

## Sistemas Operativos

### Practica 1:

Introducción al Shell: comandos básicos y scripts

### Integrantes:

No. Lista	Nombre
23	Angelo Mihaelle Ojeda Gómez
18	Hernández Hernández Julio Cesar
20	Martínez Robledo Luis Antonio
15	García Silva Jonathan
11	Estrada Corona Alexis
9	Espinoza García Hilario Sebastián

### Grupo:

4CV3

## Tabla de contenido

Objetivo .....	2
Descripción .....	2
Comandos Básicos .....	3
• date .....	3
• clear .....	3
• exit .....	3
• help .....	3
• history .....	3
• cd .....	3
• dir .....	3
• cat .....	3
• pwd .....	4
• mkdir .....	4
• locate .....	4
• find .....	4
• kill .....	4
• dif .....	4
• df .....	4
Prueba de pantalla .....	5
date .....	5
Clear .....	5
Exit .....	6
help .....	6
history .....	7
cd .....	7
dir .....	7
cat .....	8
pwd .....	8
.....	8
mkdir .....	9
locate .....	9
find .....	9
kill .....	10
diff .....	10
df .....	10
Conclusiones .....	11
Referencias .....	12

## Objetivo

Aprender el uso de los comandos básicos de intérprete de comandos de LINUX/UNIX, así como la ejecución de scripts.

## Descripción

Shell se conoce comúnmente como el shell (utilizado para distinguirlo del núcleo) y se refiere al software (analizador de comandos) que "proporciona una interfaz operativa para los usuarios". Es similar a command.com bajo DOS y posterior cmd.exe. Recibe comandos del usuario y luego llama a la aplicación correspondiente. Shell es un programa escrito en C que es un puente para que los usuarios usen Linux. Shell es tanto un lenguaje de comando como un lenguaje de programación. Shell se refiere a un programa de aplicación que proporciona una interfaz a través de la cual los usuarios acceden a los servicios del núcleo del sistema operativo.

El shell de Ken Thompson es el primer shell de Unix, y el Explorador de Windows es un shell gráfico típico. El script de shell es un programa de script escrito para el shell. El shell en la industria generalmente se refiere al script de shell, pero los lectores deben saber que shell y script de shell son dos conceptos diferentes. Shell es el intérprete de órdenes o intérprete de comandos, es el programa informático que provee una interfaz de usuario para acceder a los servicios del sistema operativo.

El shell también es la interfaz primaria entre un usuario sentado ante su terminal y el sistema operativo. Cuando un usuario ingresa en el sistema, se inicia un shell. El shell tiene la terminal como entrada y salida estándar, y lo primero que hace es exhibir la indicación (prompt), un carácter como un signo de dólar, que le indica al usuario que el shell está esperando para aceptar un comando.

## Comandos Básicos

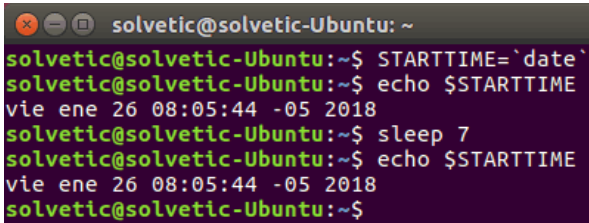
En esta categoría se incluyen todas las órdenes fundamentales que se utilizan para administrar el terminal, con las que se puede limpiar la ventana de la consola, recuperar comandos anteriores desde la historia o finalizar la sesión. Pero también existen otros tipos de comandos básicos que sirven para manipular algunos tipos de archivos, navegar por el sistema de forma sencilla, etc. En la siguiente lista incluimos algunos de los más relevantes.

