

Práctica



Aprendiendo Practicando
GNU/Básicos
Ing. José Paredes

www.codigolibre.org y www.acl.edu.do

Aprendiendo Practicando GNU/Básico

En esta practica usted aprenderá: Shell, Comandos Básicos, Comandos Unix/Linux, Manejo de archivos, Variables, Archivos Ocultos, Completando Comandos con Tab, Alias, Comando History, Jerarquía de Directorios, Expresiones Regulares y Flujo de Texto, Comodines, Tubería y Redireccionamiento .

Nota:

Esto es el prompt tomado de ejemplo, luego del símbolo \$ estará el comando que debe ejecutar. Cuando sean combinaciones de teclas, estarán en mayúscula.

```
[estudiantes@acl ~]$
```

Si necesita hacerlo como Root estará el prompt en r0j0 y con el símbolo “#”

Para acceder como Root:

```
[estudiantes@acl ~]$ su -
```

Listo!

```
[root@acl ~]#
```

Luego de ejecutar cada comando tiene que ejecutarlo utilizando la tecla “Enter”

GNUNota: Los comandos se escriben en minúscula.

1. Vamos iniciar una sesión CTRL+ALT+F2.

acl login:

Ingresa el usuario creado cuando realizó la instalación.

```
acl login: estudiantes
```

(Oprima Enter)

Password: **solucion**

(Oprima Enter)

```
[estudiantes@acl ~]$
```

GNUNota: Hagamos este ejercicio por lo menos con tres consola:

CTRL+ALT+F2

CTRL+ALT+F3

CTRL+ALT+F4

2. Tenemos dos formas para salir o cerrar la sesión que iniciamos con el usuario estudiantes.

```
CRTL+d
```

o

```
[estudiantes@acl ~]$ exit
```

GNUNota: Ahora vuelva a su X y habrá una pts y continúe con los ejercicios desde el X.

3. Eso que usted ve que esta pestañeando se llama Prompt y le está indicando que puede iniciar a ejecutar los comandos.

```
[estudiantes@acl ~]$ _
```

4. Nuestro primer comando es man man.

```
[estudiantes@acl ~]$ man man
```

Para salir del man presione la tecla "q"

5. Vamos a obtener una definición del comando ls y sus opciones.

```
[estudiantes@acl ~]$ man ls
```

6. Ya usted sabe que función realiza el programa ls, vamos a listar el contenido del directorio donde nos encontramos.

```
[estudiantes@acl ~]$ ls
```

7. Mostrar una lista de comandos que pertenecen a las palabras claves de un programa; son útiles cuando tú sabes qué hace tu programa, pero desconoces el nombre del comando.

```
[estudiantes@acl ~]$ apropos web
```

8. Para obtener un detalle mas amplio vamos a iniciar agregándole al comando ls una opción que usted puede ver dentro del man.

```
[estudiantes@acl ~]$ ls -l
```

9. Vamos a listar los archivos ocultos.

```
[estudiantes@acl ~]$ ls -a
```

GNUNota: Los archivos ocultos inician con un punto "." y solo son visibles con esta opción.

10. Vamos a listar los archivos identificando su tipo por un símbolo que esta en detalle dentro del man.

```
[estudiantes@acl ~]$ ls -F /dev
```

11. Vamos a combinar varias opciones en una sola ejecución.

```
[estudiantes@acl ~]$ ls -ltrh
```

```
GNUNota: Mi preferida.
```

12. Mostrar el tamaño de los ficheros y directorios ordenados por tamaño.

```
[estudiantes@acl ~]$ ls -lSr
```

13. Mostrar los ficheros y carpetas que contienen números.

```
[estudiantes@acl ~]$ ls /etc/*[0-9]*
```

14. En qué directorio usted se encuentra actualmente.

```
[estudiantes@acl ~]$ pwd
```

15. Ahora vamos navegar dentro del sistema de archivos, luego despliegue el contenido del mismo.

```
[estudiantes@acl ~]$ cd /dev
```

```
[estudiantes@acl dev]$ ls
```

```
[estudiantes@acl dev]$ cd /opt
```

```
[estudiantes@acl opt]$ ls
```

```
[estudiantes@acl opt]$ cd /etc
```

```
[estudiantes@acl etc]$ ls
```

```
[estudiantes@acl etc]$ cd /boot
```

16. Vamos a dirigirnos a la barra y confirme que esta en la barra.

```
[estudiantes@acl boot]$ cd /
```

```
[estudiantes@acl /]$ pwd
```

17. Regrese al ultimo directorio donde usted estaba.

```
[estudiantes@acl boot]$ cd -
```

```
GNUNota: Esto lo llevará hasta /boot que era el directorio que usted visitó antes de ir a la barra, compruebe que esta ahí utilizando el comando "pwd".
```

18. Vamos realizar cuatro manera de regresar a su HOME.

```
[estudiantes@acl boot]$ cd ~ (Oprima Enter)
```

```
[estudiantes@acl ~]$ cd $HOME (Oprima Enter)
```

```
[estudiantes@acl ~]$ cd /home/estudiantes/ (Oprima Enter)
```

```
[estudiantes@acl ~]$ cd
```

GNUNota: El símbolo “~” nos indica que estamos en nuestro Home, territorio personal de cada usuario dentro del directorio /home por defecto.

19. Vamos a crear un directorio en nuestro home utilizando la forma relativa.

```
[estudiantes@acl ~]$ mkdir fclld (Oprima Enter)
```

GNUNota: Si el shell no devuelve ningún error es que la probabilidad de ejecución del mismo fue exitosamente de un 99%.

20. Vamos a verificar si el directorio fue creado.

```
[estudiantes@acl ~]$ ls fclld/
```

GNUNota: El resultado de esta sentencia fue silenciosa o no tuvo resultado, ¿sabe el por qué? Por que el directorio creado esta vacío.

21. Error cuando digitamos incorrectamente.

```
[estudiantes@acl ~]$ ls fcl/
```

GNUNota: Este seria el error que nos imprime el shell si el nombre del directorio no es digitado correctamente.

22. Vamos acceder al directorio fclld.

```
[estudiantes@acl ~]$ cd fclld
```

23. Imprima la ruta donde se encuentra actualmente.

```
[estudiantes@acl fclld]$ pwd
```

24. Como nuestro directorio esta vacío vamos copiar un archivo en el mismo directorio que creamos.

```
[estudiantes@acl fclld]$ cp /etc/passwd /home/estudiantes/fclld/
```

GNUNota: A esto le llamamos copiando con Ruta Absoluta.

25. Ahora vamos a realizar el mismo ejercicio anterior pero mas simple.

```
[estudiantes@acl fclld]$ cp /etc/group .
```

GNUNota: Como estamos dentro del directorio donde queremos copiar el archivo solo agregamos un punto "." para decirle al shell que es aquí.

