

La Terminal – Comandos Linux

Curso de platzi

Navegar por la terminal , creación de archivos y directorios

- La carpeta con el símbolo “/” es la raíz, ahí es donde comienza todo el sistema de ficheros.
- El comando **pwd**, significa Print Working Directory y te muestra el directorio en el que te encuentras.
- **cd** significa Change Directory.
- La virgulilla (~) indica la carpeta del usuario en el home. Si quieres volver al home estando en x directorio, puedes escribir **cd ~**
- El doble punto (..) te envía a la carpeta que está atrás. El punto (.) indica la carpeta actual, por ejemplo si estas en tu home puedes escribir directamente **cd ./Documents/carpeta**
- **Slash(/)** el atajo slash te lleva a la raíz donde están todas las carpetas del sistema operativo. Ejemplo: **cd /**
- Si ejecutamos el comando **ls** veremos las carpetas y archivos que hay dentro.
- El comando **ls -lh** El comando list se le puede pasar parametros. En este caso Long y Human Readable.
 - **ls -a** muestra todos los archivos incluyendo los ocultos.
 - **ls -lSh** te ordena los archivos por tamaño (por el más pesado) y te muestra el tamaño en Human Readable.
- El comando **file** te da información acerca del tipo de elemento que hayas seleccionado: **file dev**, **file lib**.
- **clear (ctrl + l)** limpia la pantalla.
- **mkdir Make Directory** Crea carpetas. Ejemplo: **mkdir DirectorioInteresante SecretosDeEstado**.
- **touch** crea archivos. Ejemplo: **touch Secreto1 Secreto2 touch.js SecretoSecretario**
- **cp Copy** Copia archivos. **cp [nombreDelArchivoParaCopiar] [nombreParaLaCopia]**
- **mv Move** mueve archivos. **mv [archivoParaMover] [destinoDelArchivo/nombreNuevo]**
- **rm Remove** elimina archivos o carpetas. Tiene varias opciones:
 - **-i** (de interactive) te pregunta si estás seguro de eliminar el archivo.
 - **-r** (de recursive) elimina todo lo que esté dentro de una carpeta
 - **-f** (de force) fuerza a borrar todo.
- **tree -L 2** te muestra los el árbol de las carpetas y archivos a dos niveles.

Explorando el contenido de nuestros archivos

- **head** Muestra las primeras 10 líneas. Para ver las primeras 20 líneas: **head [archivo] -n 20**
- **tail** Muestra las últimas 10 líneas. Para ver las últimas 20 líneas: **tail [archivo] -n 20**
- **less** Muestra todo el contenido de un archivo dentro de la consola. Una vez viendo el contenido se puede buscar alguna palabra clave con */palabra-clave* y presionando enter. Para salir presionamos **q**
- **xdg-open** Abre un programa para inspeccionar ese archivo.
- **nautilus** Abre en la interfaz de ventanas la carpeta que selecciones.
- **ctrl + c** para matar cualquier proceso en la terminal.

Un comando en general

Un **comando** es un mensaje enviado al ordenador que provoca una respuesta en este sistema y se comporta como una orden, pues informa al dispositivo informático que debe ejecutar una acción según la indicación que pueda enviarse. Un comando puede ser:

- Un programa ejecutable.
- Un comando de utilidad de la shell.
- Una función de shell.
- Un alias.

Ejemplos de comandos básicos de la terminal:

- **type** <comando>: Nos permite conocer que tipo de comando es.
- **alias** l="*<secuencia de comandos>*": Nos permite crear comandos. Son temporales, se borran al cerrar la terminal. Hincapie en que la secuencia de comandos va entre comillas.
- **help** <comando>: Nos permite consultar un poco de documentación de un comando.
- **man** <comando>: De manual, nos permite conocer mucha más información de un comando.
- **info** <comando>: Similar al anterior, pero un poco resumido y con otro formato.
- **whatis** <comando>: Describe un comando en una sola línea ☺. No funciona con todos.

Nota: Cuando un parametro de un comando en la terminal es con palabras completas, se le pone doble guion --. Ejemplo: `ls --color=auto`

Wildcards

Las wildcards o comodines son una serie de caracteres especiales que nos permiten encontrar patrones o realizar búsquedas más avanzadas. Es aplicable para archivos y directorios.

Ademas de **ls** los wildcards tambien pueden usarse con cualquier comando que realice la manipulación de archivos como **mv**, **cp** y **rm**.

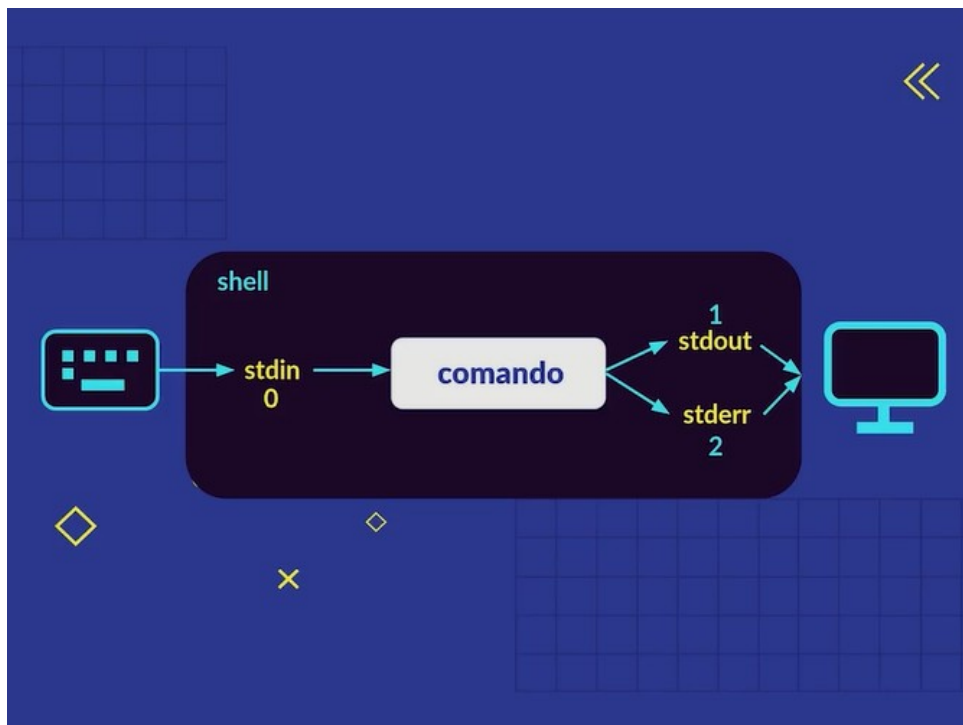
Los tipos de wilcards son:

