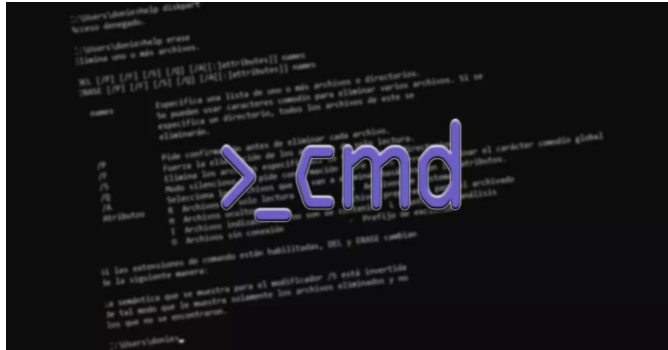




Comandos



Empezando a correr

Manejo de Permisos

Configurar variables de entorno

Comandos de búsqueda

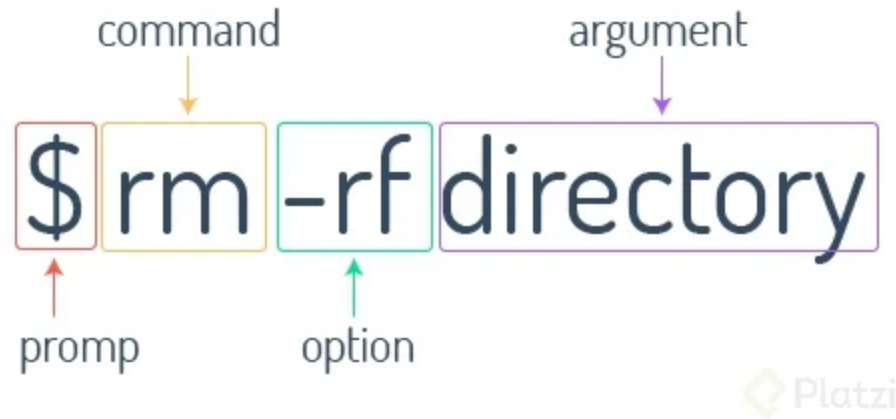
Utilidades

¿Qué es la terminal?

- La terminal es una herramienta indispensable que cualquier persona en la tecnología debe conocer 🧐. Es importante porque:
 1. Te da flexibilidad 🛠️. Con unos pocos comandos, puedes hacer mucho.
 2. Es mucho más veloz que una interfaz 💻.
 3. Es tu única opción si no hay interfaz 😄, como para configurar un servidor remoto.
 4. Puedes invocar demonios 🐱. Hay que tener cuidado con los comandos.
- Específicamente, la terminal es una interfaz gráfica muy sencilla que simula una línea de comandos:
 1. Terminal: Ventana que muestra el prompt.
 2. Shell (línea de comandos): Programa que ejecuta los comandos. Hay varios tipos de *shell*, pero sirven para lo mismo. La más comunes son *bash shell* o *Z shell*. En este curso, usaremos la primera 🍏.
- Es importante comenzar a usar Linux 😬.

- Un comando es un programa que se puede ejecutar desde la terminal.

Anatomía de un CLI o Command Line Interface



Descriptores de rutas:

- `/`: Ruta raíz del sistema
- `.`: Directorio actual
- `..`: Directorio anterior
- `~`: Directorio home del usuario

Atajos de teclado:

- `CTRL-C`: Termina el proceso de un comando en la terminal
- `CTRL-D`: Termina el input de un comando
- `CTRL-A`: Avanza al inicio de línea
- `CTRL-E`: Avanza al final de línea
- `CTRL-L`: Limpia la pantalla de la terminal

Operaciones con directorios:

- `pwd`: Imprime el directorio actual
- `mkdir dir1`: Crea el directorio con nombre dir1
- `cd dir1`: Cambia al directorio con nombre dir1
- `cd ../../`: Cambia dos directorios anteriores del actual
- `cd`: Cambia al directorio home del usuario
- `ls`: Muestra archivos y directorios

`tree` : sirve para visualizar la estructura dentro de una ruta. Para ver cierta cantidad de niveles se utiliza “-L 2”, donde el número representa la cantidad de niveles que se va a imprimir, en este caso es 2.