- **date**
  - Nos permite imprimir la fecha de hoy en muchísimos formatos. También nos permite extraer solo el día, mes, día de la semana, etc. Es útil en la ejecución de scripts pues nos permite saber cuándo se ha ejecutado, por ejemplo, si estos se ejecutan de forma automática en ciertas circunstancias.
- **clear**
  - Es un comando que borra toda la información del terminal y lo deja como si fuera nuevo. Esto borrará toda la pantalla y nos deja un terminal limpio.
- **exit**
  - Cierra ficheros, vacía buffers, devuelve estado de terminación del proceso. Se usa para que el proceso padre conozca la forma en que finaliza un hijo. Puede ser cualquier tipo de entero, pero se suelen usar las macros: EXIT\_FAILURE y EXIT\_SUCCESS. exit (), cierra ficheros, vacía buffers.
- **help**
  - Guarda y muestra información sobre los comandos de bash. Si desconocemos y tenemos dudas sobre cómo usar un comando, tipeando en la terminal – help nos va a mostrar una lista incluyendo las opciones de un determinado comando.
- **history**
  - Es un comando muy útil para averiguar los últimos comandos que se han ejecutado en un Servidor. El comando mostrará, por defecto, los últimos 500 comandos ejecutados, tomando estos datos del archivo.
- **cd**
  - Permite moverse entre directorios del sistema. Puedes cambiar de directorio especificando la ruta absoluta desde el directorio raíz o relativa desde tu ubicación actual, en Linux el directorio actual se indica con el signo.
- **dir**
  - Aparte de permitir realizar búsquedas de archivos y directorios, permite gracias a sus opciones crear índices, ordenar, descubrir rutas largas que den error en la red, encontrar ficheros ocultos o incluso detectar fichero con datos ocultos de ADS
- **cat**
  - El comando cat en Linux es uno de los más útiles que puedes aprender. Deriva su nombre de la palabra concatenar y te permite crear, fusionar o imprimir archivos en la pantalla de salida estándar o en otro archivo y mucho más.

- **pwd**
  - El comando pwd es un comando del tipo Unix, donde sus siglas provienen del en inglés print working directory, cuya traducción sería imprimir directorio de trabajo, se utiliza para imprimir el nombre del directorio actual en una sesión de comandos.
- **mkdir**
  - Usada para crear un nuevo subdirectorio o carpeta del sistema de archivos
- **locate**
  - Desarrollada para realizar búsquedas, de forma fácil y muy rápida, de elementos (archivos y directorios) basado en patrones de nombres.
- **find**
  - El comando más común utilizado para encontrar y filtrar archivos en Linux es a través del comando find.
- **kill**
  - kill es un comando utilizado para enviar mensajes sencillos a los procesos ejecutándose en el sistema. Por defecto el mensaje que se envía es la señal de terminación (SIGTERM), que solicita al proceso limpiar su estado y salir.
- **diff**
  - Es la utilidad para comparar directorios entre sí o ficheros entre sí. -r: Para indicar que 2 directorios se comparen de forma recursiva. En otras palabras, con la opción -r también se compararán todos los subdirectorios que están dentro del directorio analizado
- **df**
  - Usa el comando df para obtener un informe sobre el uso del espacio en disco del sistema, que se muestra en porcentaje y KB. Si deseas ver el informe en megabytes, escribe df -m.

## Prueba de pantalla

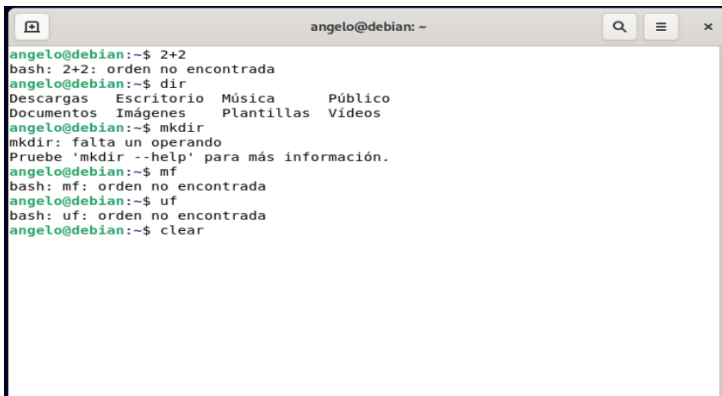
date



```
solvetic@solvetic-Ubuntu: ~  
solvetic@solvetic-Ubuntu:~$ STARTTIME=`date`  
solvetic@solvetic-Ubuntu:~$ echo $STARTTIME  
vie ene 26 08:05:44 -05 2018  
solvetic@solvetic-Ubuntu:~$ sleep 7  
solvetic@solvetic-Ubuntu:~$ echo $STARTTIME  
vie ene 26 08:05:44 -05 2018  
solvetic@solvetic-Ubuntu:~$
```

Nos muestra la fecha en la pantalla.