- **Buscar todo (*)**. El asterisco te ayuda a buscar toda la información dentro de una carpeta. Por ejemplo: **ls -l *.png** y te muestra todos los archivos cuya extensión es .png. Otro ejemplo es: **ls -l fotoDe*** y así buscarías todos los archivos que comiencen por “fotosDe”.
- **Buscar por cantidad de caracteres (?)** Ejemplo: **ls -l foto?.png** con eso buscarías “foto1.png”, “foto2.png”, “foto3.png”. Otro ejemplo **ls -l foto???.png** “foto11.jpg”.
- **Buscar por caracteres específicos ([])**. Para utilizarlos tienes que colocar dentro de los corchetes los caracteres que quieres buscar. Por ejemplo: si quisieras buscar los archivos que comiencen por las letras “c” o “i”, entonces escribes: **ls -l [ci]***. Otro ejemplo sería que si quieres buscar por rango de números también tienes que usar esta wildcard **ls -l foto[2-6]***

- **ls [[:upper:]]*** Encuentra todos los archivos/directorios que empiecen por una mayúscula.
- **ls -d [[:upper:]]*** Encuentra todos los **directorios** que empiecen por una mayúscula.
- **ls [[:lower:]]*** Encuentra todos los archivos/directorios que empiecen por una minúscula.

Redirecciones: cómo funciona la shell

- A las entradas típicamente se les suele llamar **Standard Input** y a las salidas **Standard Output**, además se les suele abreviar como **stdin** y **stdout** respectivamente. También tenemos el **Standard Error** (**stderr**).
- Los **file descriptors** son números que identifican un recurso. Funciona asociando un número con una acción, archivo o programa, en el caso de la shell tenemos 3 file descriptors:



- Cómo usar el operador de **redirección (>)**
 - Si quieres que el Standard Output no vaya a la consola sino hacia un archivo, entonces puedes usar el operador **>** seguido del nombre del archivo en el que quieres guardar la salida. Por ejemplo: **ls -l > output.txt** Y si el archivo no existe se crea de manera automática.
- Cómo **concatenar (>>)**
 - Para concatenar el contenido de un archivo ya existente con el de la salida (stdout) se ejecuta: **ls -l >> file.txt**
- **Operador 2>:** Redirecciona el file descriptor 2 (En este caso Standard Error). Ejemplo: podemos especificar que no importa lo que pase si me da un **Standard Output** o un **Standard Error**, igual tiene que guardar la salida en un archivo. Esto lo hacemos así: **ls -l > output.txt 2>&1** Otro ejemplo sería: **ls rtrytyt 2> error.txt**
- **Operador 2>&1:** Redirecciona el file descriptor 2 y 1

Redirecciones: pipe operator

- **Pipe operator** es un operador que permite tomar la salida de un comando y pasarla como entrada de otro comando. Aprendamos más comandos que te van a ayudar.
- **Unir cadenas de texto (cat):** Si queremos crear una lista de los archivos de varias carpetas, podemos usar cat para concatenar la salida de varios de ellos.
 - Ejemplo: **cat images.txt secretosDeEstado.txt**
- **Crear un archivo con base en una salida (tee):** Si queremos guardar la lista creada anteriormente, podemos pasar esa salida por medio de un pipe operator al comando tee, el cual creará un archivo con esa salida. Con el comando tee, ofrece la posibilidad no solo de dar salida a la entrada estándar de forma normal, sino también de escribirla en uno o más archivos.
 - Ejemplo: **cat images.txt secretosDeEstado.txt | tee archivos.txt**. Si el archivo no existe se crea de manera automática.
 - También se puede escribir a varios archivos a la vez. Por ejemplo: **du -h | tee disk_usage1.txt disk_usage2.txt disk_usage3.txt**
 - También puedes utilizar el comando tee junto con sudo en Linux. Esto es ventajoso o incluso necesario si se ha escrito en un archivo que tiene derechos de root. Ejemplo: **echo "Ejemplo" | sudo tee -a documento_root.txt**
- **El comando sort (ordenar):** es un comando para ordenar líneas de archivos de texto. Permite ordenar alfabéticamente, en orden inverso, por número, por mes y también puede eliminar duplicados.
 - Ejemplo: **ls | sort | tee archivosHome.txt** Aquí lo que estamos haciendo es:
 1. Listar los archivos.
 2. Organizar los archivos.
 3. Crear un archivo llamado archivosHome.txt, con las salidas anteriores.
 - El comando sort nos permite ordenar el archivo alfabéticamente: **sort archivo.txt**
 - Se puede ordenar de manera inversa con la opción -r: **sort -r archivo.txt**
- Ejemplo adicional (intentar en la terminal): **ls -lh | tee archivo.txt | less**
- **Un experimento:**
 - **sudo apt install cowsay**
 - **sudo apt install lolcat**
 - **cowsay "Me gusta Javascript!" | lolcat**

Encadenando comandos: operadores de control

Los operadores de control son símbolos reservados por la terminal que nos permiten encadenar comandos.

