

Laboratorio: Introducción a la Terminal y Línea de Comandos

Nombre: Fernando González Cedula: 2-755-8 Fecha: 4/9/24

Instrucciones Generales:

Ejecución de Comandos:

Para cada paso, ejecuta los comandos indicados en tu terminal.

Después de ejecutar el comando, toma una captura de pantalla que muestre el resultado en tu terminal.

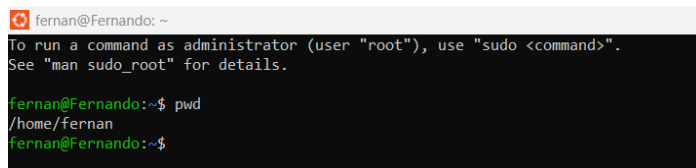
Documentación:

Debajo de cada captura de pantalla, escribe una breve leyenda explicando el propósito del comando.

La leyenda debe ser clara y precisa, indicando qué hace el comando y por qué es útil.

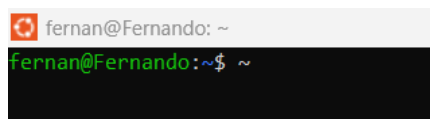
Comandos Básicos de Navegación y Gestión de Archivos

1. pwd

A terminal window with a light gray title bar showing 'fernan@Fernando: ~'. The terminal text shows a message about running commands as administrator, followed by the user typing 'pwd' and the output '/home/fernan'.

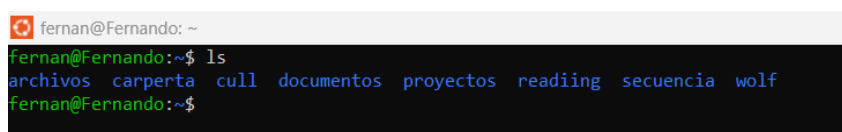
El comando pwd muestra la ruta completa del directorio actual, ayudando a identificar tu ubicación en el sistema de archivos.

2. Clear

A terminal window with a light gray title bar showing 'fernan@Fernando: ~'. The terminal text shows the user typing 'clear' and the prompt '~' appearing on the next line.

El comando clear limpia la pantalla de la terminal, eliminando todo el texto visible para proporcionar un espacio de trabajo limpio.

3. ls

A terminal window with a light gray title bar showing 'fernan@Fernando: ~'. The terminal text shows the user typing 'ls' and the output listing several files and directories: 'archivos carperta cull documentos proyectos readiing secuencia wolf'.

El comando ls lista los archivos y directorios en el directorio actual.

4. cd

```
fern@Fernando:~$ cd programas
fern@Fernando:~/programas$ cd ..
fern@Fernando:~$
```

El comando cd cambia el directorio de trabajo actual a un directorio especificado.

5. mkdir

```
fern@Fernando: ~
fern@Fernando:~$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
fern@Fernando:~$ mkdir proyectos
fern@Fernando:~$
```

El comando mkdir crea un nuevo directorio con el nombre especificado, lo que es útil para organizar y gestionar archivos y carpetas en la estructura de directorios.

6. rm

```
fern@Fernando: ~
fern@Fernando:~$ ls
archivos carpeta cull documentos laboratorios programas proyectos readiing secuencia wolf
fern@Fernando:~$ rm -r documentos
fern@Fernando:~$ ls
archivos carpeta cull laboratorios programas proyectos readiing secuencia wolf
fern@Fernando:~$ rm -r archivos
fern@Fernando:~$ ls
carpeta cull laboratorios programas proyectos readiing secuencia wolf
fern@Fernando:~$
```

El comando rm elimina archivos o directorios especificados, siendo útil para borrar elementos que ya no se necesitan.

7. mv

```
fern@Fernando: ~
fern@Fernando:~$ mkdir nuevodirectorio
fern@Fernando:~$ ls
carpeta cull laboratorios mi_directorio nuevodirectorio proyectos readiing wolf
fern@Fernando:~$ ls /home/
fern
fern@Fernando:~$ mv nuevodirectorio /home/fern@fern@mi_directorio/
mv: target '/mi_directorio/' is not a directory
fern@Fernando:~$ ls /home/fern@fern@
carpeta cull laboratorios mi_directorio nuevodirectorio proyectos readiing wolf
fern@Fernando:~$ mv nuevodirectorio /home/fern@fern@mi_directorio/
fern@Fernando:~$ ls /home/fern@fern@mi_directorio
nuevodirectorio
fern@Fernando:~$
```

Mueve o renombra archivos y directorios, facilitando la organización de datos en el sistema.