26. Realizaremos una copia del archivo "passwd" pero con otro nombre.

```
[estudiantes@acl fclld]$ cp /etc/passwd ~/fclld/clave.pdf
```

GNUNota: Aquí le agregamos un símbolo "~" para indicar que el directorio donde vamos a realizar la copia esta dentro de nuestro Home.

27. Confirme que los últimos tres ejercicios se han realizado con éxito.

```
[estudiantes@acl fclld]$ ls
```

28. Elimine los archivos copiados en el directorio fclld.

```
[estudiantes@acl fclld]$ rm *
```

```
[estudiantes@acl fclld]$ ls
```

29. Vamos realizar la copia de los archivos otra vez pero utilizando opciones.

```
[estudiantes@acl fclld]$ cp -vi /etc/passwd .
```

```
[estudiantes@acl fclld]$ cp -vir /etc/passwd /home/estudiantes/fclld/clave.pdf
```

GNUNota: Buscar en el man las definiciones de estas opciones.

30. Vamos a crear cuatro directorios dentro de fclld.

```
[estudiantes@acl fclld]$ mkdir libre bolges
```

```
[estudiantes@acl fclld]$ mkdir -p libre/clase bolges/santiago
```

```
[estudiantes@acl fclld]$ ls
```

GNUNota: Buscar en el man las definiciones de la opción "-p".

31. Vamos a listar el contenido del directorio fclld en forma de jerárquica y a la vez confirmaremos la creación de los directorios en el ejercicio 27.

```
[estudiantes@acl fclld]$ tree
```

GNUNotas: Si el comando tree no está instalado, llame su GNU/Instructor y solicite ayuda.

32. Determine qué tipo de archivo es clave.pdf y passwd.

```
[estudiantes@acl fcld]$ file clave.pdf
```

```
[estudiantes@acl fcld]$ file passwd
```

33. Renombre el archivo clave.pdf y llámelo secreto.

```
[estudiantes@acl fcld]$ mv clave.pdf secreto
```

```
[estudiantes@acl fcld]$ ls
```

34. Mueva el archivo secreto hacia el directorio libre.

```
[estudiantes@acl fcld]$ mv -v secreto libre/
```

35. Entre en el directorio libre y verifique que el archivo secreto esta hay.

```
[estudiantes@acl fcld]$ cd libre/
```

```
[estudiantes@acl libre]$ ls
```

36. Diga la ruta absoluta donde usted se encuentra actualmente.

```
[estudiantes@acl libre]$ pwd
```

37. Regrese al directorio que esta anterior a libre y comprueba que esta en el directorio fcld.

```
[estudiantes@acl libre]$ cd ..
```

```
[estudiantes@acl ~]$ pwd
```

```
[estudiantes@acl fcld]$ cd libre
```

GNUNota: Investigar la función de los “..” y explicar al GNU/Instructor.

38. Retroceder 2 niveles.

```
[estudiantes@acl libre]$ cd ../../
```

```
[estudiantes@acl ~]$ pwd
```

```
[estudiantes@acl ~]$ ls ; cd fcld/
```

39. Mover el directorio libre dentro de bolges.

```
[estudiantes@acl fcld]$ mv libre/ bolges/
```

```
[estudiantes@acl fcld]$ ls
```

40. Borre el archivo passwd con la opción -i.

```
[estudiantes@acl fcld]$ rm -i passwd
```

```
[estudiantes@acl fcld]$ ls
```

GNUNota: Buscar en el man la definición de la opción "-i", comparta esa información.

41. Ejecute el comando tree y determine en qué directorio está el archivo Secreto.

```
[estudiantes@acl fcld]$ tree
```

42. Copie el archivo passwd que se encuentra en /etc al directorio clase.

```
[estudiantes@acl fcld]$ cp /etc/passwd bolges/libre/clase/
```

```
[estudiantes@acl fcld]$ ls bolges/libre/clase/
```

43. Crear un archivo vacío.

```
[estudiantes@acl fcld]$ touch new.txt
```

```
[estudiantes@acl fcld]$ ls
```

GNUNota: Explique a su GNU/Instructor las funciones del comando "touch".

44. Vamos ver el contenido del archivo passwd y enumerado.

```
[estudiantes@acl fcld]$ cd bolges/libre/clase/
```

```
[estudiantes@acl clase]$ ls
```

```
[estudiantes@acl clase]$ cat passwd
```

```
[estudiantes@acl clase]$ cat -n passwd
```

45. Utilizando el comando cat pero con una ruta absoluta.

```
[estudiantes@acl clase]$ cat /etc/group
```

46. Imprimiendo el archivo passwd que este en el directorio clase con el comando more y less.

```
[estudiantes@acl clase]$ more passwd
```

GNUNota: Utilice la barra espaciadora para avanzar en el archivo.

```
[estudiantes@acl clase]$ less passwd
```


GNUNota: Utilice la barra espaciadora para avanzar en el archivo y las teclas “arriba” y “abajo”, investigue la diferencia entre less y more.

47. Vamos a contar la cantidad de líneas, palabras y caracteres del archivo passwd.

```
[estudiantes@acl clase]$ wc passwd
```

48. Vamos a imprimir solo la cantidad de líneas del archivo passwd.

```
[estudiantes@acl clase]$ wc -l passwd
```

GNUNota: Realice varios ejemplo utilizando las opciones “-m” y“-w”.

49. Vamos a imprimir las 10 primeras líneas del archivo passwd.

```
[estudiantes@acl clase]$ head passwd
```

50. Vamos imprimir solo las 4 primeras líneas del archivo passwd.

```
[estudiantes@acl clase]$ head -n 4 passwd
```

GNUNota: Realice un ejemplo utilizando la opción “-v”.

51. Vamos a imprimir las últimas 10 líneas del archivo passwd.

```
[estudiantes@acl clase]$ tail passwd
```

52. Vamos imprimir solo las 4 últimas líneas del archivo passwd.

```
[estudiantes@acl clase]$ tail -vn 4 passwd
```

GNUNota: Investigue qué función hacen las opciones “-v” y “-f”.

53. Ver en tiempo real qué ha sido añadido al fichero.

```
[estudiantes@acl clase]$ tail -f /var/log/messages
```

GNUNota: Cambie para Super-Usuario, para que pueda realizar este ejercicio :).

54. Copie el archivo group que está en /etc en el mismo directorio donde se encuentra.

```
[estudiantes@acl clase]$ cp /etc/group .
```

```
[estudiantes@acl clase]$ ls
```

55. Busque la diferencia entre los archivos passwd y group.

```
[estudiantes@acl clase]$ cp passwd passwd2
```

```
[estudiantes@acl clase]$ diff passwd passwd2
```

```
[estudiantes@acl clase]$ echo "Jose Paredes" >> passwd2
```

```
[estudiantes@acl clase]$ diff passwd passwd2
```

GNUNota: Investigue la salida del mismo.