- **Comandos en la misma línea (;)** Ejecuta de forma síncrona los comandos especificados. Solo necesitas escribir los comandos que quieres ejecutar separados por ; y luego presionar enter. Ejemplo: **mkdir ProyectosSecretos; ls; date**
 - El comando **date** imprime por consola la fecha actual
- **Comandos asíncronos (&)** Ejecuta de forma asíncrona los comandos especificados. Para llevar a cabo varios comandos, al mismo tiempo, usamos el operador & entre cada comando que queremos ejecutar. Ejemplo: **date & echo "Hola" & cal**
 - El comando **cal** imprime un pequeño calendario de la fecha actual y el comando **echo** imprime el texto que le pases

Comandos con condicionales

Podemos ejecutar comandos dependiendo de condiciones.

- **Condición and (&&)** Ejecuta el comando si el anterior se ejecutó correctamente. Si escribimos varios comandos separados por el operador && estamos indicando que para que estos se ejecuten, el comando anterior tuvo que ejecutarse correctamente.
 - Ejemplo: **cd NoExisto && mkdir Comida**. En este ejemplo, intentamos cambiar al directorio “NoExisto” pero ese directorio no existe, así que el comando falla y no se ejecuta el siguiente.
 - En caso de que el primer comando se haya ejecutado correctamente pasará al siguiente, y después verá si ese se puede ejecutar. Ejemplo: **cd ProyectosSecretos/ && touch ProyectoExplosivo.txt && ls**. Como en este caso todos los comandos se ejecutaron correctamente sucede esto:
 1. Cambia al directorio “ProyectosSecretos”.
 2. Crea el archivo “ProyectoExplosivo.txt”.
 3. Lista el contenido del directorio con ls.
- **Condicional or (||)** Ejecuta el comando si el anterior o la combinación de los anteriores resultaron en verdadero. Al condicional or no le importa si el comando anterior se ejecutó o no, simplemente va probando todos los comandos a ver cuál se ejecuta. Y al primero en ejecutarse termina la ejecución del condicional (||).
 - Por un momento vamos a suponer que no sabemos muy bien cuál es el comando para cambiar de directorio si es cd o cambia-carpeta, entonces para evitar un error escribimos. Ejemplo: **cd ProyectosSecretos/ || cambia-carpeta ProyectosSecretos/**. Aquí vemos que alguno de los dos comandos está mal, pero igual cambió la carpeta porque uno de esos funcionó.
- **Combinando operadores de control**. Siguiendo con el ejemplo anterior, vamos a cambiar de directorio. Si se logra cambiar de directorio creamos una carpeta adentro.
 - Ejemplo: **cd ProyectosSecretos/ || cambia-carpeta ProyectosSecretos/ && mkdir ProyectoIncreible** Esto funcionó porque el operador or devolvió verdadero al tener un comando que funcionara, por lo tanto, el operador and lo interpretó como un comando que funcionó correctamente.

Cómo se manejan los permisos

Los permisos son las capacidades que tiene cada usuario dentro del sistema operativo. No todos los usuarios pueden hacer todas las acciones sobre ciertos archivos y carpetas.

- Cuando listamos archivos utilizando el comando **ls -l** la primera columna que nos aparece es la de permisos.
- **Tipos de archivos** El primer caracter puede ser uno de estos 3: (ver tabla de abajo)

