Comandos de Linux para el shell

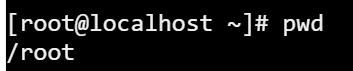
Formato general de comandos de shell

comando---- opciones ------ target

* significa home

**pwd**

En qué directorio estás



**ls**

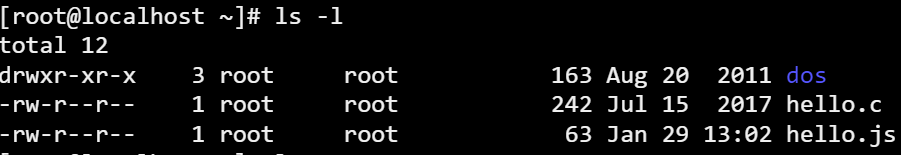
Te diste que hay en un directorio

tiene varias opciones como



**ls -l**

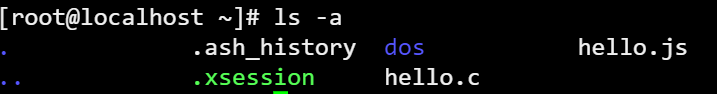
Quieres listar los archivos y directorios y saber más sobre ellos



**ls -a**

A de lall muestrame todos los archivos

El punto es el directorio actual y dos puntos es el directorio padre



**cd**

A que directorio quieres ir

La ruta realtiva (pwd)es de que directorio actual estas llegar al directorio que quieres

La ruta abosluta es desde root(inicio) hasta el directorio que quieres llegar.

Comandos que te sirven para ver el contenido de un archivo

**more**

solo puedes avanzar

**less**

te sale el archivo que estás viendo,puedes avanzar y retroceder y para salir (lo busco)

**cat**

Te sale el archivo

Viene de concatenar

**man (el comando)**

Para saber como se utiliza un comando

Cuando algo viene entre corchetes es opcional [OPTION]

touch

Crea un archivo en blanco y hay que poner que tipo de archivo

echo

es para escribir dentro de ese archivo

echo hola > archivo1.txt

> sirve para decir dónde se va a guardar

>> sobreescribe el archivo con lo que le pongas

Metacaracteres que tienen que ver con directorios

**( . ) directorio actual**

( **. . ) directorio padre**

**~ home, directorio de usuario**

**>** write

**>>** append

**<** le manda información a un comando

ejemplo combinando comandos

**sort < input.txt > output.txt**

sort < input.txt(información de entrada a uno ) (y la saluda se va a )> output.txt

si quiero saber que errores existieron puedo usar los estandares 1> o 2>

ejemplo

cat archivo > archivo 2

cat archivo x 2> errores

si quieremos ver los estandares 1 y 2

(comando ) &> (archivo al que quieres ver los errores)

**( \* )** reemplaza a 0 o más caracteres

**?**  reemplaza un o una letra 1 archivo

ejemplo: reprote\_1 ? .pdf

[a-z]

**;**  te deja usar varios comandos a la vez

**&** ejecuta comandos en segundo plano, te dice el id del proceso que se creó y se ejecuta aparte

$ variables de ambiente

ejemplo PATH = “ ”

$PATH se reemplaza por lo que vale esa variable por ejemplo el string sería ahora PATH “ “

**|** pipe le manda un output al directorio al que sigue

**ps**

te muestra los procesos

**top**

**pstree** árbol de procesos

si quieres un proceso en específico usamos

**ps ID(2344)**

**gedit &**

para saber que proceso que está abierto

**kill (id) elimina un proceso**

**getchar leer un caracter para lab 3 suspended**

**scanf recibe un input**

**ps -a**

**para ver todos los programas y aparecera un terminado**