56. Vamos a imprimir la fecha del sistema.

```
[estudiantes@acl clase]$ date
```

57. Obtener la fecha del sistema en formato (año, mes, día).

```
[estudiantes@acl clase]$ date +%F
```

58. Obtener la fecha del sistema en formato (mes, día, año).

```
[estudiantes@acl clase]$ date +%D
```

59. Obtener el nombre del mes de la fecha actual del sistema.

```
[estudiantes@acl clase]$ date +%B
```

60. Obtener la fecha del sistema en formato String June-Friday-15-2012.

```
[estudiantes@acl clase]$ date +%B-%A-%d-%Y
```

61. Obtener el siguiente día a partir de la fecha actual y anterior a la actual.

```
[estudiantes@acl clase]$ date -d "tomorrow"
```

```
[estudiantes@acl clase]$ date -d "yesterday"
```

62. Obtener fechas pasadas o futuras, en este caso saber que día cae el próximo viernes.

```
[estudiantes@acl clase]$ date -d "next friday"
```

63. Vamos desplegar el calendario.

```
[estudiantes@acl clase]$ cal
```

GNUNota: Agregar 2013 para obtener la salida del calendario de ese año.

64. Mostrar el almanaque del 2014.

```
[estudiantes@acl clase]$ cal 2014
```

65. Mostrar el almanaque para el mes agosto del 2014.

```
[estudiantes@acl clase]$ cal 08 2014
```

66. Guardar los cambios de fecha en la BIOS.

```
[estudiantes@acl clase]$ clock -w
```

GNUNota: Cambie para Super-Usuario, para que pueda realizar este ejercicio :).

67. Con que usuario esta usted trabajando.

```
[estudiantes@acl clase]$ whoami
```

68. Limpiar la pantalla del shell.

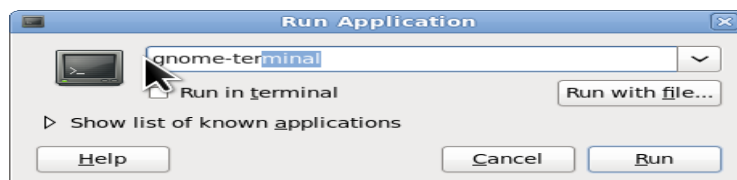
```
[estudiantes@acl clase]$ clear
```

GNUNota: Explique la diferencia de clear y la combinación de tecla "CTRL+L".

69. Cierre esa pts.

```
[estudiantes@acl clase]$ exit
```

GNUNota: Ahora combine ALT+F2 y abra una pts y vuelva entrar al directorio fclD.



70. Vamos a imprimir una cadena de texto con el comando echo.

```
[estudiantes@acl fclD]$ echo "Hola Estudiantes de la FCLD"
```

71. Imprimir dos líneas con el comando echo.

```
[estudiantes@acl fclD]$ echo -e "\\043 \\b1 Hola a todos \\n \\b\\043 \\b2  
Bienvenidos
```

72. Imprimir una cadena con tabulación, separando las palabras.

```
[estudiantes@acl fclD]$ echo -e 'Hola\\tEstudiantes'
```

73. Imprimir una cadena con la opción del tabulador vertical.

```
[estudiantes@acl fcld]$ echo -e 'Hola\vEstudiantes'
```

74. Verifique el tiempo que tiene su máquina encendida.

```
[estudiantes@acl fcld]$ uptime
```

75. Vamos a localizar dónde se encuentra el archivo `.bash_profile`.

```
[estudiantes@acl fcld]$ locate .bash_profile
```

76. En el ejercicio 43 usted creó un archivo llamado `new.txt`, localice la ubicación de ese archivo con el comando `locate`.

```
[estudiantes@acl fcld]$ locate new.txt
```

GNUNota: No lo encontró, en el siguiente ejercicio usted aprenderá el porqué.

77. Ahora actualice la base datos del comando `locate`.

```
[estudiantes@acl fcld]$ su -
```

```
[root@acl ~]# updatedb
```

```
[estudiantes@acl fcld]$ exit
```

```
[estudiantes@acl fcld]$ locate new.txt
```

```
[estudiantes@acl fcld]$ locate .bash_profile
```

78. Vamos buscar el archivo `new.txt` con el comando `find`.

```
[estudiantes@acl fcld]$ find ~ -iname "new.txt"
```

GNUNota: Utilizando el `man` investigue la diferencia entre `"name"` y `"iname"`.

79. Buscar ficheros binarios no usados en los últimos 100 días.

```
[estudiantes@acl ~]$ find /usr/bin -type f -atime +100
```

80. Buscar ficheros con extensión `'.rpm'` ignorando los dispositivos removibles como `cdrom`, `pen-drive`, etc...

```
[estudiantes@acl ~]$ find / -xdev -name *.rpm
```

81. Usted no recuerda dónde almacenó sus videos, solo recuerda el formato en que están `"avi"`. Vamos a buscar en todo el disco.

```
[root@acl ~]# find / -iname "*.avi"
```

82. El administrador de un servidor de Internet necesita realizar una auditoría de seguridad, para ello una de las pruebas que se necesita realizar es identificar aquellos archivos o directorios que poseen permisos de escritura para cualquier usuario, esto lo puede hacer como sigue:

```
[root@acl ~]# find / -perm 777
```

83. Este comando listará también los enlaces simbólicos, que aunque en el listado aparecen con todos los permisos activados, no significa que cualquier usuario los pueda modificar. Para evitar entonces este inconveniente, se puede ejecutar el comando de esta manera:

```
[root@acl ~]# find / -perm 777 -follow
```

84. Buscar todos los archivos que han sido creados o modificados en la última 24 horas en su HOME.

```
[estudiantes@acl fcltd]$ find $HOME -mtime 0
```

85. Buscar los archivos en su home cuyo tamaño sean mayor a 100 mega.

```
[estudiantes@acl fcltd]$ find /home/estudiantes/ -size +100M
```

GNUNota: Realizar el mismo ejercicio pero para menores a 100 megas y mayores a 2 Gigabytes.

86. En este ejemplo se buscan todos los archivos cuyo nombre termina con `*wd` en el directorio `/etc` y muestra el contenido de cada uno de ellos.

```
[estudiantes@acl fcltd]$ find /etc -name "*wd" -exec cat {} \;
```

87. Vamos a buscar los archivos donde el usuario tenga permiso de escritura y el grupo también.

```
[estudiantes@acl bolges]$ find . -perm /u=w,g=w
```

88. Listado de ficheros modificados en menos de 48 horas, en el directorio actual.

```
[estudiantes@acl fcltd]$ find . -mtime -1
```

89. Otro uso habitual es con `xargs` para lanzar un comando sobre cada resultado. Ejemplo: tocar todos los `.mp3` en una carpeta.

```
[estudiantes@acl fcltd]$ find $HOME -iname "*.mp3" | xargs mplayer
```

90. Seleccionar todas las líneas del fichero `'/var/log/messages'` que contienen números.

```
[root@acl ~]# grep [0-9] /var/log/messages
```

91. Vamos a visualizar las variables de entorno y sus valores.

```
[estudiantes@acl fclld]$ env
```

GNUNota: Utilice tubería "|" y less para compaginar.

