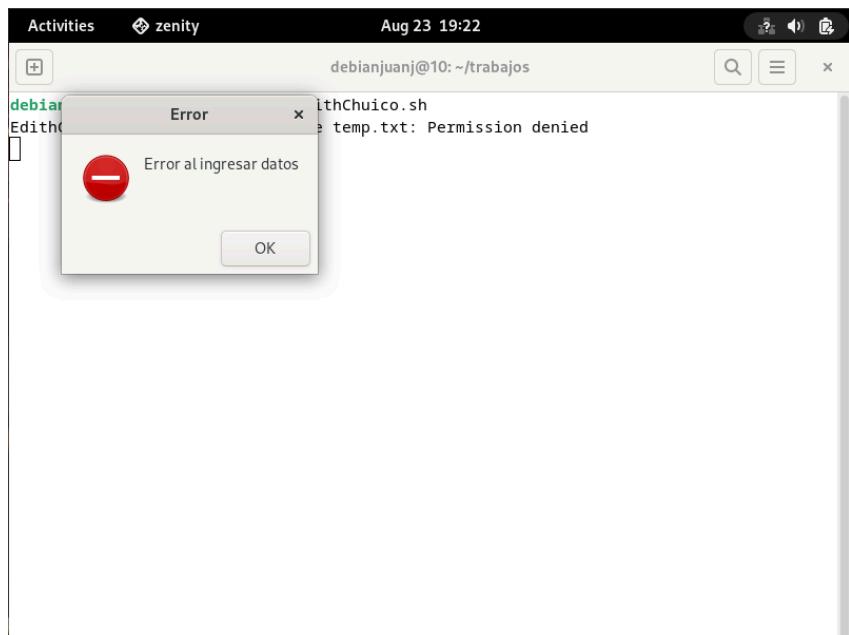


```
GNU nano 7.2          EdithChuico.sh *
zenity --forms --title= "Universidad de las Fuerzas Armadas" --text="      ESPE Sede S>
#Mostrar datos ingresados
if [ $? -eq 0 ]; then
    if [ -s temp.txt ];then
        IFS=',' read -r nombre edad cedula < temp.txt
        if ! validar_edad "$edad"; then
            zenity --error --text="Edad inválida.Debe ser un numero entre 1 y 120."
            continue
        fi
        if ! validar_cedula "$cedula"; then
            zenity --error --text="Cédula inválida.Debe contener exactamente 10 digitos."
            continue
        fi
        echo "$nombre,$edad,$cedula" >> registroUsuarios.txt
        zenity --info --title="Datos del estudiante ingresado" --text="Nombre: $nombre\nEdad:>
else
    zenity --error --text="Error al ingresar datos"
fi
else
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