8. cp

```
fern@Fernando: ~
fern@Fernando:~$ ls
carpeta cull laboratorios mi_directorio proyectos readiing wolf
fern@Fernando:~$ cp -r carpeta proyectos/
fern@Fernando:~$ ls proyectos/
carpeta
fern@Fernando:~$
```

Copia archivos o carpetas a otro lugar, creando un duplicado en el sistema.

9. cat

```
fern@Fernando: ~  
fern@Fernando:~$ echo "este es un archivo de prueba. " > archivo.txt  
fern@Fernando:~$ cat archivo.txt  
este es un archivo de prueba.  
fern@Fernando:~$ echo "prueba dos comando cat. " > archivo.txt  
fern@Fernando:~$ cat archivo.txt  
prueba dos comando cat.  
fern@Fernando:~$
```

Muestra el contenido de archivos en la terminal, permitiendo ver o concatenar su contenido de manera rápida.

10. more y less

```
fern@Fernando: ~  
fern@Fernando:~$ ls  
archivo.txt carpeta cull laboratorios mi_directorio proyectos readiing wolf  
fern@Fernando:~$ more archivo.txt  
prueba dos comando cat.  
fern@Fernando:~$
```

```
fern@Fernando: ~  
fern@Fernando:~$ ls  
archivo.txt carpeta cull laboratorios mi_directorio proyectos readiing wolf  
fern@Fernando:~$ less archivo.txt
```

```
fern@Fernando: ~  
la informatica es una disciplina muy interesante que ayuda a crear nuevas avances tecnologicos en nustra era digital  
archivo.txt (END)
```

11. head

```
fern@Fernando: ~  
fern@Fernando:~$ ls  
archivo.txt carpeta cull informatica.txt laboratorios mi_directorio proyectos readiing wolf  
fern@Fernando:~$ more informatica.txt  
La informática o computación es la ciencia que estudia los métodos y técnicas para almacenar, procesar y transmitir información de manera automatizada, y más específicamente, en formato digital empleando sistemas computarizados.  
  
No existe realmente una definición única y universal de lo que la informática es, quizá porque se trata de una de las ciencias de más reciente origen, aunque de desarrollo más vertiginoso y desenfrenado.  
  
Por eso en muchos espacios académicos suelen diferenciar entre esta disciplina y las ciencias de la computación (o la ingeniería informática), considerando que estas últimas poseen un abordaje más teórico de la materia, mientras que la informática tiene siempre un costado práctico y aplicado, vinculado con los dispositivos electrónicos.  
  
Otros, en cambio, consideran como subdisciplinas de la informática a las Ciencias de la computación, la Ingeniería informática, los Sistemas de información, la Tecnología de la información y la Ingeniería del software.  
  
Al contrario de lo que se cree, la informática es anterior a la invención de las computadoras. Posee muy antiguos antecedentes en las máquinas de cálculo mentales de los filósofos de la Antigüedad Griega, como Euclides (c. 325-265 a. C.) y su famoso algoritmo, o bien en las calculadoras mecánicas del siglo XVII y las máquinas programables del siglo XIX.  
  
Sin embargo, en la primera mitad del siglo XX se creó la tecnología necesaria para desarrollar las primeras computadoras. Entre esos avances se encuentra el tubo al vacío, las puertas lógicas y los primeros circuitos, lo cual inauguró un campo del saber que muy pronto revolucionó a todos los demás y cambió la forma en que pensamos el trabajo.  
  
También fue central el trabajo en algoritmos durante las primeras tres décadas del siglo, bajo el genio de figuras como el matemático británico Alan Turing (1912-1954). Por otro lado, el contexto de la Segunda Guerra Mundial impulsó a los primeros calculadores automáticos que se dispusieron para descifrar los códigos de guerra del enemigo.  
  
La primera máquina de cálculo totalmente programable y automática fue inventada en 1941, llamada Z3, y en 1944 la primera máquina electromecánica, en la Universidad de Harvard en Estados Unidos: la Mark I.  
  
Desde entonces, los sistemas informáticos no cesaron de cambiar. Se incorporaron nuevas tecnologías como los transistores, los semiconductores y diversos mecanismos de almacenamiento de información, desde las tarjetas perforadas hasta las primeras cintas magnéticas.  
  
El primer computador de la historia fue el ENIAC (1946), de la Universidad de Pensilvania, que ocupaba un cuarto completo. Fue el primero de varias generaciones de sistemas informáticos por venir, cada vez más pequeños y potentes.  
  
Las primeras escuelas de computación, en el marco de las universidades, surgieron entre las décadas de 1950 y 1960. Paralelamente la industria informática, naciente pero poderosa, que permeó todas las demás áreas del saber humano en apenas 60 años de desarrollo.  
  
Hola pescad  
fern@Fernando:~$ head informatica.txt  
La informática o computación es la ciencia que estudia los métodos y técnicas para almacenar, procesar y transmitir información de manera automatizada, y más específicamente, en formato digital empleando sistemas computarizados.  
  
No existe realmente una definición única y universal de lo que la informática es, quizá porque se trata de una de las ciencias de más reciente origen, aunque de desarrollo más vertiginoso y desenfrenado.  
  
Por eso en muchos espacios académicos suelen diferenciar entre esta disciplina y las ciencias de la computación (o la ingeniería informática), considerando que estas últimas poseen un abordaje más teórico de la materia, mientras que la informática tiene siempre un costado práctico y aplicado, vinculado con los dispositivos electrónicos.  
  
Otros, en cambio, consideran como subdisciplinas de la informática a las Ciencias de la computación, la Ingeniería informática, los Sistemas de información, la Tecnología de la información y la Ingeniería del software.
```