92. Vamos a visualizar las variables del sistema.

```
[estudiantes@acl fclld]$ set
```

GNUNota: Utilice tubería "|" y less para compaginar.

93. Vamos ver todas las variables de nuestro ambiente.

```
[estudiantes@acl fclld]$ echo $Tab Tab
```

GNUNota: Tecla "Tab" "Tab" significa que usted debe de teclear dos veces la tecla del tabulador, ubicada encima de CapsLk (Mayúscula).

94. Imprimamos algunas variables y observemos su valor.

```
[estudiantes@acl fclld]$ echo $BASH
```

```
[estudiantes@acl fclld]$ echo $PATH
```

```
[estudiantes@acl fclld]$ echo $USER
```

```
[estudiantes@acl fclld]$ echo $PS1
```

```
[estudiantes@acl fclld]$ echo $LS_COLORS
```

```
[estudiantes@acl fclld]$ echo $HOSTNAME
```

GNUNota: Para ver el valor de una variable debe escribir el nombre en mayúscula.

95. Crearemos una variable llamada GILRS.

```
[estudiantes@acl fclld]$ GIRLS='fedora bella'
```

96. Vamos a ver el valor de la variable creada en el ejercicio anterior.

```
[estudiantes@acl fclld]$ echo $GIRLS
```

97. Busquemos la variable a ver si esta en nuestro entorno.

```
[estudiantes@acl fclld]$ env |grep -i GIRLS
```

GNUNota: Más adelante aprenderá el comando grep. Continúe :).

98. Vamos a exportar la variable GILRS a nuestro ambiente.

```
[estudiantes@acl fcld]$ export GIRLS
```

99. Ahora repita el ejercicio 98 para confirmar que fue exportada.

```
[estudiantes@acl fcld]$ env |grep -i GIRLS
```

GNUNota: Investigue dónde debe colocar esta variable si quiere que sea permanente.

100. Vamos a ver todos los alias que tenemos disponibles.

```
[estudiantes@acl fcld]$ alias
```

101. Vamos crear un alias llamado fcld, este alias limpiara la pantalla.

```
[estudiantes@acl fcld]$ alias fcld='clear'
```

GNUNota: Para ejecutar el alias solo digite fcld en el shell donde lo creo.

102. Ahora elimine el alias creado en el ejercicio.

```
[estudiantes@acl fcld]$ unalias fcld
```

GNUNota: Investigue como podría dejar el alias permanente.

103. Vamos a listar todos los comandos o eventos realizados en este shell.

```
[estudiantes@acl fcld]$ history
```

104. Mostrar los últimos 5 comandos del history.

```
[estudiantes@acl fcld]$ history 5
```

105. Vamos a ejecutar el último comando del history.

```
[estudiantes@acl fcld]$ !!
```

106. De la lista imprimida en pantalla del comando history vamos a ejecutar el numero 204.

```
[estudiantes@acl fcld]$ !204
```

107. Por defecto el history almacena solo las ultima 500 actividades del shell si queremos cambiar a 5000.

```
[estudiantes@acl fcld]$ echo "export HISTSIZE=5000" >> ~/.bashrc
```

GNUNota: Investigue otra forma de modificar el almacenamiento del history.

108. Si queremos borrar el comando que se encuentra en la posición 211.

```
[estudiantes@acl fclld]$ history -d 211
```

109. Si queremos leer los contenidos del fichero historial y usarlos como el historial actual.

```
[estudiantes@acl fclld]$ history -r
```

110. Si queremos borrar la lista del historial (borrando todas las entradas).

```
[estudiantes@acl fclld]$ history -c
```

GNUNota: Ahora compruebe que fueron borrados.

111. Vamos mostrar el contenido del archivo passwd en orden inverso.

```
[estudiantes@acl fclld]$ tac /etc/passwd
```

GNUNota: Explicar al GNU/Instructor la diferencia con el comando cat.

112. Vamos ordenar el archivo secreto en forma ascendente.

```
[estudiantes@acl fclld]$ sort secreto
```

113. Vamos a desplegar el archivo secreto ordenado en forma descendente.

```
[estudiantes@acl fclld]$ sort -r secreto
```

114. Queremos ver un listado de los usuarios ordenado por uid.

```
[estudiantes@acl fclld]$ sort -t":" -k3n /etc/passwd
```

115. Vamos a crear un archivo que contenga líneas repetidas para luego eliminar las repetidas del archivo y dejar su contenido.

```
[estudiantes@acl fclld]$ cat > frutas.txt
```

```
mango
mango
guineos
guineos
pina
melon
```

CRTL+D


```
[estudiantes@acl fcltd]$ sort -o frutas.txt -u frutas.txt
```

```
[estudiantes@acl fcltd]$ cat frutas.txt
```

```
guineos  
mango  
melon  
pina
```

116. Vamos a crear un archivo que contenga líneas separadas por coma llamado cute.txt.

```
[estudiantes@acl fcltd]$ cat > cute.txt  
Columna1,Columna2,Columna3  
1,2,3  
4,5,6  
7,8,9
```

117. Vamos imprimir la tercera columna del archivo cute.txt con el comando cut:

```
[estudiantes@acl fcltd]$ cut -d, -f3 cute.txt
```

118. Vamos imprimir la tercera y segunda columna del archivo cute.txt con el comando cut:

```
[estudiantes@acl fcltd]$ cut -d, -f3,2 cute.txt
```

119. Vamos a sacar un texto por pantalla, utilizando como delimitador por ejemplo la coma, podemos dividir en 3 partes el texto, podemos mostrar cada una de esas tres partes (fields) con la opción -f.

```
[estudiantes@acl fcltd]$ echo "Esto es una prueba, 1 2 3, probando" | cut -d "," -f 1
```

```
[estudiantes@acl fcltd]$ echo "Esto es una prueba, 1 2 3, probando" | cut -d "," -f 2
```

```
[estudiantes@acl fcltd]$ echo "Esto es una prueba, 1 2 3, probando" | cut -d "," -f 3
```

```
[estudiantes@acl fcltd]$ echo "Esto es una prueba, 1 2 3, probando" | cut -c 1-4
```

```
[estudiantes@acl fcltd]$ echo "Esto es una prueba, 1 2 3, probando" | cut -c 6-8
```

GNUNota: Explicar al GNU/Instructor la opción "-c".

120. Realice el siguiente ejercicio y explicar su funcionamiento.

```
[estudiantes@acl fcltd]$ cut -d" " -f1,5 <<< "hello, my name is mysurface."
```

121. Vamos a buscar el usuario que estamos utilizando en el archivo passwd.

```
[estudiantes@acl clase]$ grep estudiantes passwd
```

GNUNota: Recuerde para hacer estos ejercicio debe tener por lo menos una copia del archivo passwd.

