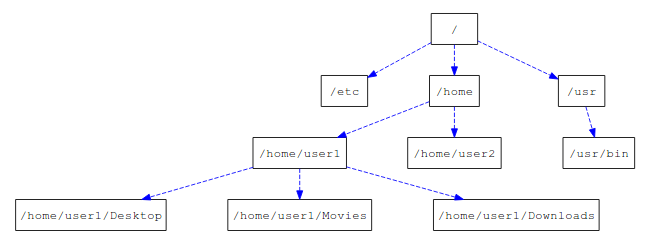
#### **Command Line Interface (CLI) - Interfaz de línea de comandos (CLI)**

**Algunos comandos**

* **cd :** Entrar en la carpeta o archivo
* **cd .. :** Ir al nivel superior
* **pwd:** mirar en la carpeta en la que estamos
* **ls:** mirar lo que hay dentro de una carpeta
* **ls -als**: muestra archivos ocultos también.
* **ls -Force:** nos muestra carpetas ocultas (ejemplo la carpeta .git)
* **rm:** elimina la carpeta o archivo
* **rm -Force**: elimina la carpeta o archivo oculto
* **mkdir**: crea una nueva carpeta
* **command** ARGUMENT
* **find . -name** "\*.py" **-type** file
* CREAR UNA VARIABLE: $ **school=**"ironhack"
* MIRAR EN UNA VARIABLE: $ **echo** school

**Organización del sistema de archivos usando comandos de terminal:**

Los archivos de un sistema Linux u OSX se organizan en lo que se denomina una **estructura de directorios jerárquica,** ya queestán **organizados en un *patrón* de carpetas** (directorios) en forma de *árbol*



### **Directorios raíz y de inicio:** El directorio de **nivel superior** del sistema de archivos se denomina **directorio raíz** ( /), el **resto subdirectorios.**

Cada usuario tiene su directorio de inicio ( /home/<your\_username> ).

Cuando abre la terminal de línea de comandos, comenzará en una ubicación determinada que suele ser su directorio de inicio (al que puede referirse con el símbolo **~**)

* **Archivos de lista - ls:** Muestra los archivos que hay dentro de una carpeta, dando así información más detallada.

**$ ls**

Applications Documents Library Music Public

Desktop Downloads Movies Pictures

* LISTA DETALLADA :

$ ls -l (“ele”)

* ORDENAR RESULTADOS POR TIEMPO DEL ARCHIVO:

$ ls -t

* ORDENAR POR EL TAMAÑO DEL ARCHIVO:

$ ls -S

* PARA INVERTIR LA CLASIFICACIÓN:

$ ls -r

* PARA INCLUIR ARCHIVOS OCULTOS:

$ ls -a

* **Ruta del directorio de trabajo:** Es la **ubicación** en la que nos encontramos en el **momento**.

**$ pwd**

# /Users/user

→ Nos devuelve la ruta absoluta del directorio de trabajo

### **Ruta absoluta vs relativa:**

* + RUTA ABSOLUTA: es la **ruta completa** de un directorio o archivo en el sistema de archivos de su máquina, que siempre comienza con un / EJEMPLO:
* **/Users/user** es la ruta absoluta de **un** **directorio**.
* **/Users/user/.DS\_Store** es la ruta absoluta de **un** **archivo**.
  + RUTA RELATIVA: es la **ruta parcial** de un directorio o archivo relativo al directorio de trabajo. En el ejemplo anterior sería .DS\_Store . No comienzan con /

Se utiliza **“.”** para hacer referencia explícita al directorio de trabajo actual. Por ejemplo, .DS\_Store es equivalente a ./.DS\_Store. Y ls es equivalente a ls .