Muestra las primeras líneas de un archivo, permitiendo ver rápidamente el comienzo del contenido.

12. tail

```
fern@Fernando: ~  
fern@Fernando:~$ ls  
archivo.txt carpeta cull informatica.txt laboratorios mi_directorio proyectos reading wolf  
fern@Fernando:~$ more informatica.txt  
La informática o computación es la ciencia que estudia los métodos y técnicas para almacenar, procesar y transmitir información de manera automatizada, y más específicamente, en formato digital empleando sistemas computarizados.  
  
No existe realmente una definición única y universal de lo que la informática es, quizá porque se trata de una de las ciencias de más reciente origen, aunque de desarrollo más vertiginoso y desenfrenado.  
  
Por eso en muchos espacios académicos suelen diferenciar entre esta disciplina y las ciencias de la computación (o la ingeniería informática), considerando que estas últimas poseen un abordaje más teórico de la materia, mientras que la informática tiene siempre un costado práctico y aplicado, vinculado con los dispositivos electrónicos.  
  
Otros, en cambio, consideran como subdisciplinas de la informática a las Ciencias de la computación, la Ingeniería informática, los Sistemas de información, la Tecnología de la información y la Ingeniería del software.  
  
Al contrario de lo que se cree, la informática es anterior a la invención de las computadoras. Posee muy antiguos antecedentes en las máquinas de cálculo mentales de los filósofos de la Antigüedad Griega, como Euclides (c. 325-265 a. C.) y su famoso algoritmo, o bien en las calculadoras mecánicas del siglo XVII y las máquinas programables del siglo XIX.  
  
Sin embargo, en la primera mitad del siglo XX se creó la tecnología necesaria para desarrollar las primeras computadoras. Entre esos avances se encuentra el tubo al vacío, las puertas lógicas y los primeros circuitos, lo cual inauguró un campo del saber que muy pronto revolucionó a todos los demás y cambió la forma en que pensamos el trabajo.  
  
También fue central el trabajo en algoritmos durante las primeras tres décadas del siglo, bajo el genio de figuras como el matemático británico Alan Turing (1912-1954). Por otro lado, el contexto de la Segunda Guerra Mundial impulsó a los primeros calculadores automáticos que se dispusieron para descifrar los códigos de guerra del enemigo.  
  
La primera máquina de cálculo totalmente programable y automática fue inventada en 1941, llamada Z3, y en 1944 la primera máquina electromecánica, en la Universidad de Harvard en Estados Unidos: la Mark I.  
  
Desde entonces, los sistemas informáticos no cesaron de cambiar. Se incorporaron nuevas tecnologías como los transistores, los semiconductores y diversos mecanismos de almacenamiento de información, desde las tarjetas perforadas hasta las primeras cintas magnéticas.  
  
El primer computador de la historia fue el ENIAC (1946), de la Universidad de Pensilvania, que ocupaba un cuarto completo. Fue el primero de varias generaciones de sistemas informáticos por venir, cada vez más pequeños y potentes.  
  
Las primeras escuelas de computación, en el marco de las universidades, surgieron entre las décadas de 1950 y 1960. Paralelamente la industria informática, naciente pero poderosa, que permeó todas las áreas del saber humano en apenas 60 años de desarrollo.  
  
Hola pescao  
fern@Fernando:~$ tail informatica.txt  
  
Desde entonces, los sistemas informáticos no cesaron de cambiar. Se incorporaron nuevas tecnologías como los transistores, los semiconductores y diversos mecanismos de almacenamiento de información, desde las tarjetas perforadas hasta las primeras cintas magnéticas.  
  
El primer computador de la historia fue el ENIAC (1946), de la Universidad de Pensilvania, que ocupaba un cuarto completo. Fue el primero de varias generaciones de sistemas informáticos por venir, cada vez más pequeños y potentes.  
  
Las primeras escuelas de computación, en el marco de las universidades, surgieron entre las décadas de 1950 y 1960. Paralelamente la industria informática, naciente pero poderosa, que permeó todas las áreas del saber humano en apenas 60 años de desarrollo
```