En la imagen se puede ver que usar un comando “zenity el cual se encarga de crear un archivo, ahora se muestra los datos ingresados con validaciones donde se lee el nombre,la edad y la cédula y verificará si la edad y la cédula son validas, donde en caso de no ingresar datos permitidos este mostrará un mensaje de error.

una vez terminado se guardará dentro de un archivo denominado “registroUsuarios.txt” que dentro tendrá los datos de los estudiantes que se vayan ingresando.



Al probar el script me salto un mensaje diciendo que no se pudo ingresar los datos esto se debe a que la el archivo temp.txt no se pudo realizar.

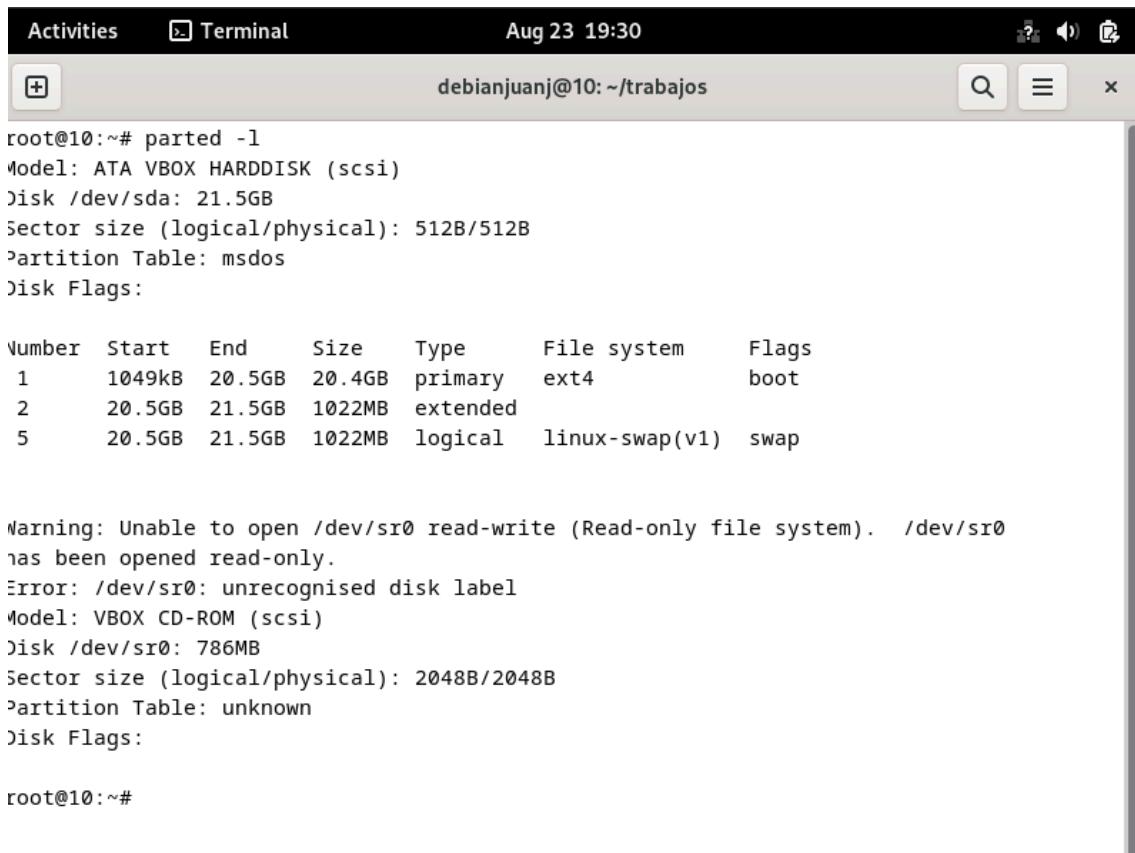


Cambiando los permisos de los archivos me permitió realizar la ejecución del programa en este caso igual que en la presentación me salio error al ingresar la cédula diciendo que tiene más de 10 dígitos.



Dentro del archivo temp.txt se puede ver el estudiante ingresado y sus datos.

Jilmar Gonzales: Particiones de disco y mantenimiento:



The screenshot shows a terminal window titled "Terminal" with the command "root@10:~# parted -l" running. The output lists disk partitions and their details, including a swap partition on /dev/sda5.