122. Vamos a crear un archivo que contendrá la misma palabra pero en mayúscula y minúscula, buscaremos la palabra perro en el archivo.

```
[estudiantes@acl clase]$ cat > grepo.txt
Perro
Gato
perro
gato
Luna
luna
(Oprima CRLT+d)
```

```
[estudiantes@acl clase]$ grep perro grepo.txt
```

```
[estudiantes@acl clase]$ grep -i perro grepo.txt
```

GNUNota: Buscar en el man la definición de la opción "-i".

123. Vamos a buscar la palabra perro recursivamente.

```
[estudiantes@acl clase]$ grep -r "Perro" /home/estudiantes/fcld/bolges/libre/
```

GNUNota: Explicar al GNU/Instructor la opción "-r".

124. Realizar los siguientes ejercicios y explicar al GNU/Instructor la función de cada opciones en el comando "grep".

```
[estudiantes@acl clase]$ grep -w "Perro" grepo.txt
```

```
[estudiantes@acl clase]$ grep -n "estudiantes" passwd
```

```
[estudiantes@acl clase]$ grep -c "estudiantes" passwd
```

```
[estudiantes@acl clase]$ grep -ic "Perro" grepo.txt
```

125. Vamos utilizar el comando grep acompañado de la tubería.

```
[estudiantes@acl clase]$ dmesg | egrep '(s|h)d[a-z]'
```

```
[estudiantes@acl clase]$ cat /proc/cpuinfo | grep -i 'Model'
```

```
[estudiantes@acl clase]$ grep -i 'Model' /proc/cpuinfo
```

126. Vamos a cambiar de color la palabra encontrada por el comando grep en un archivo.

```
[estudiantes@acl clase]$ grep --color -i 'Model' /proc/cpuinfo
```

127. Vamos realizar ejercicios con el comando sed, borraremos 15 líneas de la copia de passwd.

```
[estudiantes@acl clase]$ sed '1, 15d' passwd |nl
```

128. Vamos a sustituir todos los ":" por "@" en el archivo passwd que tenemos en el directorio clase.

```
[estudiantes@acl clase]$ sed 's/:/@/g' passwd
```

129. Vamos a eliminar la primera línea del archivo grepo.txt.

```
[estudiantes@acl clase]$ sed -e "1d" grepo.txt
```

130. Borraremos la línea 2 y la 3 utilizando el comando sed.

```
[estudiantes@acl clase]$ sed -e "2,3d" grepo.txt
```

131. Vamos realizar ejercicios con el comando awk, seleccione las columnas 4 y 9 de la salida ls -l.

```
[estudiantes@acl clase]$ ls -l |awk '{print $4 $9}'
```

132. Mostramos el nombre completo del usuario y su login:

```
[estudiantes@acl clase]$ awk 'BEGIN {FS=":"}; {print $1,$NF | "sort"}' /etc/passwd
```

133. Visualicemos los usuarios que se encuentran conectados en el sistema, mediante el siguiente formato: *nº orden -- nombre usuario*, totalizando el nº de usuarios.

```
[estudiantes@acl clase]$ who -u|awk 'BEGIN { i=0 } { i+=1; print i,"-", $1 } END { print "Total usuarios " i }'
```

134. En este ejercicio eliminaremos las líneas repetidas y contaremos las veces que se encontró cada entrada, colocando esta cantidad como primera columna.

```
[estudiantes@acl clase]$ history|awk '{print $2}'|awk 'BEGIN {FS="|"} {print $1}'|sort|uniq -c
```

135. Realice los siguientes ejercicios y explique al GNU/Instructor el resultado de cada uno.

```
[estudiantes@acl clase]$ gawk '{print $3}' FS=":" /etc/passwd
```

```
[estudiantes@acl clase]$ awk -F":" '{ print $1 $3 }' /etc/passwd
```

```
[estudiantes@acl clase]$ awk -F":" '{ print "usuario: " $1 "\t\tuid:" $3 }' /etc/passwd
```

```
[estudiantes@acl clase]$ awk '$3 >= 500 {print $1 $5 $7 }' FS=":" /etc/passwd
```

```
[estudiantes@acl clase]$ awk 'BEGIN { print "Usuario UID Shell\n----- ---" } $3 >= 500 { print $1, $3, $7 | "sort -r"}' FS=":" /etc/passwd
```

```
[estudiantes@acl clase]$ awk 'BEGIN { print "Usuario UID Shell\n-----  
-----" } $3 >= 500 { printf "%12s %4d %10s\n", $1, $3, $7 | "sort -r"}'  
FS=":" /etc/passwd
```

```
[estudiantes@acl clase]$ awk 'BEGIN { print "Usuario UID Shell\n-----  
-----" } $3 >= 500 { printf "%-12s %4d %-10s\n", $1, $3, $7 | "sort -r"}'  
FS=":" /etc/passwd
```

```
[estudiantes@acl clase]$ who | awk '{ print $1,$5 }'
```

```
[estudiantes@acl clase]$ sed 3q /etc/passwd | awk -F: '{ print $1 }'
```

```
[estudiantes@acl clase]$ awk '{ printf "%4d %s\n", NR, $0 }'
```

```
[estudiantes@acl clase]$ sed 1q /etc/passwd | awk '{split ($0,a,":"); print a[1}]'
```

GNUNota: Cuando digite este ejemplo debe teclear la entrada.

136. Vamos a listar el contenido de un directorio y a la vez redireccionar la salida a un archivo.

```
[estudiantes@acl clase]$ ls | tee ls.txt
```

GNUNota: Repita el ejercicio pero con la opción -a del comando ls, explicar al GNU/Instructor que usted entendió.

137. Vamos a renombrar todos los archivos que se encuentran dentro del directorio clase. Estaremos utilizando el comando xargs.

```
[estudiantes@acl clase]$ ls | xargs -t -i mv {} {}.bak
```

GNUNota: La opción -i le comunica a xargs que reemplace {} con el nombre de cada elemento. La opción -t ordena a xargs que imprima el comando antes de ejecutarlo.

138. Otra operación muy útil cuando quiere abrir los archivos para edición utilizando vi:

```
[estudiantes@acl clase]$ file * | grep ASCII | cut -d":" -f1 | xargs vi
```

139. Vamos a sustituir los dos puntos (:) por espacio utilizando el comando tr.

```
[estudiantes@acl clase]$ tr ':' ' ' < /etc/passwd
```

140. Vamos a convertir todo lo que esta en minúscula a mayúscula con el comando tr.

```
[estudiantes@acl clase]$ tr '[a-z]' '[A-Z]' < /etc/passwd
```

141. Vamos a cambiar el case de las vocales con el comando tr.

```
[estudiantes@acl clase]$ echo murcielago | tr aeiou AEIOU
```

142. En el siguiente ejemplo borraremos las letras 'a' y 'e' de la palabra 'academia'.

```
[estudiantes@acl clase]$ echo academia | tr -d ae
```

143. Eliminaremos todos los espacios en blanco que tengamos repetidos en una frase.

```
[estudiantes@acl clase]$ echo 'una frase muy   tonta pero valdrá' | tr -d ' '
```

144. Eliminaremos todos los caracteres que tengamos repetidos en una frase.

```
[estudiantes@acl clase]$ echo 'una frase muy   toooonta peeero vaaaaldrá' | tr -s '[a-z]'
```

145. Eliminaremos todos espacios de más y luego lo vamos a sustituir en un solo espacio.

```
[estudiantes@acl clase]$ echo 'una frase muy   tonta pero valdrá' | tr -s ' '
```

146. Iniciaremos ejercicios con el mando split, vamos a dividir el archivo passwd en varios trozos por líneas de 10.

```
[estudiantes@acl clase]$ split --verbose -l10 passwd
```

GNUNota: Puede aplicar este ejemplo con archivos de MB o GB pero tendrá que cambiar las opciones.