Muestra las últimas líneas de un archivo, permitiendo ver rápidamente el final del contenido.

Comandos de Manipulación Avanzada y Redirección

13. Echo

```
fern@Fernando: ~  
fern@Fernando:~$ ls  
archivo.txt carpeta cull informatica.txt laboratorios mi_directorio proyectos reading wolf  
fern@Fernando:~$ echo "hola mundo." > archivo.txt  
fern@Fernando:~$ cat archivo.txt  
hola mundo.  
fern@Fernando:~$
```

Muestra un texto o variable en la terminal, útil para imprimir mensajes y valores en scripts y comandos.

14. man

```
fernán@fernando: ~
LS(1) User Commands
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
  Mandatory arguments to long options are mandatory for short options too.
  -a, --all
    do not ignore entries starting with .
  -A, --almost-all
    do not list implied . and ..
  --author
    with -l, print the author of each file
  -b, --escape
    print C-style escapes for nongraphic characters
  --block-size=SIZE
    with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below
  -B, --ignore-backups
    do not list implied entries ending with ~
  -c
    with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first
  -C
    list entries by columns
  --color[=WHEN]
    colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below
  -d, --directory
    list directories themselves, not their contents
  -D, --dired
    generate output designed for Emacs' dired mode
  -f
    do not sort, enable -au, disable -ls --color
  -F, --classify
    append indicator (one of */>@) to entries
Manual page ls(1) line 1 (press h for help or q to quit)
```

Muestra el manual de referencia de otros comandos, proporcionando información detallada sobre su uso y opciones.

15. *(Wildcard)

```
fernán@fernando: ~
fernán@fernando:~$ ls *.txt
archivo.txt  informatica.txt
fernán@fernando:~$
```

Lista todos los archivos con extensión .txt en el directorio actual.

16. ? (Wildcard)

```
fernán@fernando: ~
fernán@fernando:~$ ls
arbol1  arbol3  carpeta  informatica.txt  mi_directorio  readiing
arbol2  archivo.txt  cull  laboratorios  proyectos  wolf
fernán@fernando:~$ ls arbol?
arbol1:
arbol2:
arbol3:
fernán@fernando:~$
```

Reemplaza a un solo carácter en patrones de búsqueda, permitiendo encontrar archivos con nombres similares pero con una variación de un solo carácter.

17. >

```
fern@Fernando: ~  
fern@Fernando:~$ echo "Hola, mundo" > saludo.txt  
fern@Fernando:~$ cat saludo.txt  
Hola, mundo  
fern@Fernando:~$
```

Redirige la salida de un comando a un archivo, sobrescribiendo el contenido del archivo o creándolo si no existe, facilitando la captura de resultados en lugar de mostrarlos en la pantalla.

18. <

```
fern@Fernando: ~  
fern@Fernando:~$ sort < archivo.txt  
hola mundo.  
fern@Fernando:~$
```

Redirige la entrada de un comando desde un archivo en lugar de usar la entrada estándar del teclado, permitiendo que el comando procese datos contenidos en el archivo.