```
root@10:~# parted -l
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system    Flags
 1       1049kB  20.5GB  20.4GB  primary   ext4          boot
 2       20.5GB   21.5GB  1022MB  extended
 5       20.5GB   21.5GB  1022MB  logical   linux-swap(v1)  swap

Warning: Unable to open /dev/sr0 read-write (Read-only file system).  /dev/sr0
has been opened read-only.
Error: /dev/sr0: unrecognised disk label
Model: VBOX CD-ROM (scsi)
Disk /dev/sr0: 786MB
Sector size (logical/physical): 2048B/2048B
Partition Table: unknown
Disk Flags:

root@10:~#
```

El compañero con sus comandos nos indica como realizar la particiones de un disco y como realizarle un mantenimiento.

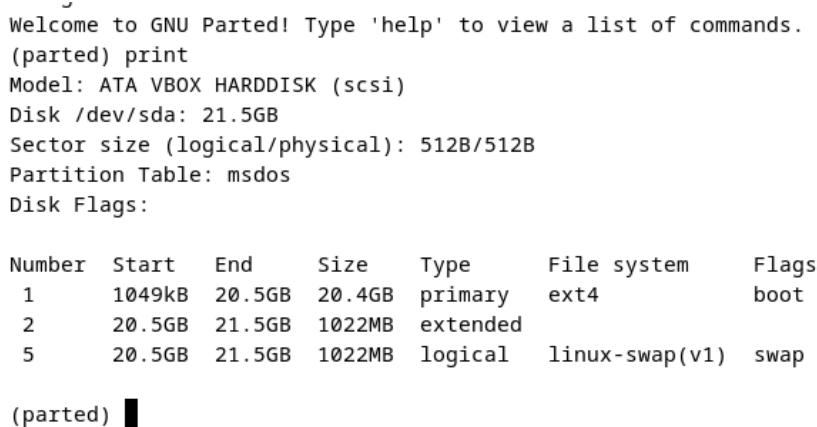
En la imagen se logra apreciar que con el comando “parted -l” que muestra las particiones del disco con algunos datos importantes como lo son el tamaño y el tipo del mismo, entre otros.



A screenshot of a Linux terminal window titled "Terminal". The window shows the command line interface for the "parted" tool, which is used for managing disk partitions. The terminal output is as follows:

```
root@10:~# parted /dev/sda
GNU Parted 3.5
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) █
```

A continuación se realiza el uso del comando “parted (particion seleccionada)” este comando sirve para ingresar al GNU de parted que nos permite realizar varias funciones dentro de las particiones del disco.



A screenshot of a Linux terminal window showing the output of the "parted print" command. The terminal output is as follows:

```
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system    Flags
 1      1049kB  20.5GB  20.4GB  primary   ext4          boot
 2      20.5GB   21.5GB  1022MB  extended
 5      20.5GB   21.5GB  1022MB  logical   linux-swap(v1)  swap

(parted) █
```

Con el comando print se muestra la información del disco en el que ingresamos y es similar a lo mostrado en el primer comando.

Activities Terminal Aug 23 19:38

