

# Lo elemental de la terminal de linea de comandos de GNU/Linux para usuarios de la UBMI

## Antes de comenzar

**Esta no es una guía exhaustiva de uso de línea de comandos y hay muchas omisiones intencionales en cuanto a las opciones y uso de los programas, pero hemos recopilado aquí lo que nos parece es un buen punto de partida para hacer mas agradable el uso de la terminal de GNU/Linux**

1. Entre [ ] se encuentran los argumentos que deben ser sustituidos por el usuario.
2. Los ejemplos en este documento van acompañados del símbolo \$ indicando que es texto en una terminal, pero este no es necesario copiarlo o escribirlo para reproducir los ejemplos.
3. No se muestran todas las opciones, sólo las que creemos son las más utilizadas.
4. En general, las opciones pueden combinarse por ejemplo: un programa con opciones -l -h -t -r puede ejecutarse cómo:

```
$ programa -l - h -t -r
```

ó como:

```
$ programa -lhtr
```

la excepción a esta regla ocurre cuando alguna de las opciones requiere un argumento adicional por el usuario, en el ejemplo anterior supongamos una opción -n que recibe un nombre, entonces se ejecutaría:

```
$ programa -lhtr -n [nombre]
```

5. Típicamente los programas contienen información que facilita su uso, en general si se utilizan las opciones -h ó --help se puede obtener información, adicionalmente se puede utilizar los comandos man

```
$ man [nombre del programa del que se quiere saber su uso]
```

ó tldr

```
$ tldr [programa]
```

6. Por último, en la terminal de linea de comandos de GNU/Linux suelen haber programas con opciones de una letra tal como: -h y opciones de mas de una letra cómo --help estas últimas las distinguimos por que tienen dos guiones en vez de uno, y a diferencia de las opciones de una letra estas no pueden ‘aplilarse’ como en el

punto 4 de esta lista, pero pueden usarse juntas en combinación con otras opciones de una o mas letras.

## Contenido

- Uso de comodines ó *wildcards*
- Teclas amigas
- Uso de comandos útiles
- Navegación
- Manipulación de archivos
- mv
- rm
- touch
- mkdir
- cat
- less
- grep
- Conexiones remotas y red
- exit
- ssh
- sshfs
- scp
- wget
- nohup
- Estado del equipo
- top
- htop
- ps
- du
- df
- free
- cockpit

## Comodines (*wildcards*)

En GNU/Linux podemos usar caracteres comodín para realizar operaciones sobre archivos o líneas de texto con parte del nombre en común, existen muchos tipos de comodín pero los que se utilizan con mas frecuencia son:

Comodín	Uso

Comodín	Uso
*	Representa cualquier carácter, cualquier número de veces, por ejemplo <code>\$ rm *txt</code> borraría todos los archivos cuyo nombre termine en “txt”, <code>\$ ls *fotos*</code> enumeraría los archivos y directorios que contengan la palabra “fotos”
?	Representa cualquier carácter <b>una</b> sola vez, en una carpeta con archivos llamados: “perros”, “gatos” “ratos” “platos” la operación <code>\$ ls ?atos</code> nos devolvería: <code>gatos ratos</code> pero no “platos”
~	Un atajo que hace referencia a la carpeta “home” del usuario que la escribe, permite sustituir por ejemplo: <code>\$ cd /home/miusuario/Documentos</code> por <code>\$ cd ~/Documentos</code>
..	Otro atajo que hacer referencia a la carpeta anterior en el arbol de directorios; <code>\$ cd ..</code> evita que tengamos que escribir la dirección completa si solo queremos regresar un nivel y puede usarse en sucesión para regresar varios niveles: <code>\$ cd ../../..</code>

## Teclas amigas

Existen varias teclas y combinaciones que nos facilitan la vida en la línea de comandos dentro de las más utiles están:

*Nota: La tecla Ctrl también suele ser representada como ^ en algunas documentaciones y archivos de ayuda*

Tecla(s)	Función
Tab	Habilita el autocompletado al escribir comandos y paths (casi siempre)
Ctrl+ c	Interrumpe la ejecucion de un programa

Tecla(s)	Función
Ctrl+ z	Manda a segundo plano la ejecución del programa, no debe confundirse con terminar el programa, es el equivalente de minimizar una pantalla
fg	Escribir \$ fg en la línea de comando revierte el efecto de Ctrl+z
Ctrl+a Ctrl+e	Nos llevan al inicio y al final de una línea de texto respectivamente
Teclas arriba y abajo	Permiten navegar entre los últimos comandos utilizados
Ctrl+r	Nos permite buscar entre los comandos usados recientemente tecleando una fracción del mismo
>	Permite escribir en un archivo de texto lo que de otra forma se representaría en pantalla como resultado de la ejecución de un programa, por ejemplo \$ ls > archivos.txt creará un archivo dónde escribirá el resultado de la operación \$ ls utilizar de nuevo \$ otro_comando > archivos.txt sobrecribirá el archivo, mientras que: \$ otro_comando >> archivos.txt agregará el resultado del programa “otro_comando” al final del archivo “archivos.txt”
“ ”	El caracter pipe nos permite dar el resultado de un programa a otro programa por ejemplo: \$ ls -l   wc -l primero ejecutará “ls -l” para listar los archivos de una carpeta y el texto resultante en vez de ponerlo en pantalla se lo dará al programa “wc -l” que sirve para contar líneas de texto y este imprimirá en pantalla un número que representará el numero de renglones de la primera operación, que en este caso se traduce en el número de archivos de la carpeta actual.

## Comandos útiles

Existen muchos comandos en GNU/Linux, aquí están los mas útiles para comenzar con lo que (creemos) son las mejores opciones para su uso.

## Navegación

Comando	Descripción
cd [/path/objetivo]	Cambia de directorio a la dirección especificada
cd ..	Nos devuelve un nivel en el arbol de directorios
cd -	Nos regresa a la carpeta anterior en la que estábamos
ls	Enlista los contenidos de la carpeta actual
ls -l	ls en forma de lista
ls -lh	ls en forma de lista detallada y en formato mas legible
ls -a	ls enlistando también archivos ocultos
ls *	ls recursivo sobre los directorios de la carpeta actual
tree	Funciona como un ls * pero con la recursividad se extiende hasta que no haya mas archivos que enlistar
tree -fi	Cambia el formato de “arbol” default por uno con el path completo de los archivos enlistados
tree -fi   grep [parte del nombre de un archivo]	Permite buscar un archivo con parte del nombre y regresa el path completo
find [path/objetivo] -name [parte del nombre de un archivo]	Igual que el anterior
pwd	Nos dice en que directorio nos encontramos actualmente con un path absoluto
history	enlista los últimos n comandos que hemos utilizado (n varía en cada equipo)
history   grep [parte de un comando]	Nos permite buscar entre los últimos comandos usando una fracción del deseado, es mejor usar simplemente Ctrl+r

Comando	Descripción
ls	Muestra los contenidos de la carpeta actual
ls -l	Muestra los contenidos en forma de lista
ls -lh	Muestra los contenidos en forma de lista y con un formato legible para humanos
ls [/path/a/un/directorio/	Muestra los contenidos de un directorio objetivo

## touch

Crea archivos vacíos. **Nota:** aunque se les asigne una extensión esta será ignorada, mientras esté vacío.

Comando	Descripción
---------	-------------