19. |

```
fern@Fernando: ~  
fern@Fernando:~$ ls | grep "archivo"  
archivo.txt  
fern@Fernando:~$
```

Redirige la salida de un comando como entrada de otro comando, permitiendo encadenar múltiples comandos para procesar datos de manera secuencial.

20. &&

```
fern@Fernando: ~/otrodirectorio  
fern@Fernando:~$ mkdir nuevoescritorio && cd nuevoescritorio  
mkdir: cannot create directory 'nuevoescritorio': File exists  
fern@Fernando:~$ mkdir otrodirectorio && cd otrodirectorio  
fern@Fernando:~/otrodirectorio$
```

Ejecuta el segundo comando solo si el primero se ejecuta con éxito (es decir, retorna un estado de salida 0), permitiendo la ejecución condicional de comandos en secuencia.

21. ||

```
fernán@Fernando: ~  
fernán@Fernando:~$ mkdir pera || echo "no se pudo crear el directorio"  
fernán@Fernando:~$ ls  
arbol1  arbol3      carpeta  informatica.txt  mi_directorio  otrodirectorio  proyectos  saludo.txt  
arbol2  archivo.txt  cull     laboratorios     nuevoescritorio  pera           reading    wolf  
fernán@Fernando:~$
```

Ejecuta el segundo comando solo si la primera falla (es decir, retorna un estado de salida diferente de 0), permitiendo la ejecución condicional de comandos en caso de error.

Comandos de Permisos y Variables de Entorno

22. ls -l

```
fernán@Fernando: ~  
fernán@Fernando:~$ ls -l  
total 64  
drwxr-xr-x 2 fernan fernan 4096 Sep  4 11:51 arbol1  
drwxr-xr-x 2 fernan fernan 4096 Sep  4 11:51 arbol2  
drwxr-xr-x 2 fernan fernan 4096 Sep  4 11:51 arbol3  
-rw-r--r-- 1 fernan fernan  12 Sep  4 11:04 archivo.txt  
drwxr-xr-x 2 fernan fernan 4096 Sep  3 22:03 carpeta  
drwxr-xr-x 4 fernan fernan 4096 Sep  3 22:41 cull  
-rw-r--r-- 1 fernan fernan 3125 Sep  4 10:51 informatica.txt  
drwxr-xr-x 2 fernan fernan 4096 Sep  3 22:16 laboratorios  
drwxr-xr-x 3 fernan fernan 4096 Sep  4 10:09 mi_directorio  
drwxr-xr-x 3 fernan fernan 4096 Sep  4 14:10 nuevoescritorio  
drwxr-xr-x 2 fernan fernan 4096 Sep  4 14:13 otrodirectorio  
drwxr-xr-x 2 fernan fernan 4096 Sep  4 14:17 pera  
drwxr-xr-x 3 fernan fernan 4096 Sep  4 10:22 proyectos  
drwxr-xr-x 2 fernan fernan 4096 Sep  3 22:03 reading  
-rw-r--r-- 1 fernan fernan  12 Sep  4 13:55 saludo.txt  
drwxr-xr-x 2 fernan fernan 4096 Sep  3 22:03 wolf  
fernán@Fernando:~$
```

Muestra una lista detallada de los archivos y directorios en el directorio actual, incluyendo permisos, número de enlaces, propietario, grupo, tamaño y la fecha de la última modificación.

23. chmod

```
Seleccionar fernán@Fernando: ~  
fernán@Fernando:~$ chmod 755 script.sh  
fernán@Fernando:~$
```

Se usa para cambiar los permisos de acceso a archivos y directorios, especificando qué usuarios pueden leer, escribir o ejecutar el archivo.

24. chown

```
fernán@Fernando: ~  
fernán@Fernando:~$ sudo chown fernan:usuarios archivo.txt  
fernán@Fernando:~$ ls -l archivo.txt  
-rw-r--r-- 1 fernan usuarios 12 Sep  4 11:04 archivo.txt  
fernán@Fernando:~$
```

Cambia el propietario y/o el grupo de un archivo o directorio, permitiendo asignar nuevas propiedades de usuario y grupo.