```
debianjuanj@10: ~/trabajos
```

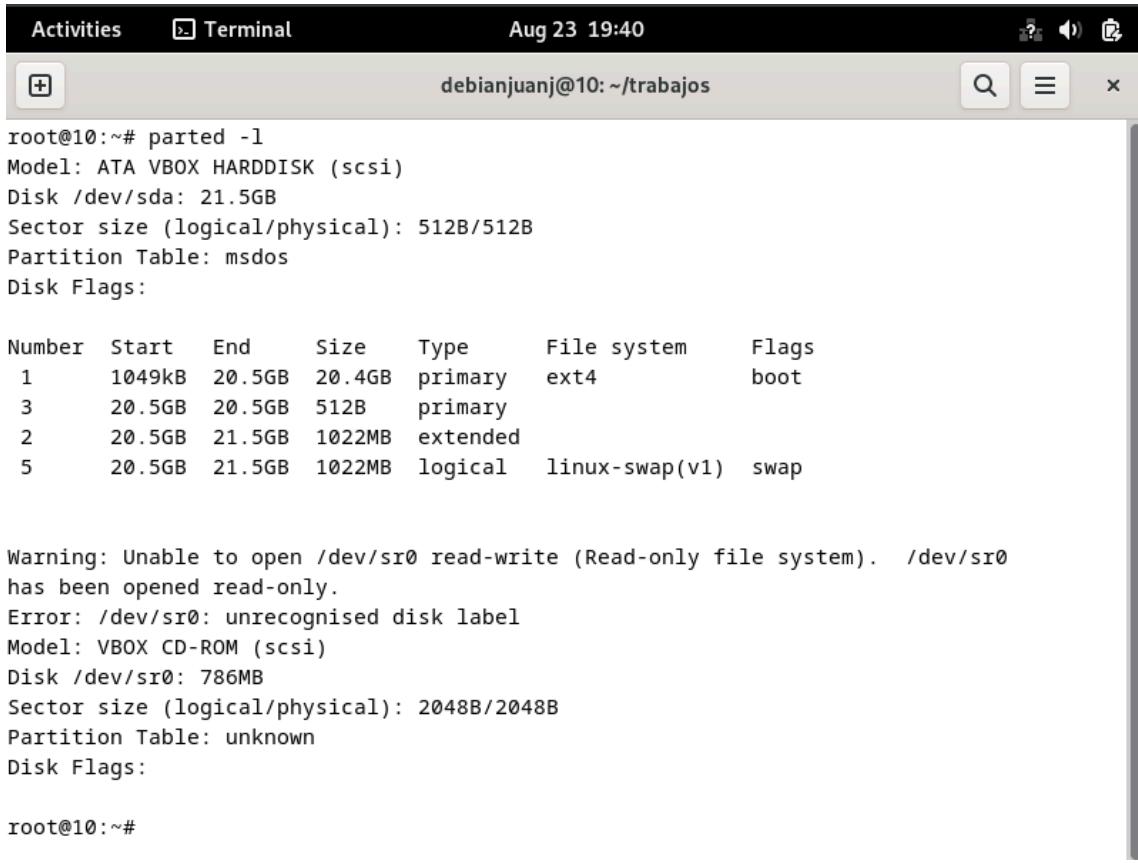
unit UNIT set the default unit to UNIT
version display the version number and copyright
information of GNU Parted
(parted) mkpart primary ext4 31.0GB 35.0GB
Error: The location 31.0GB is outside of the device /dev/sda.
(parted) mkpart primary ext4 10.0GB 15.0GB
Warning: You requested a partition from 10.0GB to 15.0GB (sectors 19531250..29296875).
The closest location we can manage is 20.5GB to 20.5GB (sectors 39942144..39942144).
Is this still acceptable to you?
Yes/No? y
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags
1 1049kB 20.5GB 20.4GB primary ext4 boot
3 20.5GB 20.5GB 512B primary ext4 lba
2 20.5GB 21.5GB 1022MB extended
5 20.5GB 21.5GB 1022MB logical linux-swap(v1) swap

(parted) █

CTRL DERECHA

Con el comando “mkpart primary ext4 31.0GB 35.0GB “ se realizo una particion al disco el cual se seleccionó como primario y el cual se muestra actualmente en el disco debido a que anteriormente solo se encontraba 3 particiones y ahora existen 4.



The screenshot shows a terminal window titled "Terminal" with the command "root@10:~# parted -l" entered. The output lists the disk geometry and partition table for the primary disk /dev/sda. It shows five partitions: one primary partition (1) of 20.4GB with ext4 file system and boot flag; one primary partition (3) of 512B; one extended partition (2) of 1022MB; and one logical partition (5) of 1022MB with linux-swap(v1) file system and swap flag. A warning message indicates that the CD-ROM /dev/sr0 is read-only. The command "root@10:~#" is shown at the bottom.

```
root@10:~# parted -l
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Partition Flags:

Number  Start   End     Size    Type      File system     Flags
 1      1049kB  20.5GB  20.4GB  primary   ext4           boot
 3      20.5GB   20.5GB  512B    primary
 2      20.5GB   21.5GB  1022MB  extended
 5      20.5GB   21.5GB  1022MB  logical   linux-swap(v1)  swap