147. Comprobar la cantidad de líneas de los archivos generados con el comando nl.

```
[estudiantes@acl clase]$ nl xaa
```

148. Realice el siguiente ejercicio y explique a su GNU/Instructor el resultado.

```
[estudiantes@acl clase]$ split -d -l10 passwd passwd
```

149. Vamos a concatenar la salida del ejercicio anterior.

```
[estudiantes@acl clase]$ cat passwd0* > clave.txt
```

GNUNota: Compruebe que el archivo clave.txt y passwd tengas las mismas cantidades de líneas.

150. Para realizar los ejercicios de mount debe usted tener una memoria usb a mano o algún otro device. Tenemos varios modos de ver si un dispositivos de almacenamiento esta conectado.

```
[root@acl ~]# fdisk -l
```

```
[root@acl ~]# cat /proc/partitions
```

GNUNota: ¿Sabe usted por qué sudo?, Explicarle al GNU/Instructor su idea sobre sudo y de qué otro modo puede realizar este ejercicio.

151. Una vez ya sabemos el nombre del dispositivo en /dev podemos montarlo, antes crearemos un directorio en mnt llamado memo donde lo montaremos.

```
[root@acl ~]# mkdir -p /mnt/memo
```

```
[root@acl ~]# mount /dev/sdb1 /mnt/memo
```

```
[root@acl ~]# df -h
```

GNUNota: Liste el contenido del directorio memo, debe salir el contenido que usted tenía en la memoria.

152. Hacer una imagen ISO de un CD o DVD también es fácil.

```
[root@acl ~]# cat /dev/cdrom > archivo.iso
```

153. Si deseamos montar ese archivo ISO como un CD o DVD (simular que esta físicamente introducido) lo hacemos como sigue.

```
[root@acl ~]# mount -o loop /home/estudiantes/archivo.iso /media/ISO
```

154. Vamos a ver un detalle de los filesystem montados actualmente.

```
[root@acl ~]# df -h
```

155. También podemos ver dónde están los dispositivos ejecutando el comando mount más sus privilegios.

```
[root@acl ~]# mount
```

156. Vamos a desmontar el dispositivo montado en /mnt/memo.

```
[root@acl ~]# umount /mnt/memo
```

GNUNota: Compruebe que no esta montado.

157. Mostrar el sistema de ficheros montado.

```
[estudiantes@acl ~]$ cat /proc/mounts
```

158. Si queremos ver información de nuestro CPU.

```
[estudiantes@acl ~]$ cat /proc/cpuinfo
```

159. Si queremos ver información de memoria en nuestro CPU.

```
[estudiantes@acl ~]$ cat /proc/meminfo
```

160. Si queremos saber el tamaño ocupado por un archivo o directorio.

```
[root@acl ~]# du -sch /var/
```

161. Vamos a mostrar información sobre la memoria RAM y Swap (total, libre y ocupada).

```
[estudiantes@acl ~]$ free -tm
```

162. Vamos a localizar la ubicación de un comando en específico.

```
[estudiantes@acl ~]$ which ls
```

163. Vamos a localizar archivos ejecutables, las fuentes y el manual de un comando.

```
[estudiantes@acl ~]$ whereis ls
```

164. Vamos a mostrar los dispositivos PCI.

```
[estudiantes@acl ~]$ lspci -tv
```

165. Vamos a mostrar los dispositivos USB.

```
[estudiantes@acl ~]$ lsusb
```

166. Vamos a mostrar el uid interno del usuario "nombre_usuario" y los grupos de seguridad a los que pertenece.

```
[estudiantes@acl ~]$ id
```

167. Mostrar la versión del kernel.

```
[estudiantes@acl ~]$ cat /proc/version
```

168. Mostrar la versión del kernel usado.

```
[estudiantes@acl ~]$ uname -r
```

169. Mostrar la información completa.

```
[estudiantes@acl ~]$ uname -a
```

GNUNota: Aplique help al comando uname y practique varias opciones.

170. Mostrar la arquitectura de la máquina.

```
[estudiantes@acl ~]$ uname -m
```

171. Mostrar la arquitectura de la máquina.

```
[estudiantes@acl ~]$ arch
```

172. Mostrar los componentes (hardware) del sistema.

```
[estudiantes@acl ~]$ dmidecode -q
```

173. Mostrar adaptadores de red y estadísticas.

```
[estudiantes@acl ~]$ cat /proc/net/dev
```

174. Vamos a sacar y entrar el cdrom o Dvd, lo que usted tenga:).

```
[estudiantes@acl ~]$ eject
```

GNUNota: Agregue la opción -t para entrar el dispositivo.

175. Vamos a dormir el shell durante 5 segundo.

```
[estudiantes@acl ~]$ sleep 5
```

176. Vamos a mostrar los módulos cargados en memoria.

```
[estudiantes@acl ~]$ lsmod
```

177. Vamos a loguearnos como super usuario.

```
[estudiantes@acl ~]$ su -
```

```
[root@acl ~]#
```

GNUNota: Para salir del ambiente root ejecute "exit" o "CTRL+D".

178. Mostrar las características de un disco duro SATA.

```
[root@acl ~]# hdparm -i /dev/sda
```


179. Vamos a cambiar el idioma de nuestro teclado desde el X.

```
[estudiantes@acl ~]$ setxkbmap us
```

GNUNota: Si desea en español cambie "us" por "es".

180. Vamos a cambiar el idioma de nuestro teclado desde consola.

```
[estudiantes@acl ~]$ loadkeys us
```

GNUNota: Si desea en español cambie "us" por "es".