Clear



```
angelo@debian: ~  
angelo@debian:~$ 2+2  
bash: 2+2: orden no encontrada  
angelo@debian:~$ dir  
Descargas Escritorio Música Público  
Documentos Imágenes Plantillas Videos  
angelo@debian:~$ mkdir  
mkdir: falta un operando  
Pruebe 'mkdir --help' para más información.  
angelo@debian:~$ mf  
bash: mf: orden no encontrada  
angelo@debian:~$ uf  
bash: uf: orden no encontrada  
angelo@debian:~$ clear
```

En este paso se introduce el comando.



```
angelo@debian: ~  
angelo@debian:~$ █
```

Aquí se muestra la consola limpia luego de ejecutarlo.

## Exit

```
angelo@debian: ~/Documentos
angelo@debian:~/Documentos$ cd ~
angelo@debian:~$ cd Documentos
angelo@debian:~/Documentos$ exit
```

Este comando mata al proceso actual, por lo que al ejecutarlo cierra la terminal y nos deja en el escritorio de nuestro SO



## help

```
angelo@debian: ~
angelo@debian:~$ help
GNU bash, versión 5.1.4(1)-release (x86_64-pc-linux-gnu)
Estas órdenes de shell están definidas internamente.  Teclee 'help' para
ver esta lista.
Teclee 'help nombre' para saber más sobre la función 'nombre'.
Utilice 'info bash' para saber más sobre la shell en general.
Utilice 'man -k' o 'info' para saber más sobre las órdenes que no están en
esta lista.

Un asterisco (*) junto a un nombre significa que la orden está desactivada.

id_trabajo [&]
(( expresión ))
. fichero [argumentos]
:
[ arg... ]
[[ expresión ]]
alias [-p] [nombre[=valor] ... ]
bg [id_trabajo ...]
bind [-lpsvPSVX] [-m comb_teclas] [-f>
break [n]
builtin [orden-interna-shell [arg ...>
caller [expresión]
case PALABRA in (PATRÓN) [[ PATRÓN]...>
cd [-L|-P [-e]] [dir]
command [-pVv] orden [arg ...]
compgen [-abcdfgjkusv] [-o option] [>
complete [-abcdfgjkusv] [-pr] [-DEI]>
compopt [-o]+o opción [-DEI] [nombre>
continue [n]
coproc [NOMBRE] orden [redirecciones>
declare [-aAffgIlnrtux] [-p] [name[=>
dirs [-clpv] [+N] [-N]
disown [-h] [-ar] [idtrabajo ... | pi>
echo [-neE] [arg ...]
enable [-a] [-dnps] [-f fichero] [nom>
eval [arg ...]
exec [-cl] [-a name] [command [argum>
exit [n]
export [-fn] [nombre[=valor] ...] ó e>
false
fc [-e nombre_e] [-lnr] [primero] [úl>
fg [id_trabajo]
for NOMBRE [in PALABRAS ... ] ; do ÓR>
for (( exp1; exp2; exp3 )); do ÓRDENE>
function nombre ( ÓRDENES ; ) o nomb>
getopts optstring name [arg ...]
hash [-lr] [-p ruta] [-dt] [nombre ...]
help [-dms] [patrón ...]
history [-c] [-d despl] [n] o histor>
if ÓRDENES; then ÓRDENES; [ elif ÓRD>
jobs [-lnprs] [idtrabajo ...] o jobs>
kill [-s id señal | -n num_señal | ->
let arg [arg ...]
local [opción] nombre[=valor] ...
logout [n]
mapfile [-d delim] [-n cuenta] [-O o>
popd [-n] [+N | -N]
printf [-v var] formato [argumentos]
pushd [-n] [+N | -N | dir]
pwd [-LP]
read [-ers] [-a matriz] [-d delim] [>
readarray [-d delim] [-n cuenta] [-O>
readonly [-aAf] [nombre[=valor] ...]>
return [n]
select NOMBRE [in PALABRAS ... ;] do>
set [-abefhkmnptuvxBCHP] [-o nombre->
shift [n]
shopt [-pqsu] [-o] [nombre_opción...>
source fichero [argumentos]
suspend [-f]
test [expresión]
time [-p] tubería
times
trap [-lp] [[arg] id_señal ...]
true
type [-afptP] nombre [nombre ...]
typeset [-aAffgIlnrtux] [-p] name[=>
ulimit [-SHabcdefiklmnpqrstuvxPT] [l>
umask [-p] [-S] [modo]
unalias [-a] nombre [nombre ...]
unset [-f] [-v] [-n] [nombre ...]
until ÓRDENES; do ÓRDENES; done
variables - Nombres y significados d>
wait [-fn] [-p var] [id ...]
while ÓRDENES; do ÓRDENES; done
{ ÓRDENES ; }
```