25. chgrp

```
Seleccionar fernan@Fernando: ~
fernand@Fernando:~$ chgrp usuarios archivo.txt
fernand@Fernando:~$ ls -l archivo.txt
-rw-r--r-- 1 fernan usuarios 12 Sep 4 11:04 archivo.txt
fernand@Fernando:~$
```

Cambia el grupo propietario de un archivo o directorio, permitiendo asignar un nuevo grupo al archivo o directorio especificado

26. export

```
Seleccionar fernan@Fernando: ~
fernand@Fernando:~$ export EDITOR= nano
fernand@Fernando:~$
```

Establece o exporta variables de entorno en el shell, haciendo que las variables definidas estén disponibles para los procesos hijos y otros comandos en el entorno de ejecución.

27. env

```
fernand@Fernando: ~
fernand@Fernando:~$ env
SHELL=/bin/bash
WSL2_GUI_APPS_ENABLED=1
WSL_DISTRO_NAME=Ubuntu
EDITOR=
NAME=fernando
PWD=/home/fernand
LOGNAME=fernand
HOME=/home/fernand
LANG=C.UTF-8
WSL_INTEROP=/run/WSL/10253_interop
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:cd=40;33:or=40;31:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lzh=01;31:*.lzm=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tztst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz2=01;31:*.tbz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.vdi=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.ppm=01;35:*.tga=01;35:*.*.bm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcc=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
MAYLAND_DISPLAY=wayland-0
LESSCLOSE=/usr/bin/lesspipe %s %s
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=fernand
DISPLAY=:0
SHLVL=1
XDG_RUNTIME_DIR=/run/user/1000/
WSLENV=
XDG_DATA_DIRS=/usr/local/share:/usr/share:/var/lib/snapd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/wsl/lib:/mnt/c/Program Files/Common Files/Oracle/Java/javapath:/mnt/c/Program Files (x86)/Common Files/Oracle/Java/javapath:/mnt/c/Program Files (x86)/Common Files/Oracle/Java/javapath:/mnt/c/windows/system32:/mnt/c/windows:/mnt/c/windows/System32/Wbem:/mnt/c/windows/System32/WindowsPowerShell/v1.0:/mnt/c/windows/System32/OpenSSH:/mnt/c/Users/Fernando_Magallon/AppData/Local/Microsoft/WindowsApps:/mnt/c/Users/Fernando_Magallon/AppData/Local/Programs/Microsoft VS Code/bin:/snap/bin
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
HOSTTYPE=x86_64
PULSE_SERVER=unix:/mnt/wslg/PulseServer
_/usr/bin/env
fernand@Fernando:~$
```

Muestra las variables de entorno actuales del sistema o ejecuta un comando en un entorno modificado, permitiendo verificar o cambiar el entorno en el que se ejecuta un comando.

28. unset

```
fern@Fernando: ~  
fern@Fernando:~$ unset EDITOR  
fern@Fernando:~$
```

Elimina una variable de entorno o una variable de shell, quitando su definición y liberando el nombre para su uso futuro.

Comandos de Red y Gestión de Procesos

29. ping

```
fern@Fernando: ~  
fern@Fernando:~$ ping google.com  
PING google.com (172.217.3.78) 56(84) bytes of data.  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=1 ttl=110 time=65.9 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=2 ttl=110 time=71.6 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=3 ttl=110 time=137 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=4 ttl=110 time=65.3 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=5 ttl=110 time=66.6 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=6 ttl=110 time=67.6 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=7 ttl=110 time=65.8 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=8 ttl=110 time=63.8 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=9 ttl=110 time=66.2 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=10 ttl=110 time=72.1 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=11 ttl=110 time=69.9 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=12 ttl=110 time=83.2 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=13 ttl=110 time=67.5 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=14 ttl=110 time=1070 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=15 ttl=110 time=67.0 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=16 ttl=110 time=68.5 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=17 ttl=110 time=68.3 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=18 ttl=110 time=68.2 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=19 ttl=110 time=66.1 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=20 ttl=110 time=68.2 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=21 ttl=110 time=72.4 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=22 ttl=110 time=168 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=23 ttl=110 time=90.5 ms  
64 bytes from mia07s54-in-f14.1e100.net (172.217.3.78): icmp_seq=24 ttl=110 time=117 ms
```

Envía paquetes de datos a una dirección IP o nombre de dominio para verificar la conectividad de red y medir el tiempo de respuesta.