Atributo	Tipo de archivo
-	Es un archivo normal, como un documento de texto, una foto, un video, etc.
d	Por directory es un directorio
l	Es un enlace simbólico. Es algo que veremos en próximas clases
b	Bloque especial, son archivos que manejan información para el sistema, como la información de un disco duro

```
miguelangel@DESKTOP-3R804MK: /mnt/c/users/migue/Documents$
miguelangel@DESKTOP-3R804MK: /mnt/c/users/migue/Documents$ ls -l
total 8652
drwxrwxrwx 1 miguelangel miguelangel 4096 Jan 23 17:00 .
drwxrwxrwx 1 miguelangel miguelangel 4096 Apr 29 10:45 ..
drwxrwxrwx 1 miguelangel miguelangel 4096 Feb 11 19:06 .Chantasia
-rwxrwxrwx 1 miguelangel miguelangel 1228429 Apr 8 15:10 'Curso Power BI - Parte 1.pbix'
-rwxrwxrwx 1 miguelangel miguelangel 770134 Apr 8 20:43 'Curso Power BI - Parte 2.pbix'
drwxrwxrwx 1 miguelangel miguelangel 4096 Apr 18 18:26 'Curso de Informática'
drwxrwxrwx 1 miguelangel miguelangel 4096 Apr 28 14:28 'Cursos'
-rwxrwxrwx 1 miguelangel miguelangel 489213 Apr 8 17:47 'Ejercicio Resumen General.pbix'
drwxrwxrwx 1 miguelangel miguelangel 4096 Jan 23 15:18 'GIA San Andreas User Files'
-rwxrwxrwx 1 miguelangel miguelangel 54948 Dec 17 19:09 'Grafico-porcentaje-por-clic.png'
drwxrwxrwx 1 miguelangel miguelangel 4096 Mar 27 11:14 'Imágenes'
drwxrwxrwx 1 miguelangel miguelangel 24 Mar 24 18:51 'Mi música' -> /mnt/c/Users/migue/Music
lrwxrwxrwx 1 miguelangel miguelangel 27 Mar 24 18:51 'Mis imágenes' -> /mnt/c/Users/migue/Pictures
lrwxrwxrwx 1 miguelangel miguelangel 25 Mar 24 18:51 'Mis videos' -> /mnt/c/Users/migue/Videos
drwxrwxrwx 1 miguelangel miguelangel 4096 Mar 19 16:25 'Muestrales'
drwxrwxrwx 1 miguelangel miguelangel 4096 Apr 23 10:04 'My Games'
-rwxrwxrwx 1 miguelangel miguelangel 116 Nov 4 01:16 'Nuevo documento de texto.txt'
drwxrwxrwx 1 miguelangel miguelangel 4096 Mar 15 11:13 'Plantillas personalizadas de Office'
drwxrwxrwx 1 miguelangel miguelangel 4096 Apr 7 14:21 'Programas'
drwxrwxrwx 1 miguelangel miguelangel 4096 Jan 29 14:04 'Proyectos'
-rwxrwxrwx 1 miguelangel miguelangel 3609862 Mar 19 18:16 'Referencias Personajes y Otras Weas.pptm'
-rwxrwxrwx 1 miguelangel miguelangel 35906 Oct 19 2019 'Tarjeta 19-10-26.PDF'
drwxrwxrwx 1 miguelangel miguelangel 4096 Mar 28 19:36 'Universidades'
-rwxrwxrwx 1 miguelangel miguelangel 649200 Nov 13 2019 carrillo_como_llora_una_estrella.pdf
-rwxrwxrwx 1 miguelangel miguelangel 402 Jan 13 22:50 desktop.ini
```

Permisos de usuario

Los siguientes caracteres se leen de 3 en 3, por cada uno de los tipos de usuario.

- **Owner** El dueño del archivo, si no se ha cambiado, es quien lo creo y tiene mayor jerarquía sobre los otros 3. Le corresponden los primeros 3 caracteres de los permisos.
- **Group** Se puede crear grupos de usuarios para darle a todos o varios los mismos permisos. A estos usuarios le corresponden el cuarto, quinto y sexto caracter de los permisos de usuarios y tienen mayor jerarquía que el último.

- **World** También llamado “otros”, es cualquier otro usuario que no pertenezca a un grupo de usuario y tampoco sea el dueño, este tiene la menor jerarquía.

Tipos de permisos

Los permisos se escriben en este orden **rwX**. Para indicar que el permiso no está disponible, se escribe un guion.

Símbolo	Significado	Permiso
r	readable	Significa que puede leer su contenido
w	writable	El usuario puede editar el contenido del archivo, también el nombre y los permisos
x	executable	El usuario puede ejecutarlo en caso de que sea un programa

Ahora que sabes todo esto vamos con un ejercicio. Observa el siguiente grupo de permisos: **drwxr-xr-x**

Recuerda que el primer caracter es el tipo y los siguientes se cuentan de 3 en 3 representando cada usuario:

drwxr-xr-x

d	rwX	r-X	r-X
Esto es un directorio	owner	group	world
	El dueño puede leer, escribir y ejecutar	El grupo puede leer y ejecutar	Los demás pueden leer y ejecutar

Representando permisos de forma octal

drwxrwxrwx

d = Directory

r = Read

w = Write

x = Execute

chmod 777

rwX | rwX | rwX
Owner | Group | Others

7	rwX	111
6	rw-	110
5	r-X	101
4	r--	100
3	-wX	011
2	-w-	010
1	--X	001
0	---	000

Platzi

- Modo simbólico

:: o

Símbolo	Significado
u	Solo para el usuario.
g	Solo para el grupo.
o	Solo para otros (es el <i>world</i>) .
a	Aplica para todos.

Platzi

Modificando permisos en la terminal

Con el comando `chmod` podemos cambiar los permisos de los archivos de dos formas, una es usando los símbolos (rwx) y otra es con el sistema octal.

Para esto, hay que escribir después del comando **chmod** el símbolo del usuario, luego el operador y por último el permiso que quieres agregar o quitar.

chmod [simboloDelUsuario][operador][permiso] [archivoParaCambiarSusPermisos]

owner	group	others
u (de user)	g	o
Operador	Función	
+	Añade un permiso	
-	Quita un permiso	
=	Asigna un permiso	

Observa los permisos del siguiente archivo:

```
miguelangel@DESKTOP-3R804MK:~/ProyectosSecretos$ ls -l
total 0
-rw-r--r-- 1 miguelangel miguelangel 0 May  1 13:51 ProyectoExplosivo.txt
```

- Supongamos que queremos añadirle permiso de escritura al grupo, entonces tenemos que escribir lo siguiente: **chmod g+w ProyectoExplosivo.txt**, y el resultado sería lo siguiente.

```
miguelangel@DESKTOP-3R804MK:~/ProyectosSecretos$ chmod g+w ProyectoExplosivo.txt
miguelangel@DESKTOP-3R804MK:~/ProyectosSecretos$ ls -l
total 0
-rw-rw-r-- 1 miguelangel miguelangel 0 May  1 13:51 ProyectoExplosivo.txt
miguelangel@DESKTOP-3R804MK:~/ProyectosSecretos$
```

- Puedes cambiar varios permisos de varios usuarios al mismo tiempo, por ejemplo, si quisieras agregar el permiso de escritura y ejecución al grupo y a otros, sería así:
chmod go+wx [archivo]
- Y si quieres permisos diferentes para cada usuario, solo sepáralos por comas:
chmod u+r,g=w [archivo] En este comando se le añadió el permiso de lectura al dueño y de escritura al grupo. No agregues espacio en las comas o provocarás un error.