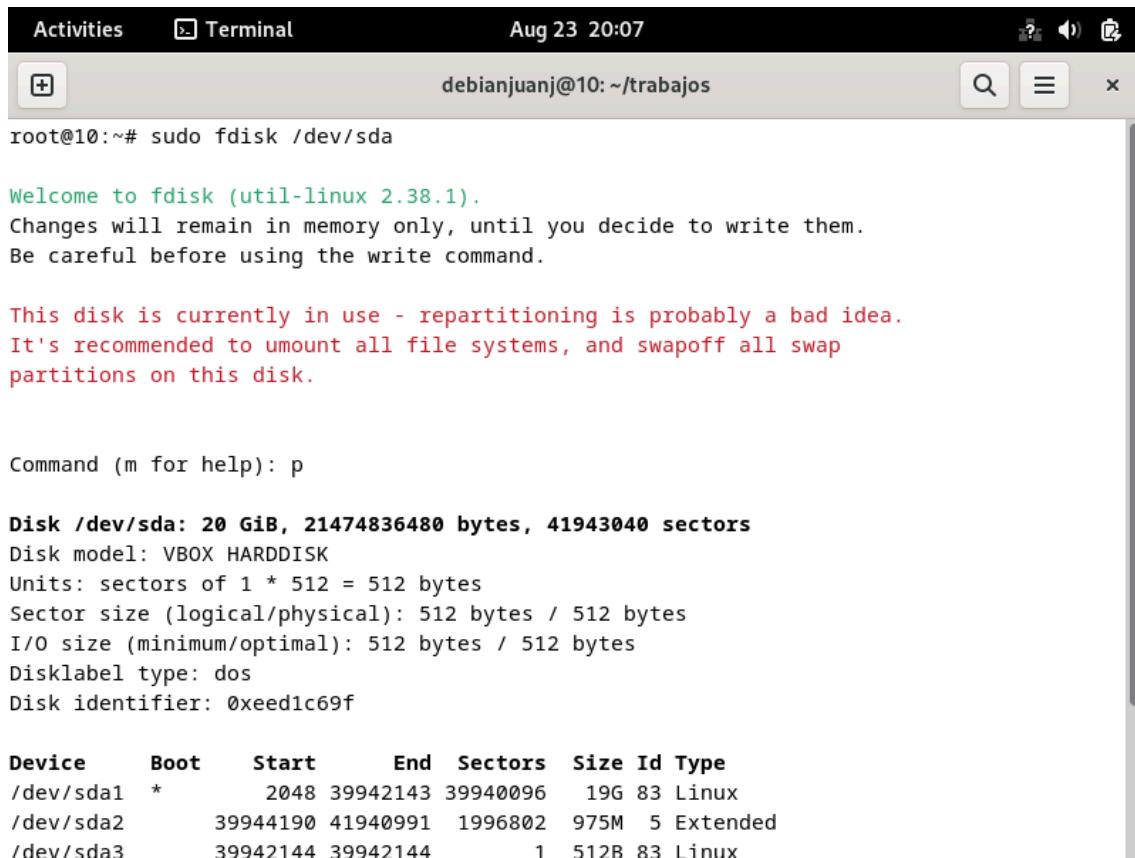
Warning: Unable to open /dev/sr0 read-write (Read-only file system).  /dev/sr0
has been opened read-only.
Error: /dev/sr0: unrecognised disk label
Model: VBOX CD-ROM (scsi)
Disk /dev/sr0: 786MB
Sector size (logical/physical): 2048B/2048B
Partition Table: unknown
Partition Flags:

root@10:~#
```

Se vuelve a utilizar el comando “parted -l” para verificar que se realizó correctamente y se logra ver que si se creó otra partición del disco aunque esto trae la desventaja de que quita espacio a los demás discos y puede que el equipo se ralentice debido a esto.

Gonzalo Mera: Gestión de archivos y almacenamiento, Gestión de discos:

El compañero realizó una guia para la creación de particiones y edición de particiones, además de tener otra sección dada para la gestión de los directorios.



The screenshot shows a terminal window titled "Terminal" with the status bar indicating "Aug 23 20:07". The user is root at "debianjuanj@10: ~/trabajos". The terminal displays the following output from the "fdisk /dev/sda" command:

```
root@10:~# sudo fdisk /dev/sda

Welcome to fdisk (util-linux 2.38.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

This disk is currently in use - repartitioning is probably a bad idea.
It's recommended to umount all file systems, and swapoff all swap
partitions on this disk.