Operaciones de ls:

- `-a` : Muestra todo (incluyendo archivos ocultos)
- `-R` : Muestra una lista de manera recursiva
- `-r` : Muestra listando de forma inversa
- `-t` : Muestra los últimos modificados
- `-S` : Muestra ordenando por tamaño
- `-l` : Muestra usando un formato largo

Manipulación de archivos y directorios:

- `touch newFile` : Crea un nuevo archivo llamado newFile
- `file mi_archivo` : Muestra las características de mi_archivo
- `cp file1 /destino` : Copia el archivo file1 a la ruta /destino
- `cp -r dir1 dir1_cp` : Copia el directorio dir1 y su contenido a uno nuevo llamado dir1_cp
- `mv <archivo> <archivo con nueva ruta>` : Para mover archivos a otra ruta. También te sirve para renombrarlos 🌟, funciona con directorios y con archivos.
- `rm <archivo>` : Elimina archivos. Hay una opción muy útil que es `rm -i <archivo>` (interactivo) que te pregunta explícitamente si en verdad si quiere borrarlo 😞. Funciona de manera directa para *directorios vacíos*, pero si hay archivos dentro, debes usar la opción recursiva `rm -r <directorio>`, que lo que hace es ir borrando todo lo que hay dentro y al final borra el directorio 🗑️. Te va preguntando por cada cosa.
- Ojo: Lo que borras desde la terminal se borra **para siempre** 😞.

Manipulación de archivos de texto:

- `head <documento de texto>` : Nos muestra las primeras 10 líneas de un archivo de texto. Para especificar el número de líneas `head -n <numero de lineas> <archivo>`
- `tail <documento>` : Nos muestra las últimas 10 líneas.
- `less <archivo>` : Explora el contenido de archivo con paginación, nos permite hacer scroll, y nos permite hacer búsquedas haciendo `/<palabra a buscar>`. Para salir presionamos `q` 🔍.

`xdg-open <archivo>` : Para abrir un archivo desde la terminal. Usa las aplicaciones predeterminadas. Esto para linux, para mac, es `open` . Esto crea un proceso en la terminal que no nos dejará hacer nada mas. Para terminar el proceso `ctrl+c` .

Exploración de comandos y ayuda dentro de la terminal:

`nautilus` nos permite abrir el explorador de archivos en una posición dada (en linux) 📁

`type <comando>` : Nos permite conocer que tipo de comando es 🤔.

`alias aliasname="<secuencia de comandos>"` : Nos permite crear comandos. Son temporales, se borran al cerrar la terminal 🧠.

`help <comando>` : Nos permite consultar un poco de documentación de un comando 📄.

`man <comando>` : De *manual*, nos permite conocer mucha mas 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.

`man command` : Muestra el manual de usuario del comando

Wildcards

Las **wildcards** son comodines de búsqueda, que nos ayuda a encontrar o trabajar con esos archivos de manera más rápida. Ejemplo de su suso: `rm suel?.txt`

* Coincide con cualquier carácter

? Coincide con cualquier carácter individual

[caracteres] Coincide con cualquier carácter que sea miembro del conjunto caracteres

[!caracteres] Coincide con cualquier carácter que no sea miembro del conjunto caracteres

[[:clase:]] Coincide con cualquier carácter de la clase

Clases dentro de los Wildcards:

[[:alnum:]] Coincide con cualquier carácter alfanumérico

[[:alpha:]] Coincide con cualquier carácter alfabético

[[:digit:]] Coincide con cualquier número

[[:lower:]] Coincide con cualquier letra minúscula

[[:upper:]] Coincide con cualquier letra mayúscula