181. Vamos a ejecutar repetidamente el comando entre comillas y muestra el resultado en pantalla.

```
[estudiantes@acl ~]$ watch "free"
```

182. Vamos a reiniciar nuestro sistema con el comando reboot.

```
[root@acl ~]# reboot
```

183. Vamos a reiniciar nuestro sistema con el comando init.

```
[root@acl ~]# init 6
```

184. Vamos a reiniciar nuestro sistema con el comando shutdown.

```
[root@acl ~]# shutdown -r now
```

185. Vamos a apagar nuestro sistema.

```
[root@acl ~]# poweroff
```

186. Vamos a apagar nuestro sistema con el comando halt.

```
[root@acl ~]# halt
```

187. Vamos a apagar nuestro sistema con el comando shutdown.

```
[root@acl ~]# shutdown -h now
```

188. Vamos a apagar nuestro sistema con el comando init.

```
[root@acl ~]# init 0
```

189. Vamos a apagar nuestro sistema con el comando telinit.

```
[root@acl ~]# telinit 0
```

190. Apagado planificado del sistema.

```
[root@acl ~]# shutdown -h 120 &
```

191. Cancelar un apagado planificado del sistema.

```
[root@acl ~]# shutdown -c
```

192. Crear un alias permanente.

```
[estudiantes@acl ~]$ echo "alias fcld=clear" >> ~/.bashrc
```

GNUNota: Verifique que el alias fue agregado y hacer un 'source' al .bashrc para que el alias sea efectivo en ese momento.

193. Desplegar el contenido de un directorio en una sola columna.

```
[estudiantes@acl ~]$ ls -1
```

194. El operador -R hace que el comando ls se ejecute repetitivamente, es decir, desplegar los subdirectorios y mostrar esos archivos también.

```
[estudiantes@acl ~]$ ls -R
```

195. ¿Qué sucede si quiere mostrar los archivos del más grande al más chico? Esto puede realizarse con el parámetro -S.

```
[estudiantes@acl ~]$ ls -lS
```

196. Ahora, queremos utilizar el comando ls -l y pasar la lista de arriba como parámetros, uno a la vez. El comando xargs le permitió hacer eso. La última parte, xargs ls -ltr, toma el output y ejecuta el comando ls -ltr sobre ellos, como si ejecutase:

```
[estudiantes@acl ~]$ file -Lz /etc/* |grep ASCII |cut -d":" -f1 |xargs ls -ltr
```

197. Aquí se muestra otro ejemplo, donde queremos contar la cantidad de líneas de los archivos ASCII de /etc:

```
[estudiantes@acl ~]$ file /etc/* | grep ASCII | cut -d":" -f1 | xargs wc -l
```

198. Varios ejemplos con el comando FOR:

```
[estudiantes@acl ~]$ for cantidad in dos tres cuatro cinco seis siete; do echo ${cantidad} elefantes se balanceaban sobre la tela de una araña; echo como veían que resistía fueron a llamar a otro elefante...; done
```

199. En este ejercicio primero se ejecuta ls, el cual dará el listado de todos los archivos de un directorio, y a todos esos archivos se les aplica un "file".

```
[estudiantes@acl clase]$ for archivo in `ls`; do file ${archivo}; done
```

200. Imprimamos varias variables al mismo tiempo:

```
[estudiantes@acl ~]$ for var in $PATH $MAIL $HOME; do echo $var; done
```

201. Este tipo de bucle for se caracteriza por contar. El rango se especifica mediante un comienzo (# 1) y el número final (# 5). El bucle for ejecuta una secuencia de comandos para cada miembro de una lista de elementos:

```
[estudiantes@acl ~]$ for VARIABLE in 1 2 3 4 5 .. N; do uptime; who; date; done
```

202. A veces puede ser necesario establecer un valor de paso (lo que permite contar con una de dos, o contar hacia atrás, por ejemplo).

```
[estudiantes@acl ~]$ for i in 1 2 3 4 5; do echo "Bienvenido Estudiante $i"; done
```

Otro ejemplo pero con llaves:)

```
[estudiantes@acl clase]$ for i in {1..5}; do echo "Bienvenido Estudiante $i"; done
```

```
[estudiantes@acl clase]$ for i in {0..10..2}; do echo "Bienvenido Estudiante $i"; done
```

203. Vamos excluir algunos de la lista:

```
[estudiantes@acl Saved]$ for i in $(seq 1 2 20); do echo "Bienvenido Estudiante $i"; done
```

204. Ejemplo de un bucle con el comando FOR, utilizando 3 variables.

```
[estudiantes@acl Saved]$ for (( c=1; c<=5; c++ )); do echo "Bienvenido Estudiante $c ..."; done
```

205. Parando un bucle infinito con una combinación de tecla.

```
[estudiantes@acl Saved]$ for (( ; ; )); do echo "Bucle Infinito codigolibre.org acl.edu.do [ hit CTRL+C to stop]"; done
```

206. A veces puede ser necesario establecer un valor de paso (lo que permite contar con una de dos, o contar hacia atrás, por ejemplo).

```
[estudiantes@acl Saved]$ x=1
```

```
[estudiantes@acl Saved]$ while [ $x -le 5 ]; do echo "Bienvenido Estudiante $x"; x=$(( $x + 1 )); done
```

207. Bucle infinito con un WHILE:

```
[estudiantes@acl Saved]$ while ;; do echo "Bucle Infinito estudiantes Vs GPL [ hit  
CTRL+C to stop]"; done
```

208. Como utilizar el comando IF:

```
[estudiantes@acl ~]$ if [ $pwd $HOME ] ; then cd "/home/estudiantes/Desktop/"; echo  
"Directorio cambiado..."; pwd; fi
```

GNUNota: Recuerde modificar la ruta.

209. Ruta dónde se encuentra el binario ls.

```
[estudiantes@acl ~]$ which ls
```

210. Nos devolverá todas las ocurrencias que encuentre del comando find.

```
[estudiantes@acl ~]$ which -a find
```

GNUNota: Para realizar la búsqueda which localiza los ficheros ejecutables mediante el PATH.

211. Para buscar un ejecutable y sus paginas man.

```
[estudiantes@acl ~]$ whereis find
```

212. Para buscar solo el binario.

```
[estudiantes@acl ~]$ whereis -b ls
```

213. Apaga el sistema de forma organizada desde una terminal texto.

```
[estudiantes@acl ~]$ [Ctrl]+[Alt]+[Del]
```

214. Termina el proceso actual si no está corriendo de fondo.

```
[estudiantes@acl ~]$ xclock
```

```
[estudiantes@acl ~]$ CTRL+c
```

215. Cierra la terminal actual.

```
[estudiantes@acl ~]$ CTRL+d
```

216. Información sobre las interrupciones en uso.

```
[estudiantes@acl ~]$ cat /proc/interrupts
```

217. Información sobre los sistemas de archivos que se pueden utilizar (compilados en el kernel).

```
[estudiantes@acl ~]$ cat /proc/filesystems
```

218. Para saber qué tarjeta gráfica tengo instalada.

```
[estudiantes@acl ~]$ lspci |grep -i vga
```

219. Para saber qué modelo de tarjeta de sonido tengo.

```
[estudiantes@acl ~]$ lspci |grep -i audio
```

220. Desplegar el contenido de un directorio con su número de inode.

```
[estudiantes@acl ~]$ ls -i /dev
```

221. Ayuda rápida ¿Qué es?.

```
[estudiantes@acl ~]$ whatis ls
```

222. Contiene los archivos más importantes para iniciar el sistema, incluyendo la configuración del cargador de arranque y el kernel compilado.

```
[estudiantes@acl ~]$ ls /boot
```