Command (m for help): p

Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xeeed1c69f

Device      Boot   Start     End   Sectors  Size Id Type
/dev/sda1    *      2048 39942143 39940096   19G 83 Linux
/dev/sda2          39944190 41940991 1996802  975M  5 Extended
/dev/sda3      39942144 39942144       1  512B 83 Linux
```

Con el comando “sudo fdisk /dev/sda” ingresamos a un administrador de los Discos donde con el comando “p” nos permite ver la información de estos y su id ,el tipo, entre otros.

```
root@10:~# sudo fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
DiskLabel type: dos
Disk identifier: 0xeeed1c69f

Device      Boot   Start     End   Sectors  Size Id Type
/dev/sda1        *    2048 39942143 39940096   19G 83 Linux
/dev/sda2            39944190 41940991 1996802   975M  5 Extended
/dev/sda3            39942144 39942144       1   512B 83 Linux
/dev/sda5            39944192 41940991 1996800   975M 82 Linux swap / Solaris

Partition table entries are not in disk order.
root@10:~#
```

Con el comando “sudo fdisk -l” muestra así mismo las particiones pero en una forma un poco más resumida y con alguna información menos.

```
Command (m for help): d
Partition number (1-3,5, default 5): 2

Partition 2 has been deleted.

Command (m for help): n
Partition type
  p  primary (2 primary, 0 extended, 2 free)
  e  extended (container for logical partitions)
Select (default p): e
Partition number (2,4, default 2): 2
First sector (39942145-41943039, default 39944192):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (39944192-41943039, default 41943039):

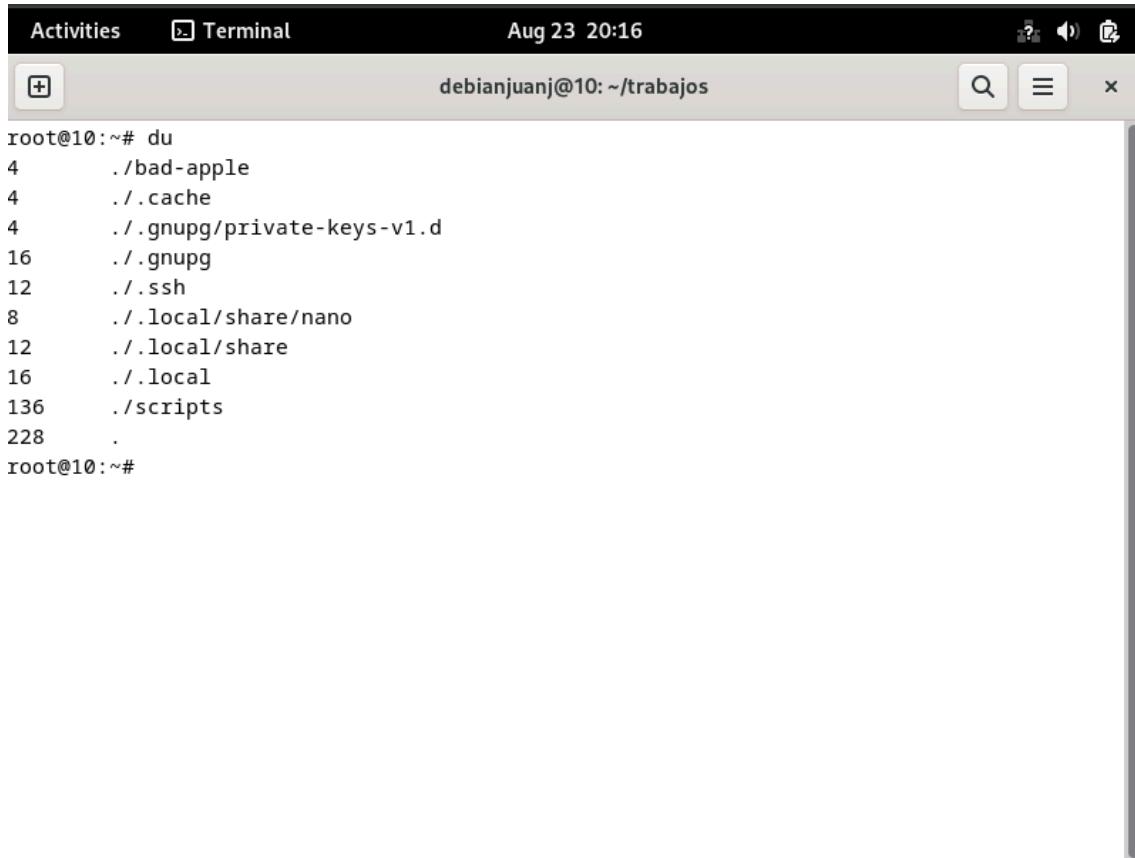
Created a new partition 2 of type 'Extended' and of size 976 MiB.
Partition #2 contains a swap signature.