- También puedes cambiar los permisos usando su forma octal, por ejemplo el conjunto de permisos **rwxr-xr-x** en su forma octal es **755**. Ejemplo: **chmod 755 [archivo]**.

Cómo gestionar usuarios (whoami | su)

- El comando **whoami**, literalmente “¿Quién soy yo?”, te muestra cual es el usuario que se está ejecutando.
- Para cambiar de usuario se usa el comando **su Switch User**, seguido del usuario al que quieres cambiar, en este caso vamos a cambiar al superusuario root. Si el sistema operativo es **ubuntu** entonces sería de la siguiente manera: **sudo su**. En cualquier caso usa los privilegios del root con cuidado ya que, puedes eliminar cosas que no deberías eliminar y puede hacer mucho desastre.
- Si quieres ver el home del root ejecuta esto: **cd ~; pwd**
- Si creamos un archivo con root: **touch ArchivoRoot.txt; ls -l**

```
root@DESKTOP-3R804MK:/home/miguelangel# touch ArchivoRoot.txt; ls -l
total 40
-rw-r--r-- 1 root      root      0 May  1 21:07 ArchivoRoot.txt
```

Ahora si intentamos borrar ese archivo con otro usuario que no sea root no vamos a poder porque no tiene los permisos para hacerlo.

¿Qué hacer en caso de olvidar una contraseña?

Si estás usando Windows Subsystem for Linux (wsl) y se te olvidó la contraseña del root. Sigue estos pasos:

1. Abre el cmd de windows y ejecuta este comando **wsl --user root**. Esto hará que se inicie en la terminal wsl con el usuario root.
2. Luego ejecuta el comando **passwd root** el cual te permitirá cambiar la contraseña del usuario root.

Cambiar el propietario (chown)

Puede que no te quieras hacer responsable de tus archivos, así que se los quieres dejar a alguien más. Para eso usa el comando **chown** Change Owner. La sintaxis es muy simple:

chown [usuarioAlQuePertenece] [archivo]

Variables de entorno en Linux

Las variables de entorno son útiles cuando necesitamos que cierta información prevalezca para poder trabajar más rápido o necesitamos guardar información para no tener que recordarla constantemente.

En linux hay varias variables de entorno que ya están preestablecidas, para verlas desde la consola es tan simple como usar el comando seguido de la variable de entorno. Todas las variables de entorno se mandan a llamar con un signo de peso por delante, de lo contrario se interpretará como un comando.

Variable	Contenido
HOME	Indica el home del usuario
PATH	Indica las direcciones de donde están los binarios que usa el sistema
BASH_VERSION	Indica la versión del bash que estás utilizando
SHELL	Dirección de la shell que estás utilizando

¿Cómo crear tus propias variables de entorno?

En el home de tu usuario (**cd ~**) debe haber un archivo oculto llamado “.bashrc”, lo puedes ver ejecutando el comando **ls -la** la opción **-a** es de all.

Si tienes instalado VS Code se puede abrir dicho archivo de la siguiente manera: **code .bashrc**. Si estás en WSL y no te funciona, ve a la cmd, ejecuta el comando **wsl** y vuélvelo a intentar desde ahí.

En dicho archivo se puede crear variables de entorno.

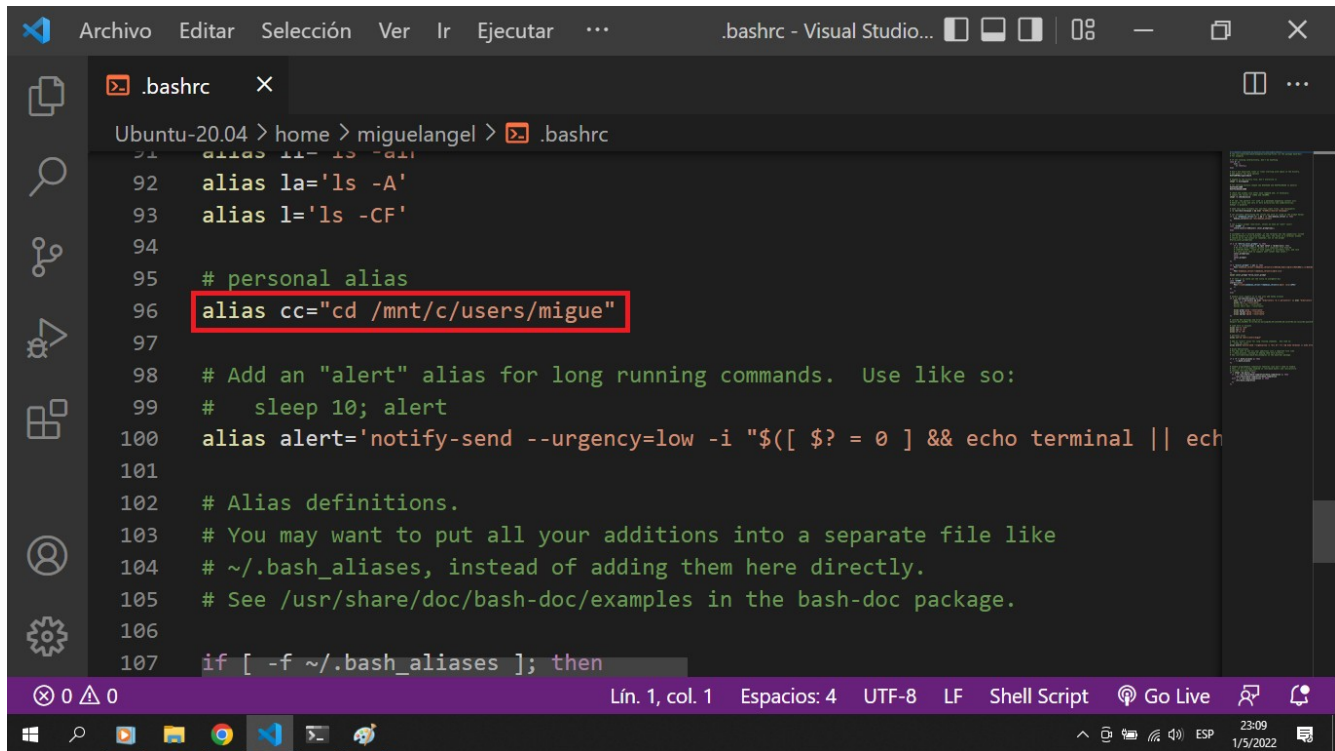
```

1  WINDOWS="/mnt/c/users/migue"
2
3  # ~/.bashrc: executed by bash(1) for non-login shells.
4  # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
5  # for examples
6
7  # If not running interactively, don't do anything
8  case $- in
9      *i*) ;;
10     *) return;;
11  esac
12
13  # don't put duplicate lines or lines starting with space in the history.
14  # See bash(1) for more options
15  HISTCONTROL=ignoreboth
16
17  # append to the history file, don't overwrite it

```

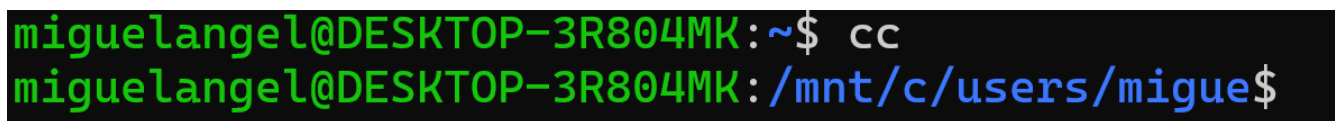
Ahora guardo el archivo, reinicio la terminal y ejecuto: **cd \$WINDOWS**

También puedo crear un alias que no se borre cuando cierre la terminal.



```
.bashrc
91 alias ll='ls -la'
92 alias la='ls -A'
93 alias l='ls -CF'
94
95 # personal alias
96 alias cc="cd /mnt/c/users/migue"
97
98 # Add an "alert" alias for long running commands. Use like so:
99 # sleep 10; alert
100 alias alert='notify-send --urgency=low -i "[ $? = 0 ] && echo terminal || echo error" -- "Terminal $?"'
101
102 # Alias definitions.
103 # You may want to put all your additions into a separate file like
104 # ~/.bash_aliases, instead of adding them here directly.
105 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
106
107 if [ -f ~/.bash_aliases ]; then
```

Ahora solo tengo que ejecutar **cc** para ir a mi carpeta en Windows sin necesidad de escribir la variable de entorno.



```
miguelangel@DESKTOP-3R804MK:~$ cc
miguelangel@DESKTOP-3R804MK:/mnt/c/users/migue$
```

- Un **link simbólico** se puede ver como un acceso directo desde la terminal.
ln -s <ruta> <Nombre> Un link simbólico se puede remover con: **rm <Nombre Link>**.
- Con **printenv** nos muestra todas las variables de entorno que tenemos configuradas.
- Para agregar una ruta a la variable **PATH** ponemos en **.bashrc** **PATH=\$PATH:<ruta>**
- Puedes ejecutar **bash** en tu terminal para que se vean los cambios.

¿Cuáles son los comandos de búsqueda en la terminal?

Para encontrar archivos de forma efectiva, usa el comando **find**, el cual buscará en la ruta que le indiques el tipo de archivos que necesitas. Su sintaxis es:

find [rutaDesdeDondeEmpezarBuscar] [opciones]

- **Segmentar por el nombre (-name)** Veamos un ejemplo, voy a buscar en mi carpeta home todos los archivos que tenga una extensión “.png”. **find . -name *.png** El punto indica que

debe empezar desde la carpeta en la que está y la opción **-name** es para especificar el nombre que debe buscar.

- **Segmentar por el tipo (-type)** si es un archivo o si es un directorio utiliza la opción **-type**, el cual acepta **f** para archivos, **d** para directorios y **l** para enlaces simbólicos.
find ./ -type f -name "f*" Esto me muestra todos los archivos que comiencen con la letra "f".
 - Veamos un ejemplo buscando archivos y directorios: **find ./ -type f,d -name "D*"**
- **Segmentar por tamaño (-size)** Con la opción **-size** podemos segmentar por tamaño ingresando el peso que queremos buscar. Hay que colocar la unidad de peso **c** para byte, **k** para Kilobyte, **M** para Megabyte y **G** para Gygabyte. Para no buscar por peso exacto, podemos especificar que sea ese peso en adelante con el símbolo **+** o de ese peso para abajo con el símbolo **-**. Ejemplo:
find ./ -size +4k
- **Buscar vacíos (-empty)** Para buscar los archivos o directorios vacíos usamos la opción **empty** que es fácil de usar, no tienes que especificarle nada, solo escribirla. Por ejemplo, si quisiera buscar todas las carpetas vacías, habría que escribir: **find ./ -type d -empty**
- **Un ejemplo importante:** **find ./ -name '*.txt' > text_docs.txt && echo "Files saved succesfully"**
- **Un ejemplo importante:** **find ./ -name *.txt | less**
- **El comando whereis** se utiliza para localizar el archivo binario, el código fuente y la página del manual de un determinado comando. **whereis [-bms] comando** El significado de las opciones es el siguiente:
 - **-b** Buscar solamente el archivo binario.
 - **-m** Buscar solamente la página manual.
 - **-s** Buscar solamente el código fuente.
- **which** es un comando muy similar a whereis, ejemplo: **which code**

¿Qué significa grep?