30. ifconfig

```
fern@Fernando: ~  
fern@Fernando:~$ ifconfig  
Command 'ifconfig' not found, but can be installed with:  
sudo apt install net-tools  
fern@Fernando:~$ sudo apt install net-tools  
[sudo] password for fernan:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  net-tools  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 204 kB of archives.  
After this operation, 819 kB of additional disk space will be used.  
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]  
Fetched 204 kB in 2s (112 kB/s)  
Selecting previously unselected package net-tools.  
(Reading database ... 24208 files and directories currently installed.)  
Preparing to unpack .../net-tools 1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...  
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...  
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...  
Processing triggers for man-db (2.10.2-1) ...  
fern@Fernando:~$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.26.214 netmask 255.255.240.0 broadcast 192.168.31.255  
    inet6 fe80::215:5dff:feb2:f0e2 prefixlen 64 scopeid 0x20<link>  
    ether 00:15:5d:b2:f0:e2 txqueuelen 1000 (Ethernet)  
    RX packets 162 bytes 216042 (216.0 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 79 bytes 5460 (5.4 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 10 bytes 1278 (1.2 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 10 bytes 1278 (1.2 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
fern@Fernando:~$
```

Muestra y configura las interfaces de red en sistemas Unix y Linux, permitiendo ver detalles como la dirección IP, la máscara de red y el estado de cada interfaz.

31. tar

```
fernán@Fernando: ~  
fernán@Fernando:~$ tar -cvf archivo.tar /home/usuarios/proyectos  
tar: Removing leading `/' from member names  
/home/usuarios/proyectos/  
fernán@Fernando:~$
```

Crea, modifica o extrae archivos archivados (tarballs), permitiendo empaquetar varios archivos y directorios en un solo archivo para facilitar su almacenamiento o transferencia.

32. zip

```
fernán@Fernando: ~  
fernán@Fernando:~$ zip -r archivos.zip /home/usuarios/proyectos  
adding: home/usuarios/proyectos/ (stored 0%)  
fernán@Fernando:~$
```

Comprime archivos y directorios en un archivo comprimido en formato ZIP, reduciendo su tamaño y facilitando su almacenamiento o transferencia.

33. unzip

```
fernán@Fernando: ~  
fernán@Fernando:~$ unzip archivos.zip  
Archive:  archivos.zip  
creating: home/usuarios/proyectos/  
fernán@Fernando:~$
```

Extrae archivos y directorios de un archivo comprimido en formato ZIP, restaurando su contenido original.

34. ps

```
fernán@Fernando: ~  
fernán@Fernando:~$ ps  
  PID TTY          TIME CMD  
   360 pts/0    00:00:00 bash  
   6089 pts/0    00:00:00 ps  
fernán@Fernando:~$
```

Muestra una lista de los procesos en ejecución en el sistema, proporcionando información sobre cada proceso como su ID, estado y uso de recursos.

35. Kill

```
fernán@Fernando: ~  
fernán@Fernando:~$ kill 1234  
-bash: kill: (1234) - No such process  
fernán@Fernando:~$
```

Envía una señal a un proceso para terminarlo o para solicitarle que realice una acción específica, como detenerse o reiniciarse, usando su ID de proceso (PID).

36. &

```
fernán@Fernando: ~  
fernán@Fernando:~$ sleep 30 &  
[1] 7268  
fernán@Fernando:~$
```

Al final de un comando ejecuta el comando en segundo plano, permitiendo que la terminal siga disponible para otros comandos mientras el proceso se ejecuta en segundo plano.

37. fg

```
fernán@Fernando: ~  
fernán@Fernando:~$ fg  
-bash: fg: current: no such job  
fernán@Fernando:~$
```

Trae un proceso en segundo plano al primer plano, permitiendo interactuar con él directamente en la terminal.

Comandos de Edición y Personalización

38. nano

```
fernán@Fernando: ~  
fernán@Fernando:~$ nano archivo.txt  
fernán@Fernando:~$
```

```
fernán@Fernando: ~  
GNU nano 6.2  
hola mundo.
```

Abre un editor de texto en la terminal que permite crear y editar archivos de texto de manera sencilla e intuitiva.

Define un nombre corto o un comando personalizado para un comando o una serie de comandos más largos, facilitando su uso y ahorro de tiempo en la línea de comandos.