En este caso el comando help nos brinda de información general de los comandos propios de Linux en el momento

## history



```
angelo@debian: ~  
angelo@debian:~$ history  
1 2+2  
2 dir  
3 mkdir  
4 mf  
5 uf  
6 clear  
7 history  
angelo@debian:~$
```

Muestra el historial de comandos que se han ejecutado incluyéndose a sí mismo.

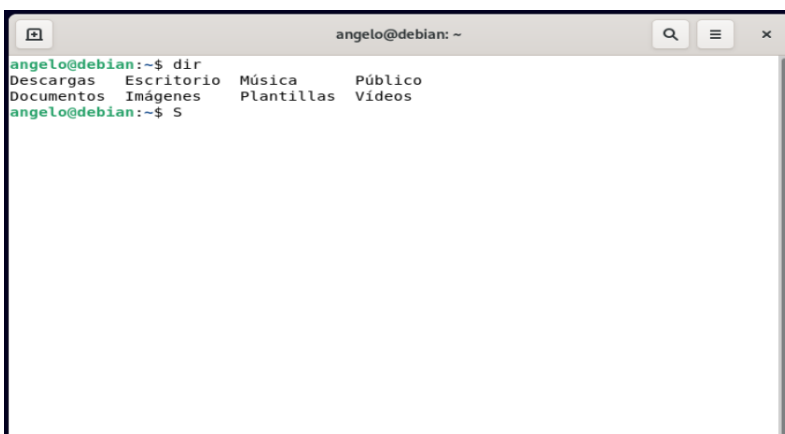
## cd



```
angelo@debian: ~/Documentos  
angelo@debian:~/Documentos$ cd ~  
angelo@debian:~$ cd Documentos  
angelo@debian:~/Documentos$
```

Cambia la dirección o ruta en la que se encuentra.

## dir



```
angelo@debian: ~  
angelo@debian:~$ dir  
Descargas  Escritorio  Música      Público  
Documentos Imágenes    Plantillas  Vídeos  
angelo@debian:~$ S
```

Muestra todas las carpetas y archivos existentes en la ruta/ directorio en el que te encuentras



## cat

```
angelo@debian: ~/Documentos/ArcNuv
angelo@debian:~/Documentos/ArcNuv$ pwd
/home/angelo/Documentos/ArcNuv
angelo@debian:~/Documentos/ArcNuv$ cat > file.txt
Hola
exit
dkkdk
^C
angelo@debian:~/Documentos/ArcNuv$ dir
file.txt
angelo@debian:~/Documentos/ArcNuv$ file.txt
bash: file.txt: orden no encontrada
angelo@debian:~/Documentos/ArcNuv$ ls -l
total 4
-rw-r--r-- 1 angelo angelo 16 sep 16 16:35 file.txt
angelo@debian:~/Documentos/ArcNuv$ cat file.txt
Hola
exit
dkkdk
angelo@debian:~/Documentos/ArcNuv$ S
```

El comando cat tiene muchas funciones, entre ellas crear un archivo de forma rápida como se ve en esta primera imagen.

```
file.txt
angelo@debian:~/Documentos/ArcNuv$ file.txt
bash: file.txt: orden no encontrada
angelo@debian:~/Documentos/ArcNuv$ ls -l
total 4
-rw-r--r-- 1 angelo angelo 16 sep 16 16:35 file.txt
angelo@debian:~/Documentos/ArcNuv$ cat file.txt
Hola
exit
dkkdk
angelo@debian:~/Documentos/ArcNuv$ cat file2.txt
cat: file2.txt: No existe el fichero o el directorio
angelo@debian:~/Documentos/ArcNuv$ cat > file2.txt
Adios
entrar
xdxdxdxd
^C
angelo@debian:~/Documentos/ArcNuv$ cat file2.txt
Adios
entrar
xdxdxdxd
angelo@debian:~/Documentos/ArcNuv$ cat file2.txt > file.txt
angelo@debian:~/Documentos/ArcNuv$ cat file.txt
Adios
entrar
xdxdxdxd
angelo@debian:~/Documentos/ArcNuv$ S
```

Aquí creamos un segundo archivo y redirigimos su contenido a el primer archivo creado.

## pwd

```
angelo@debian: ~/Documentos/ArcNuv
angelo@debian:~/Documentos/ArcNuv$ pwd
/home/angelo/Documentos/ArcNuv
angelo@debian:~/Documentos/ArcNuv$
```

Imprime el directorio/ruta en la que nos encontramos.

## mkdir

```
angelo@debian: ~/Documentos/ArcNuv
angelo@debian:~$ mkdir ~/Documentos/ArcNuv
angelo@debian:~$ cd ~/Documentos/
angelo@debian:~/Documentos$ dir
ArcNuv  myFile  nuvArc
angelo@debian:~/Documentos$ cd ArcNuv
angelo@debian:~/Documentos/ArcNuv$
```

Creamos un nuevo directorio llamado “ArcNuv” en la carpeta “Documentos” luego entramos a ella y ejecutamos el comando dir para verificar que se haya creado el directorio nuevo, finalmente nos movemos a él.

## locate

```
sebas@sebas-VirtualBox: ~/Descargas
sebas@sebas-VirtualBox:~/Descargas$ locate "*.mp3"
sebas@sebas-VirtualBox:~/Descargas$ locate "*.mp4"
sebas@sebas-VirtualBox:~/Descargas$ locate file.txt
/usr/share/doc/alsa-base/driver/Procfile.txt.gz
sebas@sebas-VirtualBox:~/Descargas$ locate "*.html"
/snap/core18/2128/usr/share/doc/python3/python-policy.html
/snap/core18/2128/usr/share/doc/python3.6/python-policy.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Attributes.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Classes.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Concepts.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Delegates.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Enumerated_types__Enum$.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Errors.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Expressions.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/GIDL_metadata_format.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/GIR_metadata_format.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Generics.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Interfaces.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Methods.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Namespaces.html
/snap/gnome-3-34-1804/72/usr/share/devhelp/books/vala-0.46/Overview.html
```

“Locate” es utilizada para encontrar archivos por nombre dando su direccion exacta.

## find

```
sebas@sebas-VirtualBox: ~
sebas@sebas-VirtualBox:~$ find -name "*.txt"
./mozilla/firefox/l2zrb4nr.default-release/pkcs11.txt
./mozilla/firefox/l2zrb4nr.default-release/serviceworker.txt
./mozilla/firefox/l2zrb4nr.default-release/AlternateServices.txt
./mozilla/firefox/l2zrb4nr.default-release/SiteSecurityServiceState.txt
./cache/tracker/db-version.txt
./cache/tracker/db-locale.txt
./cache/tracker/first-index.txt
./cache/tracker/last-crawl.txt
./cache/tracker/parser-version.txt
sebas@sebas-VirtualBox:~$ find -type l
./mozilla/firefox/l2zrb4nr.default-release/lock
sebas@sebas-VirtualBox:~$ find -type d
.
./Documentos
./ssh
./Música
./Escritorio
./config
./config/evolution
./config/evolution/sources
./config/ibus
./config/ibus/bus
./config/dconf
./config/gtk-3.0
./config/update-notifier
./config/nautilus
./config/pulse
./config/goa-1.0
./local
```

“Find” sirve para encontrar archivos, directorios o enlaces. Puedes hacer búsquedas por filtros como tamaño, nombre, extension, fecha, etc.

## kill

```
sebas@sebas-VirtualBox: ~  
1133 ?      00:00:00 gsd-sound  
1136 ?      00:00:00 gsd-usb-protect  
1143 ?      00:00:00 colord  
1155 ?      00:00:00 gsd-printer  
1166 ?      00:00:00 gsd-wacom  
1172 ?      00:00:00 kworker/0:5-events  
1173 ?      00:00:00 gsd-wwan  
1174 ?      00:00:00 gsd-xsettings  
1204 ?      00:00:00 ibus-engine-sim  
1299 ?      00:00:00 gnome-terminal-  
1307 pts/0   00:00:00 bash  
1317 ?      00:00:00 update-notifier  
1383 ?      00:00:00 deja-dup-monito  
1393 pts/0   00:00:00 ps  
sebas@sebas-VirtualBox:~$ kill 1393  
bash: kill: (1393) - No existe el proceso  
sebas@sebas-VirtualBox:~$ ps  
PID TTY      TIME CMD  
1307 pts/0    00:00:00 bash  
1394 pts/0    00:00:00 ps  
sebas@sebas-VirtualBox:~$ kill 1394  
bash: kill: (1394) - No existe el proceso  
sebas@sebas-VirtualBox:~$ kill 1307  
sebas@sebas-VirtualBox:~$
```

El comando kill te permite terminar un proceso utilizando un ID de proceso específico, también conocido como pid.

## diff

```
sebas@sebas-VirtualBox:~/Escritorio$ diff file.txt file2.txt  
1,15c1,5  
< Hola Buenas noches  
< as  
<  
<  
<  
<  
< C  
< .  
< C  
< -C  
<  
<  
< .  
< $  
<..  
> ASdasd  
> ad32213  
> asd  
> qe  
> qwe  
sebas@sebas-VirtualBox:~/Escritorio$
```

El comando diff (diferencia), se usa para mostrar las diferencias entre archivos de Linux.

## df

```
angelo@debian: ~  
angelo@debian:~$ df  
S.ficheros bloques de 1K Usados Disponibles Uso% Montado en  
udev 833104 0 833104 0% /dev  
tmpfs 171212 1128 170084 1% /run  
/dev/sda1 50303512 4282764 43433000 9% /  
tmpfs 856048 0 856048 0% /dev/shm  
tmpfs 5120 4 5116 1% /run/lock  
tmpfs 171208 128 171080 1% /run/user/1000  
angelo@debian:~$ df -m  
S.ficheros bloques de 1M Usados Disponibles Uso% Montado en  
udev 814 0 814 0% /dev  
tmpfs 168 2 167 1% /run  
/dev/sda1 49125 4183 42416 9% /  
tmpfs 836 0 836 0% /dev/shm  
tmpfs 5 1 5 1% /run/lock  
tmpfs 168 1 168 1% /run/user/1000  
angelo@debian:~$
```

Nos muestra un informe del disco duro, tanto megas libres y usados en KB y en porcentaje., luego lo ejecutamos con la etiqueta -m y hace lo mismo, pero con la información en MB.

## Conclusiones

Al termino de esta práctica podemos concluir que shell o intérprete de órdenes o intérprete de comandos es el programa informático que provee una interfaz de usuario para acceder a los servicios del sistema operativo. El cual es fundamental para poder conocer a fondo se nuestro sistema operativo. Y que estos pueden ser gráficos o de texto simple, dependiendo del tipo de interfaz que empleen. Y que los shells están diseñados para facilitar la forma en que se invocan o ejecutan los distintos programas disponibles en el computador. Los comandos básicos de Linux ayudan a los usuarios a ejecutar tareas de manera fácil y efectiva. Puede llevar un tiempo recordar algunos de los comandos básicos, pero nada es imposible con mucha práctica.

## Referencias

- 35 Comandos Básicos de Linux que Todo Usuario Debe Saber. (2021, July 29). Retrieved September 16, 2021, from <https://www.hostinger.mx/tutoriales/linux-comandos>
- Objetivo de la Asignatura. (n.d.). Retrieved September 16, 2021, from <https://www.fing.edu.uy/inco/cursos/sistoper/>
- Sysenter Based System Call Mechanism in Linux 2.6. (2006, July 10). Retrieved September 19, 2021, from <https://manugarg.wordpress.com/2006/07/10/sysenter-based-system-call-mechanism-in-linux-26/>
- System calls: ¿qué son y para qué se emplean? (n.d.). Retrieved September 19, 2021, from <https://www.ionos.mx/digitalguide/servidores/know-how/que-son-las-system-calls-de-linux/>
- Wielsch, M., Prahm, J., Esser, H., Liger, F., Wolf, P. M., & Springinsfeld, S. (2000). *Linux*. Paris: Micro Application.
- (n.d.). Retrieved September 19, 2021, from <https://tldp.org/LDP/khg/HyperNews/get/syscall/syscall86.html>