“Grep” significa **G**lobal **R**egular **E**xpression **P**rint. El comando grep utiliza regex (Regular Expression) para realizar su búsqueda. La sintaxis es:

grep [ExpresiónRegular] [archivoDondeBuscar/stdoutDondeBuscar]

- Ejemplo: Vamos a buscar palabras dentro de este archivo: **grep 1915 movies.csv**

```
toreohm@DESKTOP-06TNPIJ:~/Documents$ grep 1915 movies.csv
7065,"Birth of a Nation, The",Drama|War,1915,6,92
7243,Intolerance: Love's Struggle Throughout the Ages,Drama,1915,4,82
62383,"20,000 Leagues Under the Sea",Action|Adventure|Sci-Fi,1915,9,92
112062,Camille Claudel 1915,Drama,2012,5,72
119155,Night at the Museum: Secret of the Tomb,Adventure|Children|Comedy|Fantasy,2014,4,74
toreohm@DESKTOP-06TNPIJ:~/Documents$
```

- **Ignorar case sensitive (-i)** Puede que queramos buscar la palabra “Action” pero eso dará exclusivamente las coincidencias con la “A” mayúscula. Esto lo podemos ignorar con la opción **-i**, que buscará independientemente de si la letra “A” es mayúscula o minúscula.
grep -i Action movies.csv
- **Contar ocurrencias (-c)** Si quieres saber cuántas veces se repite una palabra, usa la opción **-c** seguida de la palabra que quieres buscar. Ejemplo: **grep -c Drama movies.csv**
- **Excluir una expresión (-v)** Para saber cuáles son los resultados que NO coinciden con tu expresión regular, usas la opción **-v**. Por ejemplo, si queremos contar todas las películas que no son de drama, escribimos: **grep -cv Drama movies.csv**
- **Limitar la búsqueda (-m)** Para no buscar en todo el archivo, sino las primeras ocurrencias, podemos limitar la búsqueda en líneas con la opción **-m** seguida del número de líneas que queremos encontrar. Por ejemplo, si queremos buscar las primeras 10 líneas que concuerden con la palabra “Fan” escribimos: **grep -m 10 Fan movies.csv**
- **Otro ejemplo: grep -vim 10 Drama movies.csv > resultado.txt && ls -l**
- **Otro ejemplo:** Imagina que tienes un archivo llamado test.txt y adentro contiene la siguiente frase: *Imagina que quieres buscar algo* Entonces, podemos usar grep así:
grep "Imagina .* algo" test.txt grep buscará alguna coincidencia, la expresión **.*** indica que ahí dentro puede haber una o más letras, cualquier que sea, así que podrías leerla como: Imagina (cualquier cosa) algo.

Utilidades de red

- **ifconfig** Cuando ingresamos el comando podemos ver el nombre del dispositivo de red, en este caso es “eth0”, y su configuración, tenemos su dirección IPv4 e IPv6 y su máscara de red.
- También tienes la opción del comando **netstat -i** solo que te lo mostrará de forma más amigable usando una tabla.
- **El comando ping** A veces queremos saber si una página está disponible desde nuestra dirección IP. Para esto escribimos el comando seguido de la URL a la que queremos acceder. El comando ping envía paquetes a esa página y evalúa el tiempo de respuesta. Ejemplo
ping www.google.com
 - **Limitar los paquetes enviados (-c)** Para limitar la cantidad de paquetes que enviamos, usamos la opción **-c** seguida del número de paquetes por enviar. **ping -c 4 www.google.com**
 - **Especificar el tamaño de los paquetes (-s)** Para probar la conectividad con paquetes de diferentes tamaños se utiliza la opción **-s** seguido del tamaño del paquete que desees usar. El tamaño debe ser en bytes. Para hacer pruebas con paquetes de 20 bytes escribimos:
ping -s 20 www.google.com
- **curl** Podemos obtener archivos que nos proporcione un sitio web o dirección IP con el comando curl. Este te mostrará la información que haya encontrado en la consola. **curl www.google.com**
Al ejecutar este comando te dará el documento “.html” de Google.

- **El comando wget** hace algo similar al comando curl. Solo que en vez de mostrar lo que ha obtenido por consola lo guarda en el archivo que le especifiques. **wget www.google.com**
También podemos especificar varias direcciones para descargar varias páginas al mismo tiempo. **wget www.google.com www.platzi.com**
- **Ruta de acceso a la página (traceroute)** Nos sirve para ver por cuáles computadoras (servidores) tenemos que ir pasando para llegar por ejemplo a una pagina web. En este caso nos saldrán las ips que tenemos que pasar para llegar a la pagina que queremos.

Comprimiendo archivos con formato .tar

Para comprimir con este formato en la terminal usamos el comando tar que tiene ciertas opciones para aprender. Sintaxis: **tar [opciones] [nombreDelArchivoComprimido] [archivoAComprimir]**

- **Comprimir (-c)** Para comprimir un archivo utilizamos la opción **-c**. En todos los casos hay que usar la opción **-f** para indicar que estamos comprimiendo o descomprimiendo archivos.
Ejemplo: **tar -cf compressed.tar Documents/toCompress/**
- **Ver lo que está haciendo el comando (-v)** Si queremos ver lo que el comando está comprimiendo a medida que se va ejecutando, usamos la opción **-v**. Por cierto la opción **-v** es de “Verbose” y muchos comandos la usan, también te la puedes encontrar como **-verbose**.
Ejemplo: **tar -cvf compressed.tar Documents/toCompress/**
- **Comprimir con formato “.tar.gz” (-z)** El formato “.tar.gz” o también “.tgz” es una versión extendida del formato tradicional de compresión “.zip” que puede manejar y comprimir archivos más grandes. Para manejar la compresión de archivos “.tar.gz” o “.tgz” se usa la opción **-z** además de tener que especificar en el nombre de archivo la extensión que quieres usar. Ejemplo: **tar -czvf compressed.tar.gz Documents/toCompress/**
- **Descomprimir (-x)** Para descomprimir es mucho más sencillo, solo hay que especificar la opción **-x** y el archivo comprimido que se quiere descomprimir. Si se quiere descomprimir un archivo de extensión “.tar.gz” o “.tgz” hay que especificar la opción **-z** también.
Ejemplo: **tar -xzf compressed.tar.gz**

Comprimiendo archivos .zip

- Para comprimir usamos el comando zip con el nombre que quieres que tenga y lo que quieres comprimir. Si quieres comprimir una carpeta con archivos dentro, tienes que especificar la opción **-r** de “recursive”. Ejemplo: **zip -r compressed.zip Documents/toCompress/**.
- Y para descomprimir es incluso más fácil, solo escribe el comando unzip seguido de lo que quieres descomprimir. Ejemplo: **unzip compressed.zip**

Manejo de procesos

- **El comando (ps)** muestra los procesos que están activos en una tabla muy sencilla de entender, donde en la primera columna tenemos el process ID y en la última el nombre.

- Si quieres ver una lista más detallada de los procesos con su consumo en CPU y en RAM, además del usuario que lo activó, usamos el comando **top**.
 - Aquí podemos filtrar por user. Si presionas la tecla “u” podrás escribir el nombre de usuario por el cual quieres buscar y si presionas la tecla “h” te mostrará un cuadro de ayuda para más opciones. Para salir presiona “q”.
 - Una alternativa más amigable a top es **htop**: **sudo apt install htop**
- **Matar un proceso (kill)** Para matar un proceso usamos el comando kill seguido del PID del proceso que queremos matar. Ejemplo:

```
miguelangel@DESKTOP-3R804MK:~$ ps
PID TTY          TIME CMD
 595 pts/0        00:00:00 bash
 608 pts/0        00:00:00 ps
miguelangel@DESKTOP-3R804MK:~$ kill 595
```

En este caso si ejecuto el comando **kill 595** voy a cerrar la terminal.

Procesos en foreground y background

Cuando un proceso está en ejecución sin que sea mostrado en la terminal se dice que se está ejecutando en el **background (segundo plano)**. Si se muestra la ejecución del comando dentro de la terminal se dice que está en el **foreground (primer plano)**.

Imagina que queremos una nota desde la terminal y para eso usamos: **cat > mi_notas.txt**. Nuestra terminal se verá con el prompt esperando a que ingresemos texto...

- Podemos escribir algo y después terminar el input del texto con **CTRL+D**
- Podemos suspender/pausar el proceso con **CTRL+Z**.
- Para consultar todos los procesos que tenemos en background podemos hacerlo con el comando **jobs**.
- Si queremos traer la ejecución de nuevo a la terminal, es decir, al **foreground**; debemos usar el comando **fg** y especificar qué número de trabajo queremos continuar. Ejemplo: **fg 1**

...En este caso se suspende el proceso y se ejecuta el comando **jobs**. Seleccionamos el número de tarea y ejecutamos **fg #**. Una vez enviado al foreground veremos como se activa la ejecución del comando en la terminal y podremos seguir escribiendo nuestra nota. Recuerda que una vez terminemos de escribir presionamos **CTRL+D** para terminar el input y guardar.

- Para enviar comandos al background se puede usar el operador de control **&** al final de un comando. Ejemplo: **cat > mi_notas.txt &**
- La segunda forma es con el comando **bg**. Este sirve de manera similar que **fg** solo que en vez de traerlo al foreground este lleva un trabajo al **background**. Por ejemplo: **bg 1**

- Podemos ejecutar google chrome desde la terminal con el comando: **google-chrome-stable**

Cómo usar Vim en la terminal

- Para abrir o crear un archivo utilizando Vim escribe el comando: **vim [nombre del archivo]**
 - **vim** abre el archivo especificado. Si no existe lo crea.
- Por defecto no podrás escribir hasta que actives el modo de inserción. Para hacerlo usa la tecla **i**.
- Para salir del modo de inserción y entrar al modo normal presiona la tecla **escape**.
- En el modo normal (en el que no puedes escribir) si escribes el **slash** / activarás un buscador similar al del comando less.
- Estando en el modo normal si escribes **u** equivale a **undo** y **CTRL + R** para rehacer.
- Para borrar una línea, estando el modo normal, tienes que ubicarte sobre ella y presionar **dd**.
- Para guardar y salir presiona estando en el modo normal, activa los comandos usando **:** y escribe **"wq"**. La letra **"w"** es para guardar y la letra **"q"** es para salir, también los puedes usar por separado.
 - **wq!** fuerza la salida y guarda.
- Luego podemos revisar el contenido con el comando **cat**.
- **vim** es la actualización de **vi**