### **¿Cómo hacer referencia al directorio principal** (directorio de un nivel superior)**? → “escalar un nivel arriba”**

→ Usando cd **..**

EJEMPLO: si su directorio de trabajo es /Users/user,  **cd ..** se refiere a /Users

* **Crear y eliminar CARPETAS:**
  + CREAR: mkdir <foldername>

$ mkdir new-folder

* + ELIMINAR: rmdir <foldername>

$ rmdir folder-name

→ Puede eliminar cualquier carpeta siempre que esté vacía. Si tiene archivos, tendrá que eliminar estos archivos antes (usando rmdir - siguiente punto de la documentación)

* **Crear y eliminar ARCHIVOS:** 
  + CREAR: touch <filename>

$ touch new-file.<*fileExtension> fileExtension* es lo que caracteriza el tipo de archivo; si está creando un archivo html, lo habría hecho

→ $ touch new-file.html

* + ELIMINAR: rm <filename>

→ $ rm new-file.html

→ Hay una opción de eliminar todo de golpe usando -r para recursivo y la opción -f para forzarlo → rm -rf.

¡OJO! al eliminar un archivo con rm -rf. Puede eliminar todos los archivos en su sistema operativo si va a la raíz de su computadora y ejecuta el comando (Se elimina incluso de la Papelera).

$ cd ..

$ rm -rf my-files

* **ENTRAR:** cd

EJEMPLO: CREAR carpeta, ENTRAR carpeta, Crear archivos:

$ mkdir my-files

$ cd my-files

$ touch file-1.txt

$ touch file-2.txt

$ touch file-3.txt

ò

$ mkdir my-files

$ cd my-files

$ touch file-1.txt file-2.txt file-3.txt

* **COMBINAR:** & Puede **combinar** dos o más comandos en una línea y guardar algunas pulsaciones de teclas:

$ mkdir & cd my-files ==> create directory my-files and enter it

$ touch file-1.txt file-2.txt file-3.txt ==> create files

* **MOSTRAR un archivo:** cat <filename>

$ cat file-1.txt

Si el archivo especificado es demasiado grande, no podrá ver todo el contenido en la pantalla. Para poder ver todo el contenido página por página, debe utilizar los comandos more o less:

$ more file-1.txt

* **COPIAR un archivo:** cp <filename>
  + COPIAR un nuevo Archivo

$ touch hello.txt ==> create files

**$ cp hello.txt copied.txt**

* + COPIAR un nuevo Archivo en OTRO DIRECTORIO

$ mkdir copied-files ==> create directory

**$ cp hello.txt copied-files/**

Observe la / (barra) al final de un directorio. Eso asegura que la ruta sea un directorio, por lo que si el directorio no existe, obtendrá un error.

* + COPIAR un Directorio: $ cp -r
* **MOVER un archivo:** Se pueden CAMBIAR NOMBRES o MOVER archivos usando mv .
  + CAMBIAR NOMBRE:

$ mv awesome.txt uncool.txt ==> Ahora debería tener un archivo llamado uncool.txt

* + MOVER ARCHIVO a otro directorio:

$ mv uncool.txt /Users/my\_name/Desktop

EJEMPLO completo de como lo deberíamos de hacer (1º creamos el directorio y entramos en él 2º creamos el archivo *3º hacemos nuestra acción* 4º verificamos que se encuentra ahí):

$ mkdir temp

$ cd temp

$ touch awesome.txt

$ ls temp

**$ mv awesome.txt uncool.txt**

$ ls

**$ mv uncool.txt /Users/my\_name/Desktop**

$ ls /Users/my\_name/Desktop

OTRO EJEMPLO: queremos mover index.txt de la carpeta 1 a la carpeta 2, y actualmente estamos en la carpeta 1:

Desktop

├── folder-1

| └── index.txt

└── folder-2

$ pwd

# ~/Desktop/folder-1 ⇒ vemos que estamos en la carpeta 1

$ mv index.txt ../folder-2 ⇒ movemos el archivo a la carpeta 2 (tenemos que ir antes al nivel anterior)

# exit the folder-1

$ cd ..⇒ salimos de la carpeta 1 y entramos en desktop (un nivel más arriba)

$ ls

# folder-1 folder-2 ⇒ vemos lo que hay en las carpetas

$ cd folder-2

$ ls

# index.txt ⇒ insertamos ambas carpetas y vemos lo que hay en ellas

OTROS COMANDOS

* cat,
* head,
* tail,
* cp,
* mv.
* "if",
* "for",
* "while"
* and both > and >>.

COLORES:

* blanco: archivo
* morado: carpeta