223. Se almacenan los archivos de configuración del sistema.

```
[estudiantes@acl ~]$ ls /etc
```

224. Aquí se coloca la configuración global del sistema (la configuración personal en ~/.bashrc y se leerá después de la configuración global) referente a los valores de las variables de entorno, los alias, la ruta de búsqueda. El valor de umask, las variables prompt, history, savehist, etc.

```
[estudiantes@acl ~]$ cat /etc/bashrc
```

225. Constituye un listado de los servidores DNS consultados por nuestro equipo para resolver direcciones.

```
[estudiantes@acl ~]$ cat /etc/resolv.conf
```

226. Muestra una lista con los shells disponibles.

```
[estudiantes@acl ~]$ cat /etc/shells
```

227. En este archivo se guarda información de los puntos de montaje del sistema.

```
[estudiantes@acl ~]$ cat /etc/fstab
```

228. Aquí se almacenan las transacciones.

```
[estudiantes@acl ~]$ ls /var/log
```

229. Atajos que funcionan en la shell Bash de GNU/Linux.

```
Ctrl + A      [estudiantes@acl Saved]$ for VARIABLE in 1 2 3 4 5 .. N; do uptime; who; date; done Manda el
cursor al principio de la línea
Ctrl + B      Funciona como la flecha izquierda, hace que el cursor se mueva hacia atrás.
Ctrl + C      Aborta la ejecución de el proceso actual.
Ctrl + D      Borra el carácter que este dentro del cursor. También cuando la shell esta en espera de ordenes
cierra la ventana y si hay varias pestañas abiertas cierra de una en una.
```

```
Ctrl + E      Manda el cursor al final de la línea
Ctrl + F      Mueve el cursor hacia adelante, funciona como la flecha derecha
Ctrl + H      Borra el carácter a la izquierda del cursor. Equivale a la tecla retroceso (backspace)
Ctrl + K      Elimina todo lo que este del cursor a su derecha
Ctrl + L      Limpia la pantalla. Es similar al comando clear.
Ctrl + N      Sigue el comando en el historial. Trabaja igual que la flecha abajo.
Ctrl + O      Equivale a enter
Ctrl + P      Trabaja igual que la flecha arriba. Muestra el comando anterior ejecutado.
Ctrl + R      Búsqueda en el archivo .bash_history. Busca un comando tecleado con anterioridad que empiece con
las letras que tecleamos.
```

```
Ctrl + S      Congela la terminal.
Ctrl + T      Cambia los dos caracteres anteriores a la posición del cursor.
Ctrl + U      Borra y guarda en el portapapeles lo que esta a la izquierda del cursor.
Ctrl + XX     Mueve el cursor a su posición anterior.
Ctrl + II     Muestra todos los comandos posibles
Ctrl + Y      Pega la palabra que este guardada en el portapapeles cuando se haya usado los Ctrl + U y Ctrl + W
Ctrl + W      Borra y guarda en el portapapeles la palabra a la izquierda del cursor.
Ctrl + Z      Suspende/detiene el comando. Manda el proceso actual al background (segundo plano), el proceso
puede ser regresado a primer plano con los comandos: jobs y fb.
```

```
Ctrl + Re Pág (Page Up)      Cambia a la pestaña izquierda
Ctrl + Av Pág (Page Down)    Cambia a la pestaña derecha
Ctrl + Flecha derecha        Mueve el cursor una palabra a la derecha.
Ctrl + Flecha izquierda      Mueve el cursor una palabra a la izquierda.
```

***** Alt *****

```
Alt + *      (Si es el carácter arriba en la tecla hay que presionar además shift) Inserta todas las posibles
entradas que pueden completar la palabra.
Alt + <      Mueve a la primera línea en el historial.
```

```
Alt + =      Mueve a la ultima línea en el historial.
Alt + ?      (Si es el carácter arriba en la tecla hay que presionar además shift) Muestra la lista completa
de sugerencias de posibilidades.
```

```
Alt + backspace Borra hacia atrás del cursor
Alt + .          Muestra la ultima parte del comando previo utilizado
Alt + C          Hace que la primera letra de una palabra se convierta a mayúscula y el resto a minúsculas. Cuando
esta en medio de una palabra convierte a mayúscula la letra donde esta el cursor y deja en minúsculas las letras
del cursor hacia la derecha.
Alt + D          Borra una palabra. Me funciona cuando estaba al principio de la misma.
Alt + L          Convierte una palabra a minúsculas
Alt + N          Busca hacia delante en el historial, non-incremental
Alt + P          Busca hacia atrás, non-incremental
Alt + U          Hace mayúscula una palabra
Alt + 1-9        Nos mueve de pestaña en pestaña de acuerdo a la posición que ocupa y el número que hayamos
presionado.
```

***** Ctrl + Shift *****

```
Ctrl + Shift + C Copiar
Ctrl + Shift + N Abre terminal
Ctrl + Shift + Q Cierra ventana
Ctrl + Shift + T Abre pestaña
Ctrl + Shift + W Cierra pestaña
Ctrl + Shift + V Pegar
Ctrl + Shift + Re Pág (Page Up)      Mueve la pestaña a la izquierda
Ctrl + Shift + Av Pág (Page Down)    Mueve la pestaña a la derecha
```

***** Alt + Shift *****

```
Alt + Shift + B Mueve el cursor 1 palabra hacia atrás en la línea en la que esté situado.
Alt + Shift + F Mueve el cursor 1 palabra hacia adelante en la línea en la que esté situado.
Alt + Shift + T Invierte el lugar de dos palabras
```

***** Tabulador (Primero el símbolo y luego presionar dos veces tab) *****

```
2T          Muestra todos los comandos disponibles
/ + 2T      Muestra la estructura completa de directorios raíz incluidos los ocultos.
[Directorio]/ + 2T    Igual que /+2T. Muestra la estructura completa del directorio, incluido los ocultos.
[Dentro de un directorio] + 2T  Muestra todos los subdirectorios incluidos los ocultos.
* + 2T      Muestra los subdirectorios excluyendo a los ocultos.
~ + 2T      Muestra todos los usuarios (users) presentes en el sistema desde "/etc/passwd"

$ + 2T      Todas las variables del sistema.
@ + 2T      Muestra las entradas desde "/etc/hosts"
= + 2T      La salida es igual a ls en el directorio.
```

***** Otros atajos *****

```
Flecha arriba  Muestra el comando anterior ejecutado
Flecha abajo   Muestra el siguiente comando ejecutado
Flecha izquierda Mueve el cursor hacia la izquierda
Flecha derecha  Mueve el cursor hacia la derecha
Inicio         Mueve el cursor al principio de la línea
Fin            Mueve el cursor al final de la línea
Esc y luego la letra D  Borra de la posición actual al final de la palabra
Esc y luego la letra P  Busca en el historial de comandos. Similar a Ctrl+R
Esc y luego la letra T  Invierte el orden de dos palabras contiguas
Esc y luego la letra U  Cambia a mayúsculas la palabra adelante del cursor
```