Do you want to remove the signature? [Y]es/[N]o: y
The signature will be removed by a write command.

Command (m for help):
```

Con el comando “d” dentro de la sección de comandos fdisk realizara la función de eliminar una partición en este caso se seleccionó la opción 2 y muestra una imagen de que se eliminó.

En cambio con el comando n se creó otra partición donde se eligió que sea una extendida con la letra “e” y se le seleccionó los valores por defecto.



A screenshot of a terminal window titled "Terminal". The window shows the command "du" being run by root at the prompt "root@10:~#". The output lists various directory paths and their sizes in kilobytes (4, 16, 12, 8, 12, 16, 136, 228). The terminal interface includes a header bar with "Activities", a search bar, and standard window controls.

```
root@10:~# du
4      ./bad-apple
4      ./cache
4      ./gnupg/private-keys-v1.d
16     ./gnupg
12     ./ssh
8      ./local/share/nano
12     ./local/share
16     ./local
136    ./scripts
228   .
root@10:~#
```

Con el comando “du” se ve el espacio de cada directorio de la ubicación que estamos donde podemos gestionar y analizar cuáles carpetas son las que pesan demasiado.

4. Conclusiones

- La ejecución de comandos específicos para la administración de archivos y directorios, junto con la creación de scripts, facilita el manejo eficiente del espacio y la organización de los datos.
- La automatización no solo ahorra tiempo, sino que también reduce el riesgo de errores humanos, lo que mejora la eficiencia y fiabilidad en la administración del sistema.
- El informe elaborado sobre los comandos proporcionados ha resaltado la importancia de comprender cada comando y su función específica. Analizar y documentar los comandos ayuda a consolidar el conocimiento sobre sus aplicaciones y beneficios.

5. Recomendaciones

- Es aconsejable probar los scripts creados en entornos de prueba o de desarrollo antes de aplicarlos en sistemas en producción.
- Se recomienda explorar comandos más avanzados y sus combinaciones para mejorar aún más la gestión del almacenamiento y la automatización. Esto incluye investigar sobre opciones adicionales, banderas y técnicas de scripting más complejas para tareas específicas.
- Mantener la documentación actualizada sobre los comandos y scripts utilizados es crucial para asegurar que toda la información esté al día y sea relevante.

6. Bibliografía/ Referencias

- "Sistemas Operativos Modernos" de Andrew S. Tanenbaum.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (Año desconocido). Conceptos de Sistemas Operativos.
- Tanenbaum, A. S. (Año desconocido). Sistemas Operativos Modernos.
- Love, R. (Año desconocido). Desarrollo del Kernel de Linux.
- Russinovich, M., Solomon, D. A., & Ionescu, A. (Año desconocido). Internals de Windows

7. Anexos: