

Sesión 2 – Iniciación al entorno Linux

Miguel Juliá y Sarai Varona

BU-ISCIII

Unidades Científico Técnicas – SGAFI-ISCIII

17-28 Junio 2021, 8ª Edición
Programa Formación Continua, ISCIII

Índice

Aspectos teóricos >>

- Introducción. Generalidades
- Virtualización
- Cuentas de usuario

Navegación por el árbol de directorios >>

- Sistema de ficheros
- Comandos básicos

Manejo y gestión de ficheros >>

- Comandos procesamiento de archivos
- Administración ficheros y directorios

Manejo de permisos >>

- Gestión de permisos

Introducción

- Principales componentes un ordenador
- Sistema Operativo (SO)
- Interfaz
- Software Libre
- Distribuciones GNU/LINUX
- Principales elementos del Sistema operativo Linux

Principales componentes un ordenador



- **Procesador o CPU**
 - Controla el funcionamiento del computador y procesa los datos
- **Memoria y dispositivos de almacenamiento**
 - Almacenan datos, instrucciones y Programas
- **Subsistema de interconexión o buses**
 - Proporciona un medio de comunicación entre el procesador, los elementos de almacenamiento y los periféricos.
- **Dispositivos de entrada/salida o periféricos**
 - Transfieren datos entre el computador y el entorno externo

Sistema Operativo (SO)

- **Definición**

- El SO es un programa o conjunto de programas que actúa como intermediario entre las aplicaciones y el hardware del ordenador, gestionando los recursos del sistema y optimizando su uso

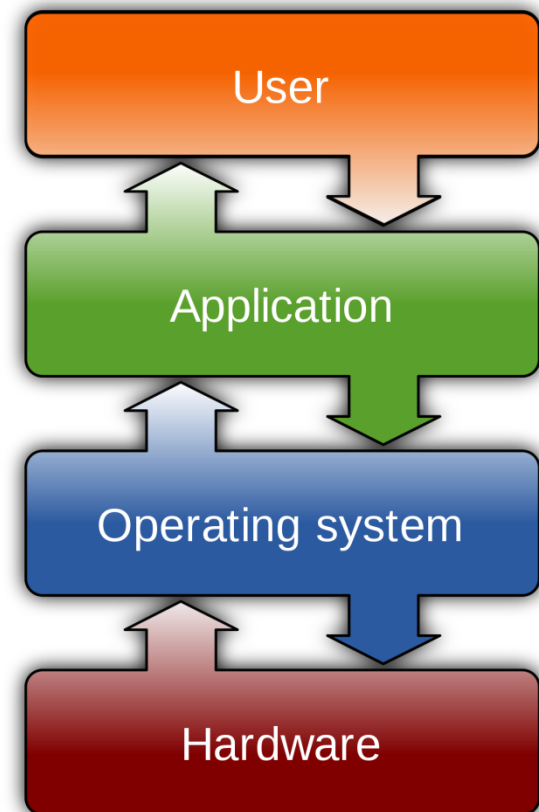
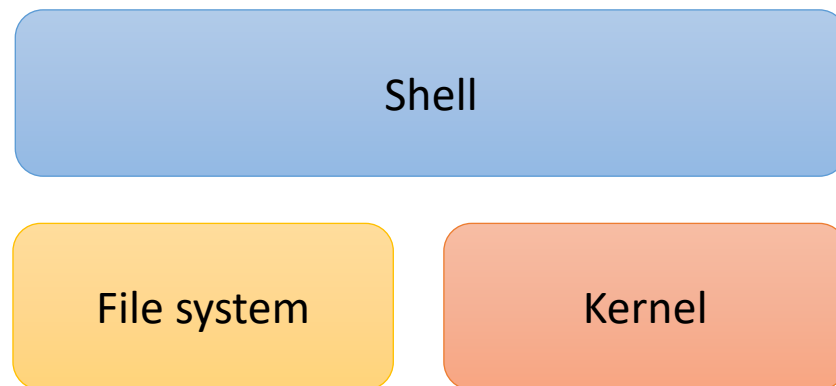
- **Funciones:**

- Control de la ejecución de los programas
- Administración de periféricos
- Gestión de permisos de usuario
- Control de concurrencia y errores
- Control de seguridad

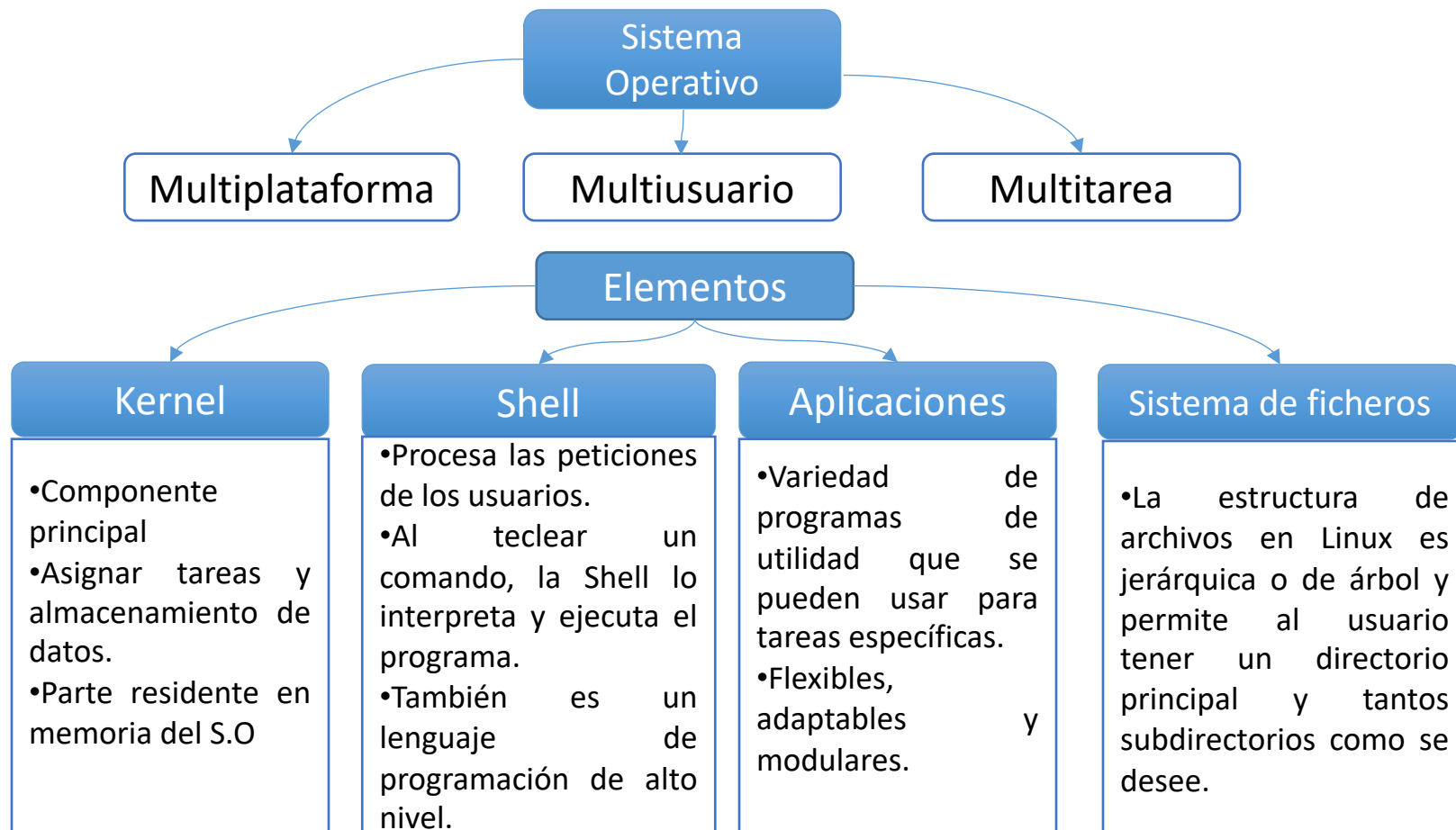


Sistema Operativo (SO)

Principales componentes:



Elementos del SO Linux



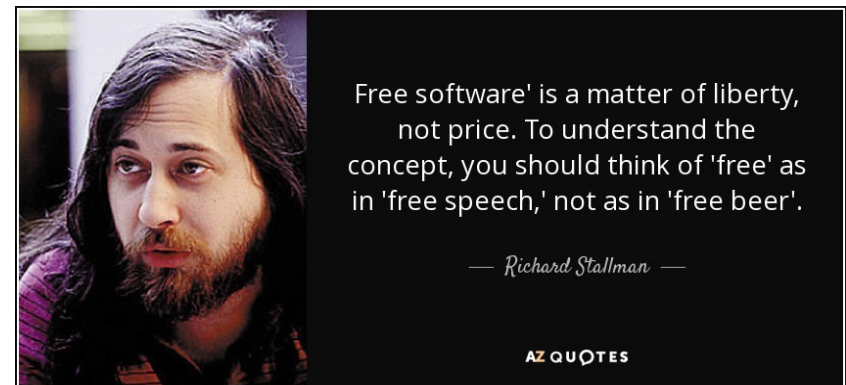
Linux OS – Open source I

The distribution terms of open-source software must comply with the following criteria:

- Free Redistribution
- Source Code
- Derived Works
- Integrity of the Author's Source Code
- No Discrimination Against Persons or Groups
- No Discrimination Against Fields of Endeavour
- Distribution of license
- License Must Not Be Specific to a Product
- License Must not Restrict Other Software
- License Must Be Technology-Neutral

Software Libre

- Concepto creado por el movimiento GNU, fundado por Richard Stallman
- ¿Qué es software libre?
 - El código fuente debe ser público y libre
 - Libre significa tener libertad sobre el código para:
 - Modificarlo, adaptarlo o mejorarlo
 - Redistribuirlo o comercializarlo
 - Ejecutarlo para cualquier propósito
 - Estudiar su funcionamiento
 - El software libre se distribuye bajo licencias distintas a las del software privativo, siendo la GNU (General Public License (GPL)) la más popular.

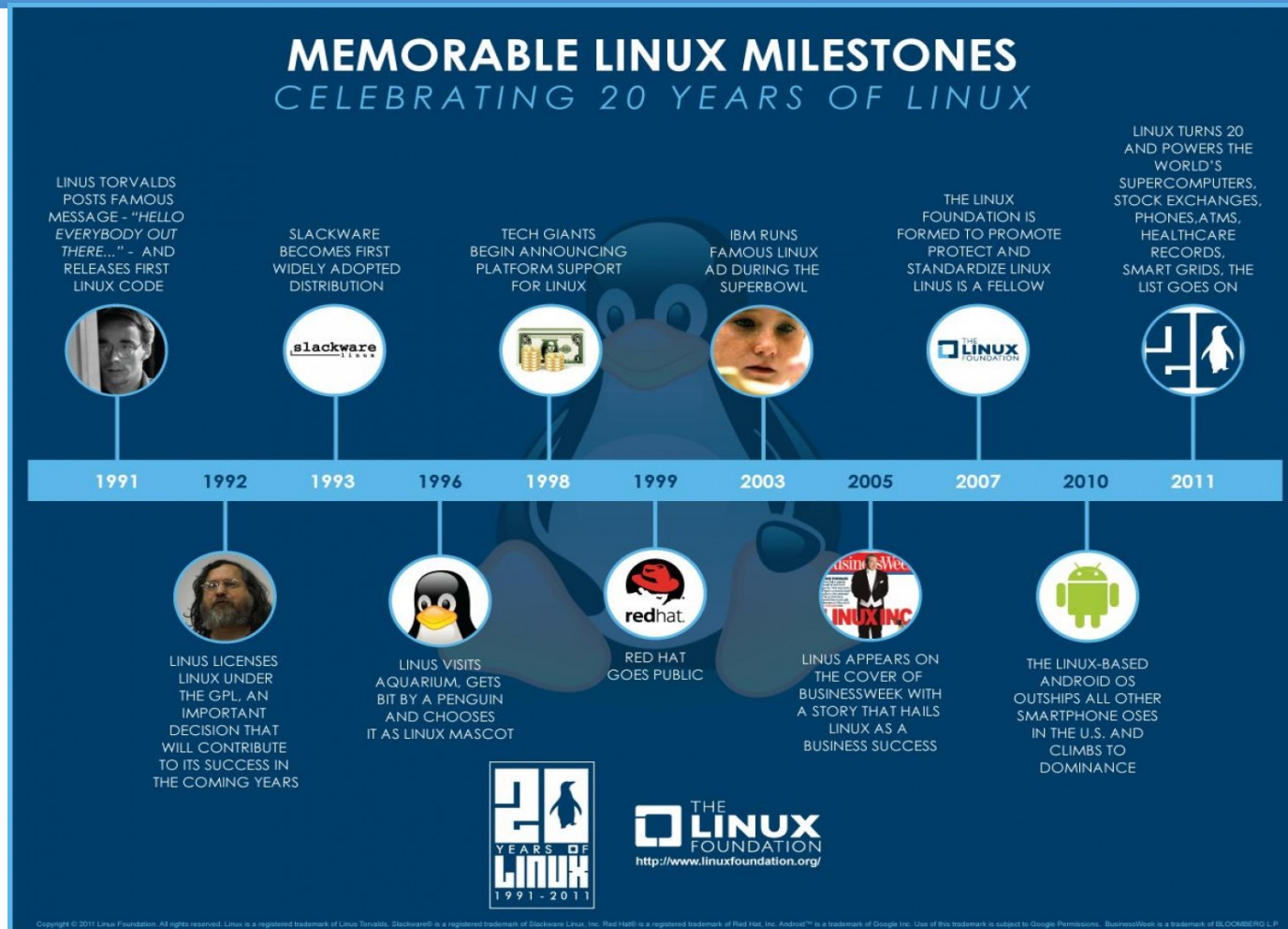


Linux OS – Open source II

Linux Distributions

- A distro is a Linux kernel based operating system made from a software collection and sometimes a package management system.
- There are distros for a wide variety of platforms.
- A typical Linux distro comprises a Linux kernel, GNU tools and libraries, additional software, documentation, a window system, a window manager, and a desktop environment.
- Most of the included software is free and open-source software made available both as compiled binaries and in source code form, allowing modifications to the original software. Usually, Linux distributions optionally include some proprietary software that may not be available in source code form, such as binary blobs required for some device drivers.

S0 Linux



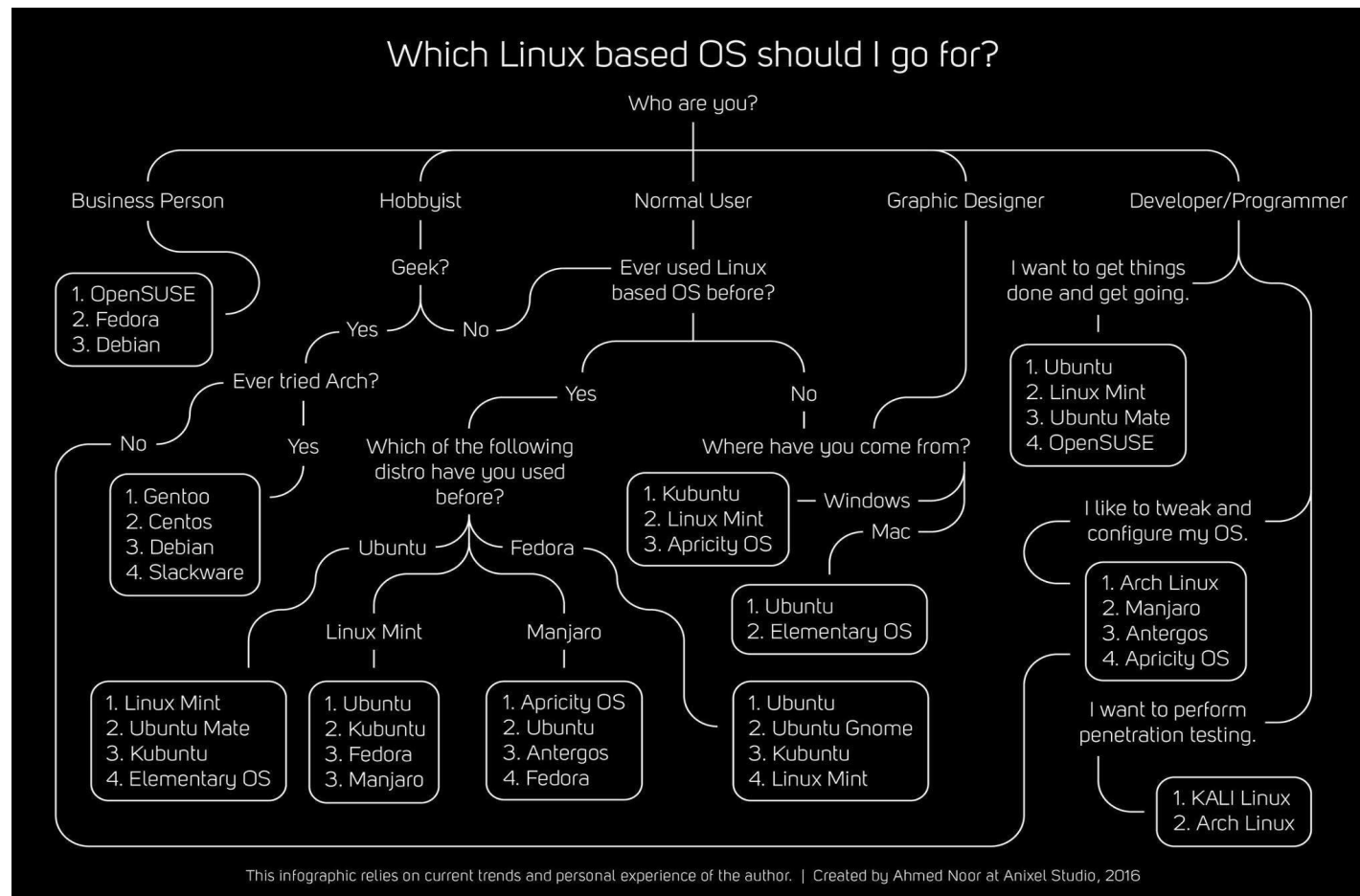
Distribuciones Linux

- Una distribución Linux es un conjunto de aplicaciones y herramientas de configuración reunidas por un grupo, empresa o persona.
- Existen numerosas distribuciones Linux. Cada una de ellas puede incluir cualquier número de software adicional (libre o no) como entornos gráficos, suites ofimáticas, servidores web, servidores de correo, servidores FTP, etc.

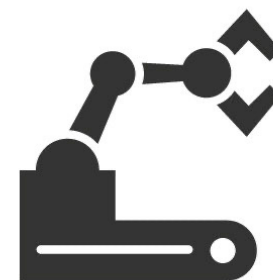
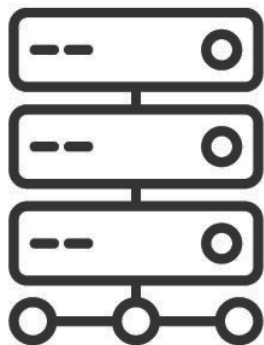
Distribuciones Linux



Distribuciones Linux



Linux OS - Multiplatform



Linux OS – Multi-user

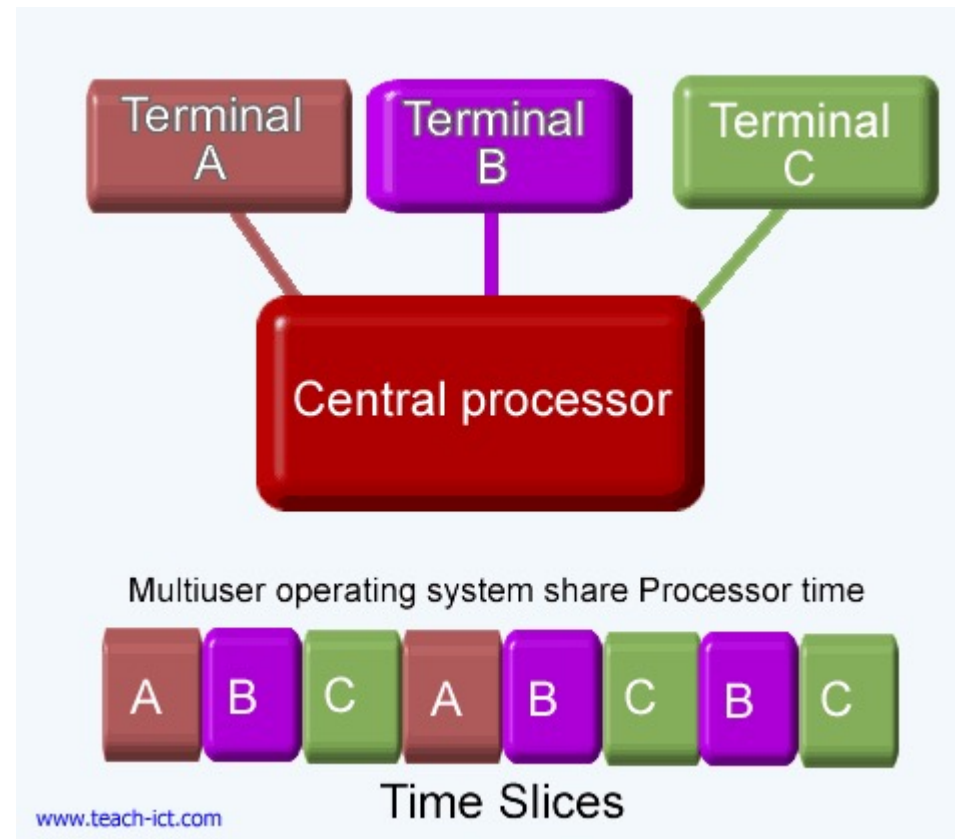
- Multi-user software is software that allows access by multiple users of a computer.
- An example is a Unix server where multiple remote users have access (such as via a serial port or Secure Shell) to the Unix shell prompt at the same time.



Linux OS - Multi-task

A multi-task operative systems allows a user to perform more than one computer task (such as the operation of an application program) at a time.

The operating system is able to keep track of where you are in these tasks and go from one to the other without losing information.



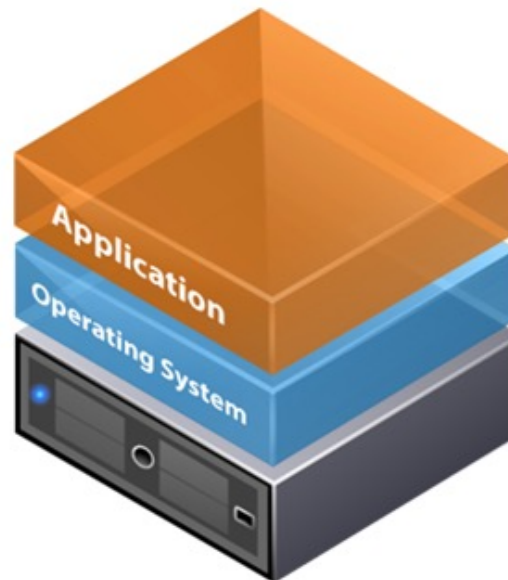
Pipeline frameworks

- Basic scripting for automating scripts is inefficient, difficult to port and maintain, and does not include support for dependencies or reentrancy.
- High-throughput bioinformatic analyses increasingly rely on pipeline frameworks to process sequence and metadata.
- Modern implementations of these frameworks differ on three key dimensions: using an implicit or explicit syntax, using a configuration, convention or class-based design paradigm and offering a command line or workbench interface.
- Choosing the right framework depends on both your lab developers and users needs.

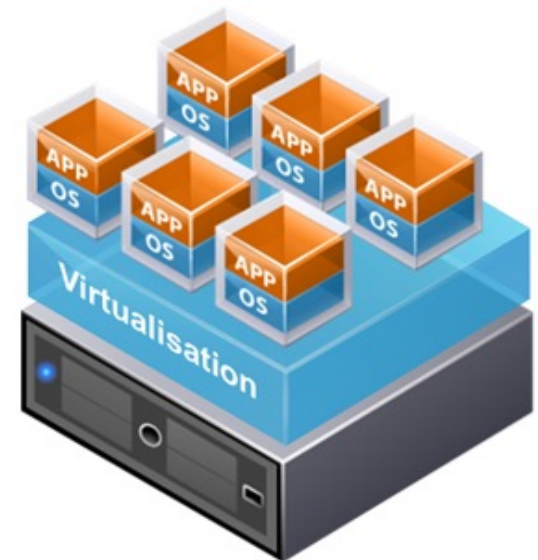
Virtualization

Virtualisation:

the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources.



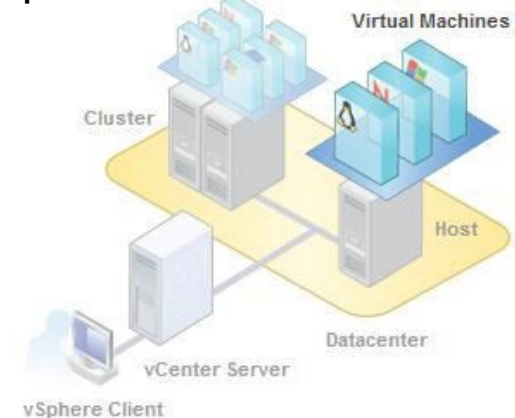
Traditional Architecture



Virtual Architecture

Virtualización

- Mecanismo de abstracción entre los recursos hardware de una máquina y el o los sistemas operativos instalados. Esto se realiza mediante la interfaz llamada Hipervisor
- La virtualización de servidores, permite ‘particionar’ un servidor físico en varios servidores virtuales
- Diferentes máquinas virtuales pueden ejecutar diferentes sistemas operativos y múltiples aplicaciones al mismo tiempo utilizando un solo equipo físico



Resumen Recursos Servidor Bioinfo01

bioinfo01.isciii.es VMware ESXi, 5.5.0, 1331820

Getting Started Summary Virtual Machines Resource Allocation Performance Configuration Tasks & Events Alarms Permissions

General

Manufacturer: HP

Model: ProLiant DL385 G7

CPU Cores: 32 CPUs x 2,199 GHz

Processor Type: AMD Opteron(TM) Processor 6274

Processor Sockets: 2

Cores per Socket: 16

Logical Processors: 32

Hyperthreading: Inactive

Number of NICs: 4

State: Connected

Virtual Machines and Templates: 34

vMotion Enabled: No

VMware EVC Mode: Disabled

vSphere HA State: N/A

Host Configured for FT: No

Active Tasks:

Host Profile:

Image Profile: ESXi-5.5.0-1331820-standard

Profile Compliance: N/A

DirectPath I/O: Not supported

Resources

CPU usage: **8753 MHz** Capacity 32 x 2,199 GHz

Memory usage: **98296,00 MB** Capacity 131021,70 MB

Storage	Status	Drive Type
Local Store	Warning	Non-SSD

Network	Type	Status
Inside_bioinformat...	Standard port group	Connected

Fault Tolerance

Fault Tolerance Version: 5.0.0-5.0.0-5.0.0

Refresh Virtual Machine Counts

Total Primary VMs: --

Powered On Primary VMs: --

Total Secondary VMs: --

Powered On Secondary VMs: --

Máquinas virtuales Servidor Bioinfo01

bioinfo01.isciii.es VMware ESXi, 5.5.0, 1331820

Name	State	Status	Provisioned Space	Used Space	Host CPU - MHz	Host Mem - MB	Alarm Actions
Ideafix16_CursoNGS	Powered On	Normal	48,00 GB	23,63 GB	21	5134	Enabled
Ideafix19_CursoNGS	Powered On	Normal	48,00 GB	23,53 GB	21	4523	Enabled
Ideafix21_CursoNGS	Powered On	Normal	48,00 GB	23,53 GB	21	5849	Enabled
Ideafix14_CursoNGS	Powered On	Normal	48,00 GB	23,55 GB	21	7056	Enabled
Ideafix17_CursoNGS	Powered On	Normal	48,00 GB	23,61 GB	153	4001	Enabled
Ideafix20_CursoNGS	Powered On	Normal	48,00 GB	23,53 GB	21	3711	Enabled
Ideafix13_CursoNGS	Powered On	Normal	48,00 GB	23,50 GB	109	4172	Enabled
Asterix01	Powered On	Normal	90,12 GB	90,12 GB	21	20485	Enabled
Ideafix23_CursoNGS	Powered On	Normal	48,00 GB	23,44 GB	21	1428	Enabled
Ideafix11_CursoNGS	Powered On	Normal	48,00 GB	23,77 GB	2924	7278	Enabled
Ideafix15_CursoNGS	Powered On	Normal	48,00 GB	23,55 GB	21	8030	Enabled
Ideafix18_CursoNGS	Powered On	Normal	48,00 GB	23,57 GB	21	5305	Enabled
Ideafix03	Powered On	Normal	65,12 GB	50,44 GB	43	6165	Enabled
Ideafix22_CursoNGS	Powered On	Normal	48,00 GB	23,49 GB	21	4544	Enabled
Panoramix01	Powered On	Normal	4,74 TB	4,74 TB	2462	2531	Enabled
Ideafix12_CursoNGS	Powered On	Normal	48,00 GB	23,54 GB	2133	5236	Enabled
Asterix_CentOs_6_5_basico	Powered Off	Normal	64,56 GB	5,15 GB	0	0	Enabled
panoramix04_swat	Powered Off	Normal	60,32 GB	6,02 GB	0	0	Enabled
Ideafix_basico_Ubuntu_1404_...	Powered Off	Normal	58,47 GB	8,18 GB	0	0	Enabled
Ideafix_Ubuntu_1404_Unity_n...	Powered Off	Normal	58,47 GB	8,18 GB	0	0	Enabled
Asterix-bio	Powered Off	Normal	92,62 GB	60,00 GB	0	0	Enabled
Bio-linux_curso	Powered Off	Normal	33,33 GB	11,78 GB	0	0	Enabled
Asterix03_H	Powered Off	Normal	64,56 GB	7,99 GB	0	0	Enabled
Panoramix03	Powered Off	Normal	60,32 GB	5,09 GB	0	0	Enabled
biowebserver01	Powered Off	Normal	52,21 GB	7,93 GB	0	0	Enabled
Asterix02	Powered Off	Normal	82,47 GB	26,52 GB	0	0	Enabled
Panoramix02	Powered Off	Normal	52,25 GB	50,00 GB	0	0	Enabled
Ideafix05_nolvm	Powered Off	Normal	71,30 GB	15,60 GB	0	0	Enabled
Ideafix_Curso	Powered Off	Normal	33,33 GB	15,17 GB	0	0	Enabled
Asterix_CentOS_6_5_basico_t...	Powered Off	Normal	64,56 GB	5,15 GB	0	0	Enabled
Ideafix_Ubuntu1404_CURSON...	Powered Off	Normal	33,33 GB	15,17 GB	0	0	Enabled
Amnesix01	Powered Off	Normal	50,29 GB	8,37 GB	0	0	Enabled
Ideafix04	Powered Off	Normal	60,47 GB	50,00 GB	0	0	Enabled
Ideafix03_curso	Powered Off	Normal	91,04 GB	32,74 GB	0	0	Enabled

Containers

Linux containers is a generic term for an implementation of operating system-level virtualization for the Linux operating system.

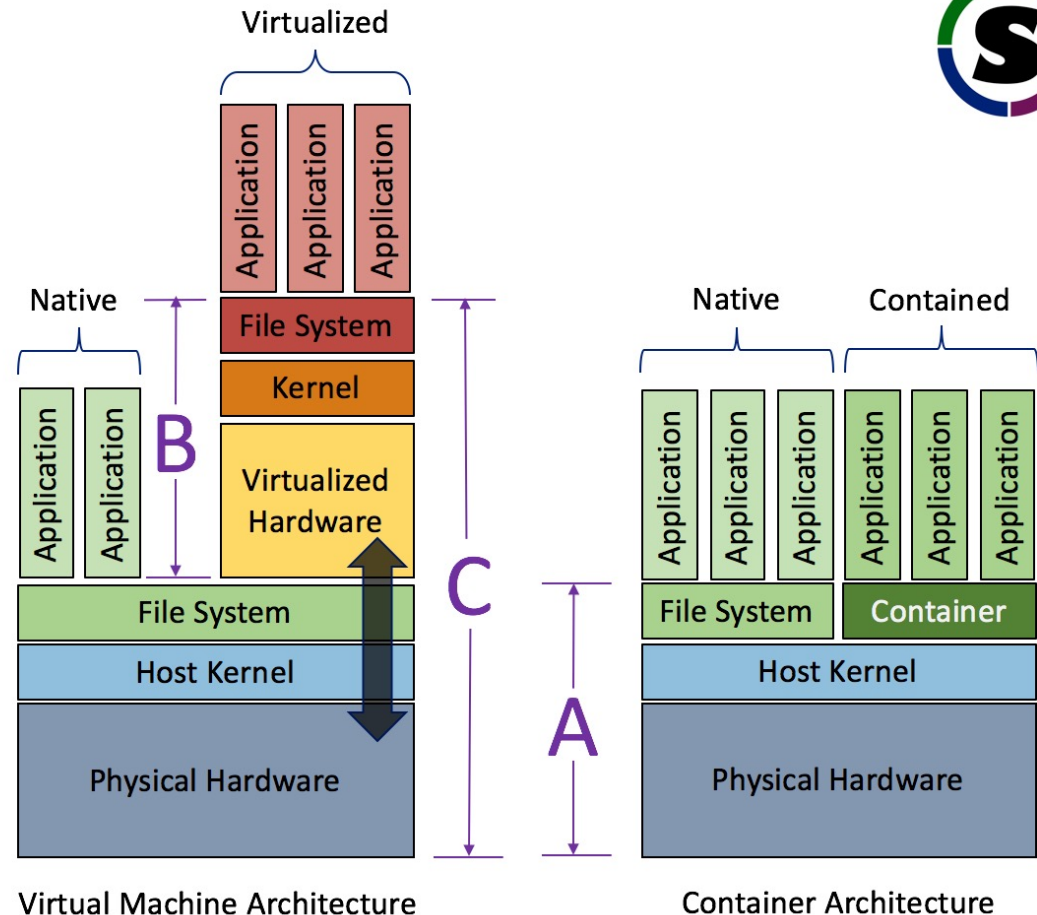
Containers allow us to port pipelines and replicate their exact execution environments across different hardware.

Currently, a number of such implementations exist, and they are all based on the virtualization, isolation, and resource management mechanisms provided by the Linux kernel. Most famous ones in Biosciences are Docker and Singularity.

Containers vs VMs



- Applications running within a container will always be “closer” to the physical hardware
 - Notice how close to native a container behaves
- Applications running through a virtual machine will always have multiple levels of indirection
- The container’s proximity to the physical hardware equates to less overhead, higher performance and lower latency



Environments

- Un **environment** (**env**) es el conjunto de archivos, recursos, librerías, paquetes de software, y variables de entorno que intervienen en la ejecución de un proceso de cómputo.
- Cualquier modificación de los componentes del environment puede inducir cambios en los resultados del cómputo, cuando no directamente a errores.
- Las VMs y containers diseñados para la ejecución de un pipeline suelen contener al menos un env preparado para la ejecución de dicha pipeline.

Conda

- **Conda** es un gestor de environments multiplataforma.
- Permite crear, modificar, y exportar environments de forma controlada y transparente.
- Permite la coexistencia de distintos envs en tu máquina y seleccionar en cada momento el necesario para el cómputo que se quiere realizar.
- El uso de entornos transparentes y bien gestionados permite la reproducibilidad de resultados y facilita la instalación y mantenimiento de las pipelines.

Cuentas de usuario

- Para acceder a cualquier sistema Linux necesitamos una **cuenta de usuario**
- La cuenta de usuario engloba los siguientes elementos
 - Nombre de usuario (Login)
 - Contraseña (Password)
 - Identificador de usuario y grupo (UID y GID)
 - Directorio de trabajo (Home directory)
 - Intérprete de órdenes (Shell)
- Al acceder con nuestra cuenta de usuario automáticamente nos ubicamos en nuestro directorio de trabajo (home)
 - Un usuario suele tener permiso total de acceso a todos los archivos y subdirectorios de su directorio de trabajo
 - El acceso a otros directorios que no pertenezcan al directorio de trabajo del usuario suele estar limitado o incluso prohibido

Cuentas de usuario

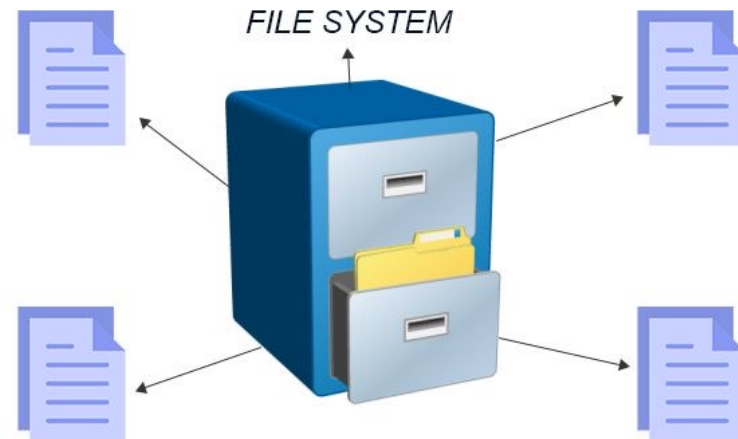
- **La cuenta del superusuario o root**
 - El superusuario es un usuario especial que actúa como administrador del sistema
 - Tiene acceso a todos los archivos y directorios del sistema
 - Tiene capacidad para crear nuevos usuarios o eliminar usuarios
 - Puede detener cualquier proceso que se esté ejecutando en el sistema
 - Tiene capacidad para detener y reiniciar el sistema
 - El **Login** del superusuario suele ser **root**, aunque no es estrictamente necesario

Navegación por el árbol de directorios

Sistema de ficheros

- Características
- Árbol de Directorios Linux
- Rutas de acceso.
 - Ruta absoluta, desde el directorio padre
 - Ruta relativa desde el directorio actual

Comandos básicos



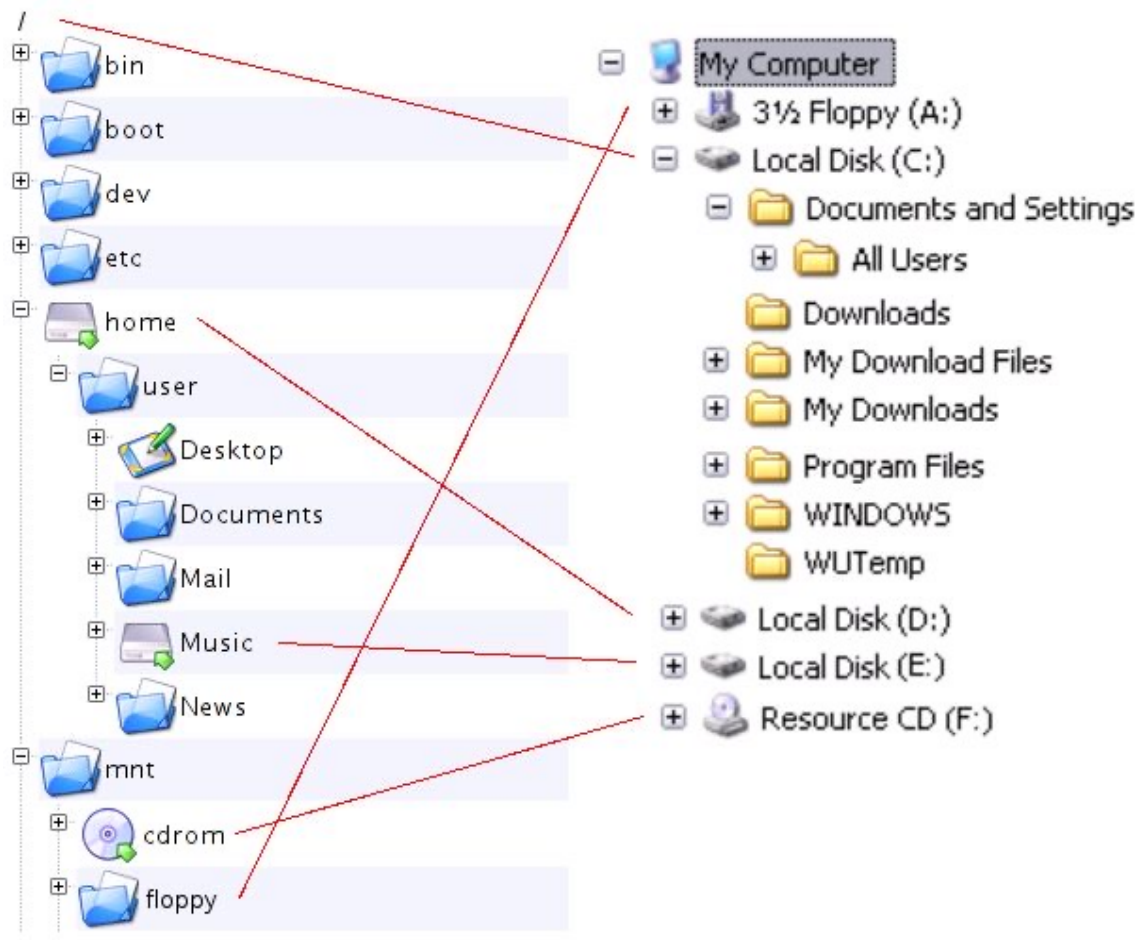
Sistema de ficheros

Características

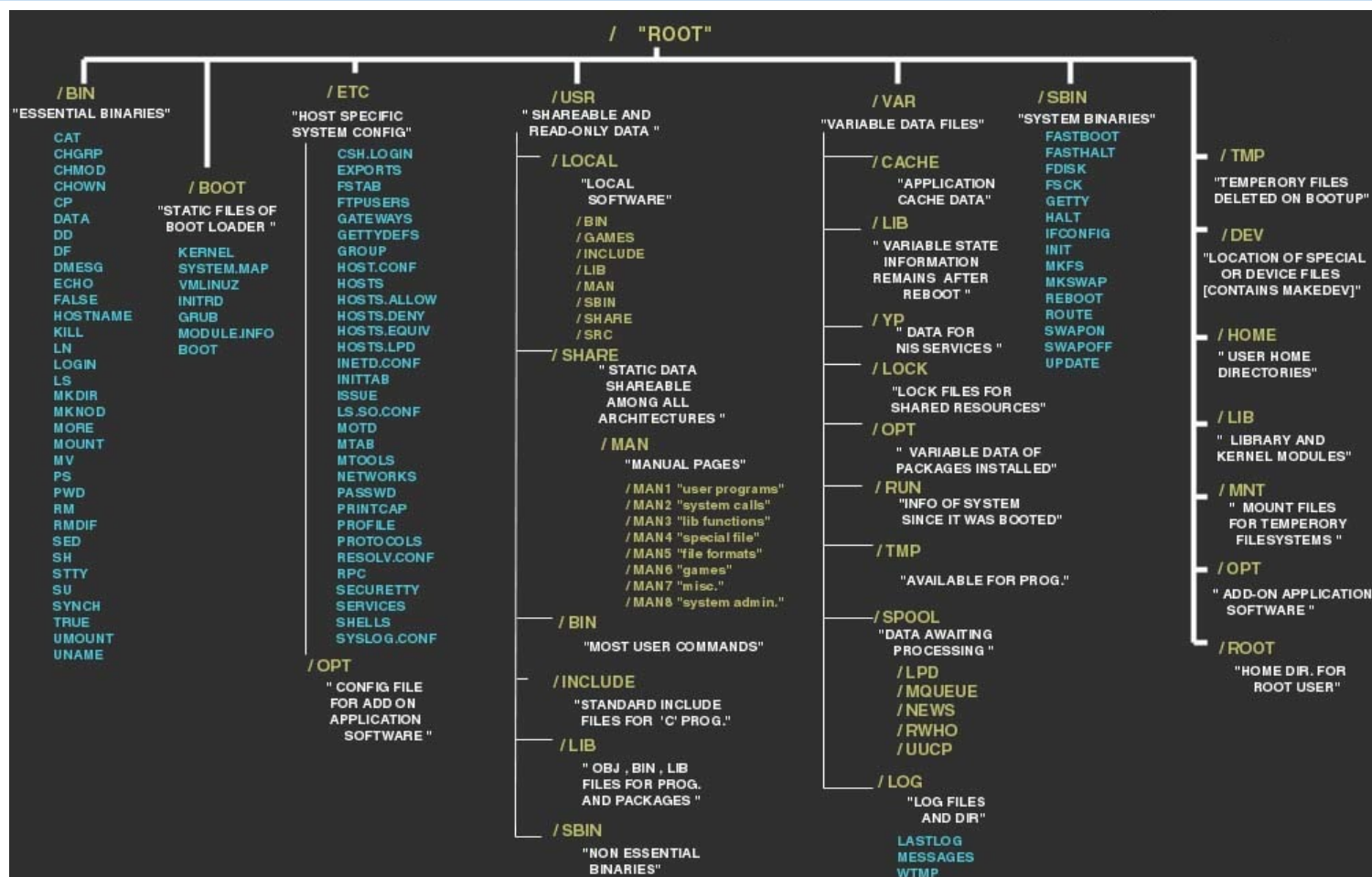
Todo es un fichero

- Los ficheros son ficheros, los directorios son ficheros y los dispositivos se representan con ficheros.
- No existe el concepto de extensiones, un programa es un fichero que tiene permisos para ser ejecutado
- Todo lo que hay en el sistema cuelga del directorio raíz /

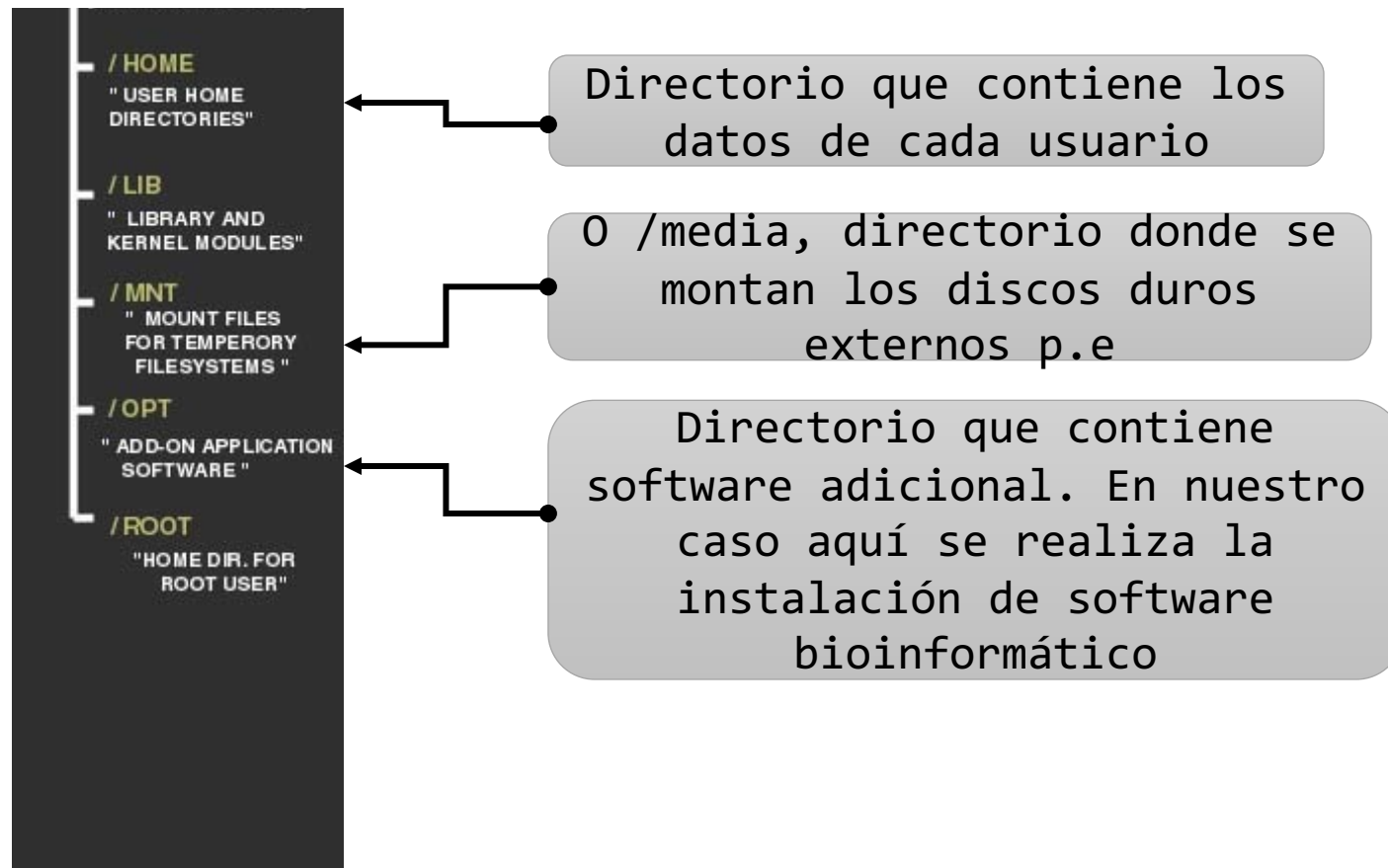
Árbol de directorios



Árbol de directorios



Árbol de directorios

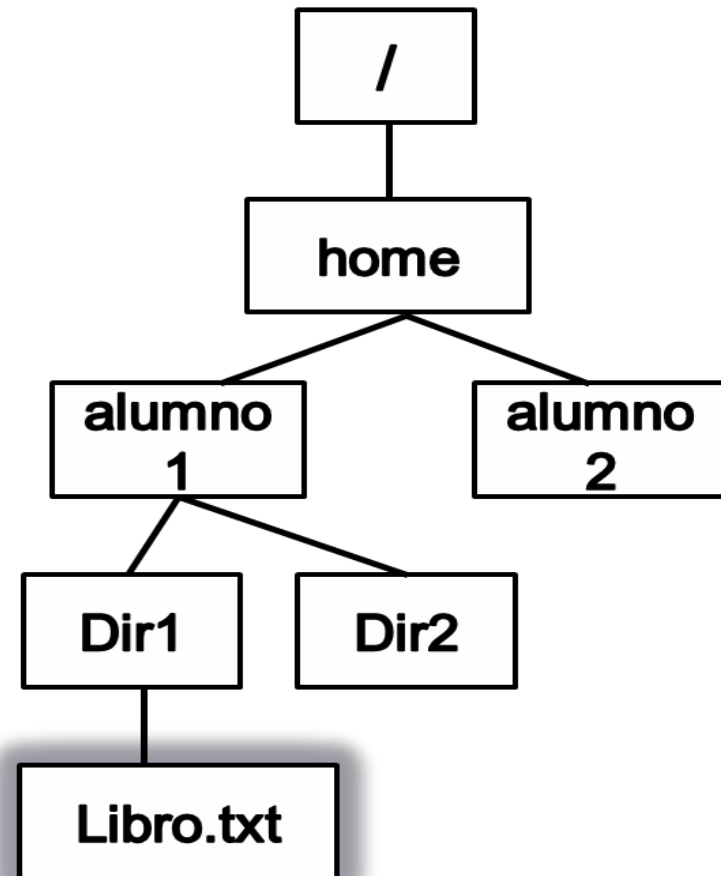


Rutas de acceso

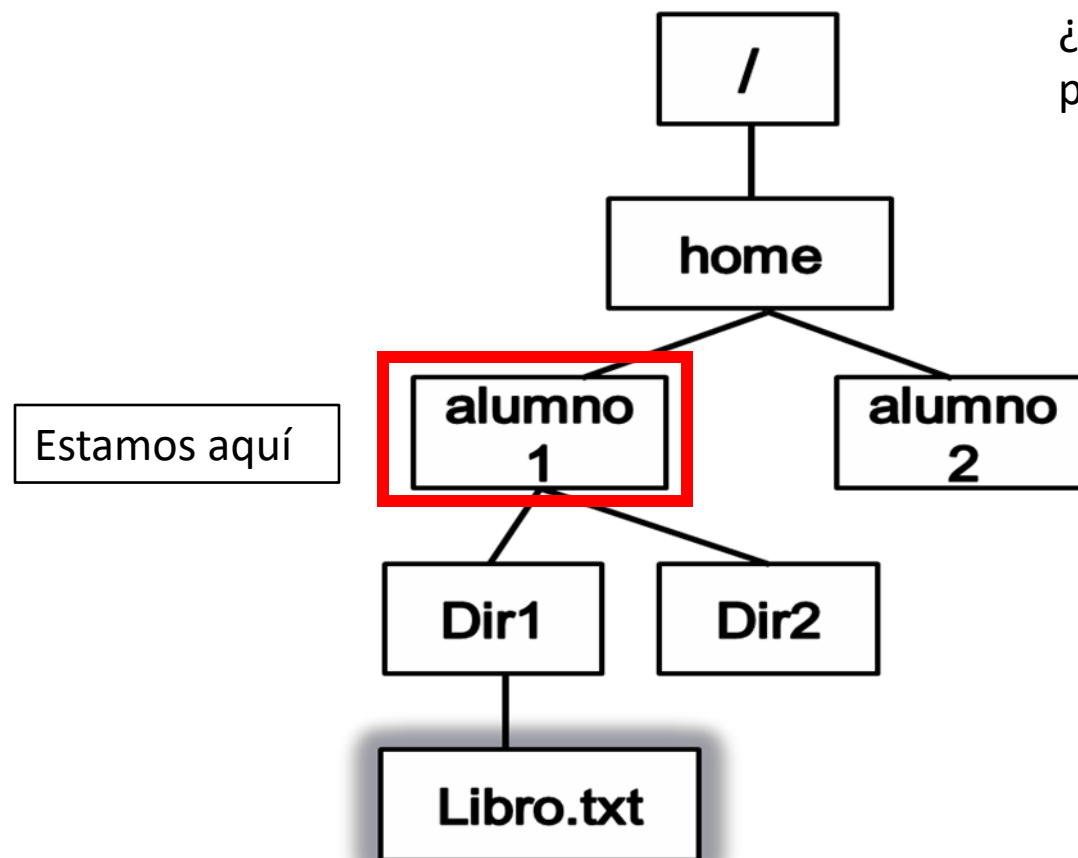
- La ruta o path de un fichero es la secuencia de directorios que se ha de recorrer para acceder a él, separados por /, es decir es la dirección donde se encuentra el fichero.
- El nombre (ruta de acceso a un fichero) puede especificarse de dos modos:
 - **Ruta absoluta:** Tomando como punto de partida el raíz
 - `/home/alumno1/dir1/libro.txt`
 - **Ruta relativa:** Tomando como punto de partida el directorio actual
 - `./alumno1/dir1/libro.txt` (desde `/home`)
 - `dir1/libro.txt` (desde `/home/alumno1`)
 - `Libro.txt` (desde `/home/alumno1/dir1`)

Rutas de acceso

- El directorio actual, se representa por el punto “.”
- El directorio padre se representa por dos puntos “..”
- Ejemplos
 - `../..` Para ir al home desde Dir1 o Dir2
 - `../..alumno2` Para ir a alumno2 desde Dir1 o Dir2.
 - `../Dir2` Para ir desde Dir1 a Dir2.

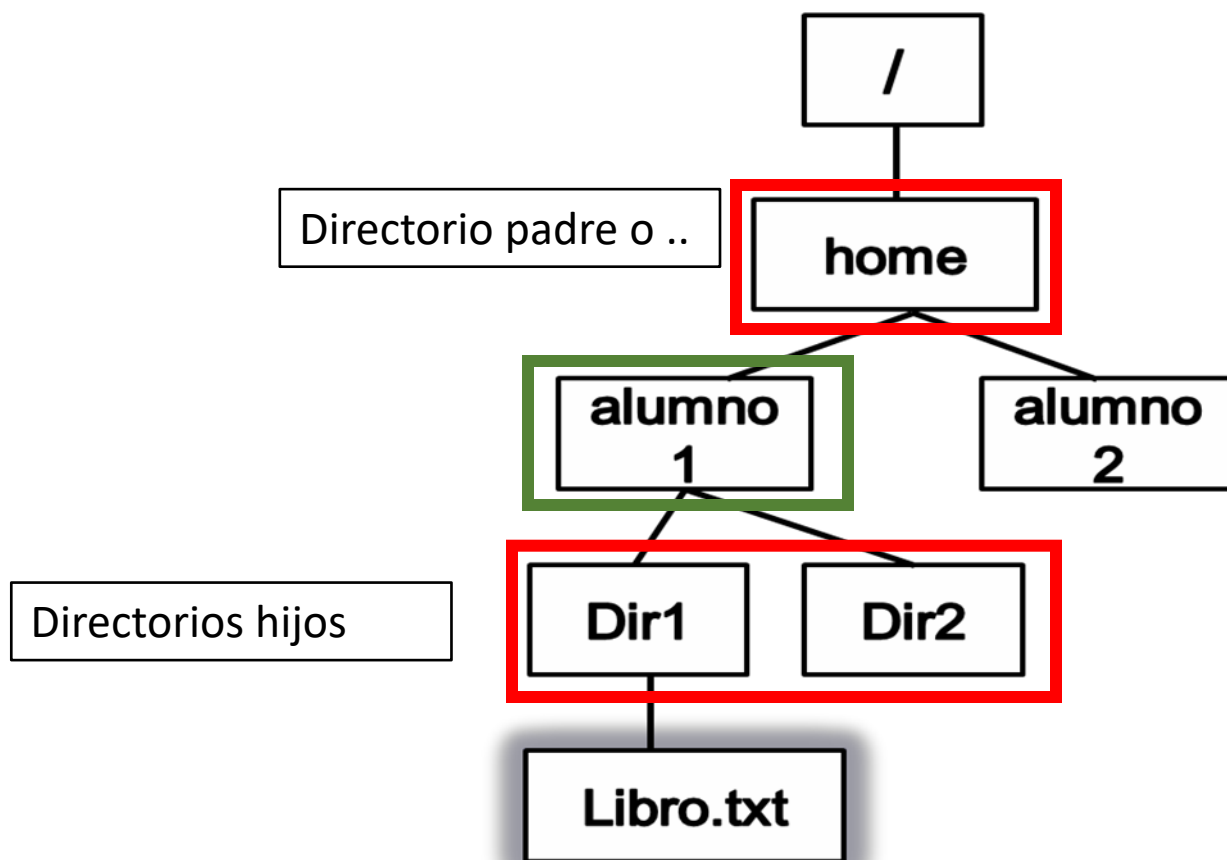


Rutas de acceso

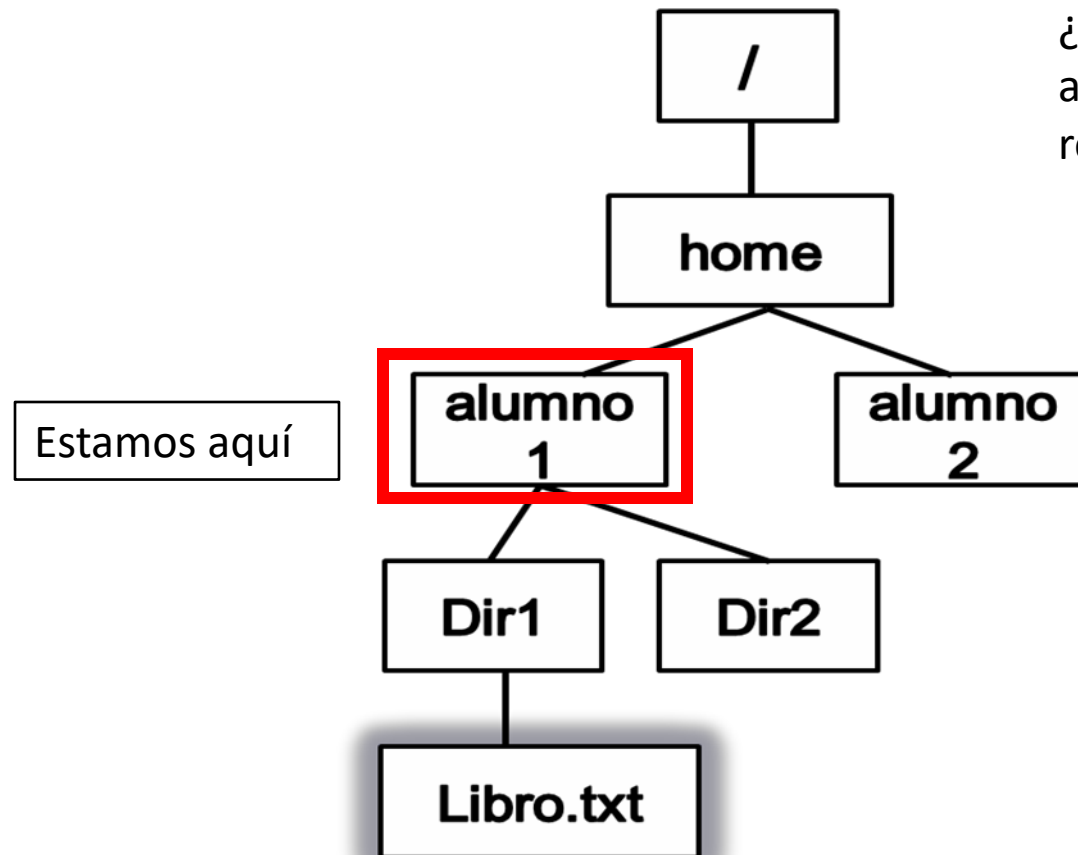


¿Quién es el directorio padre? ¿Y el hijo?

Rutas de acceso

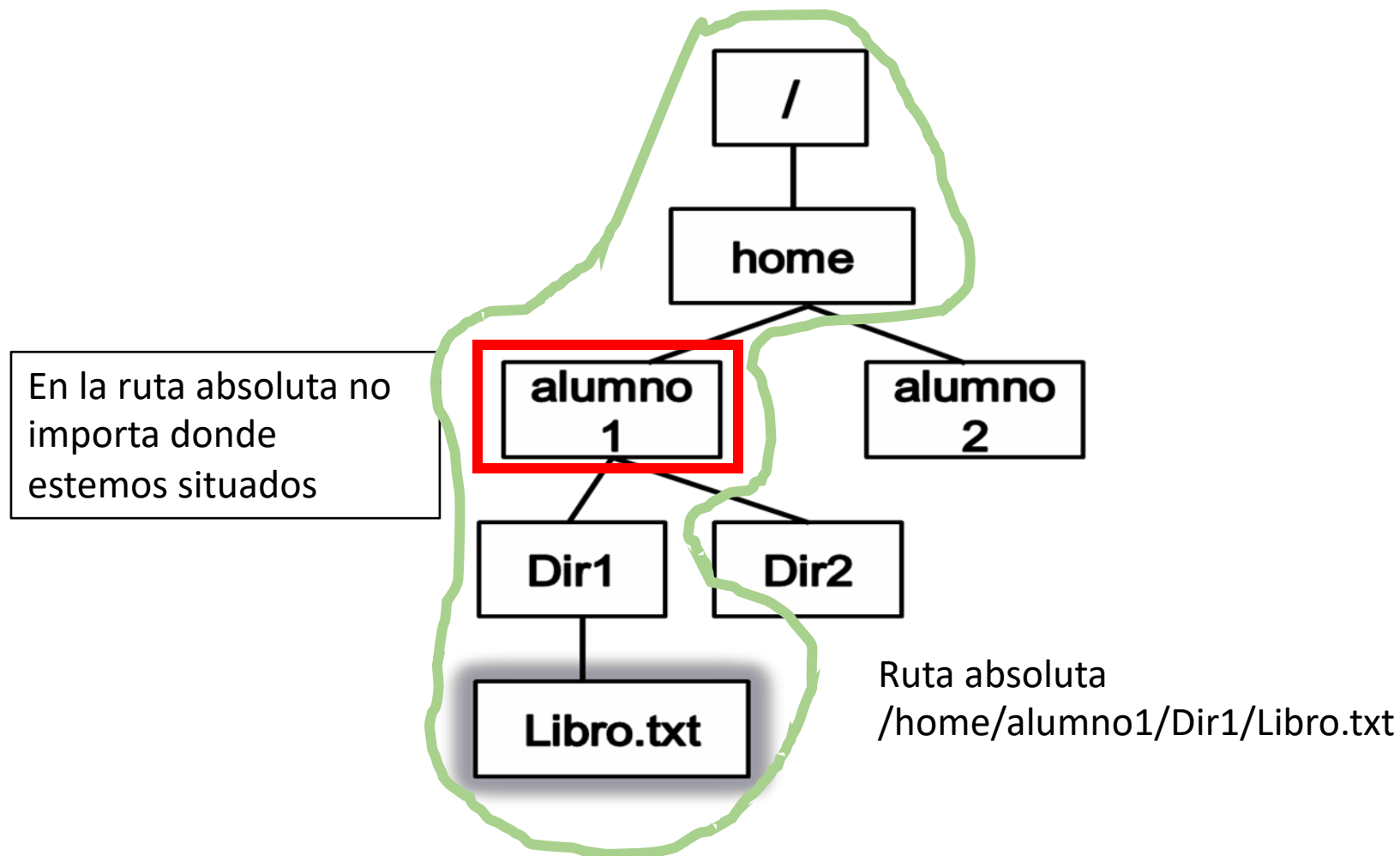


Rutas de acceso

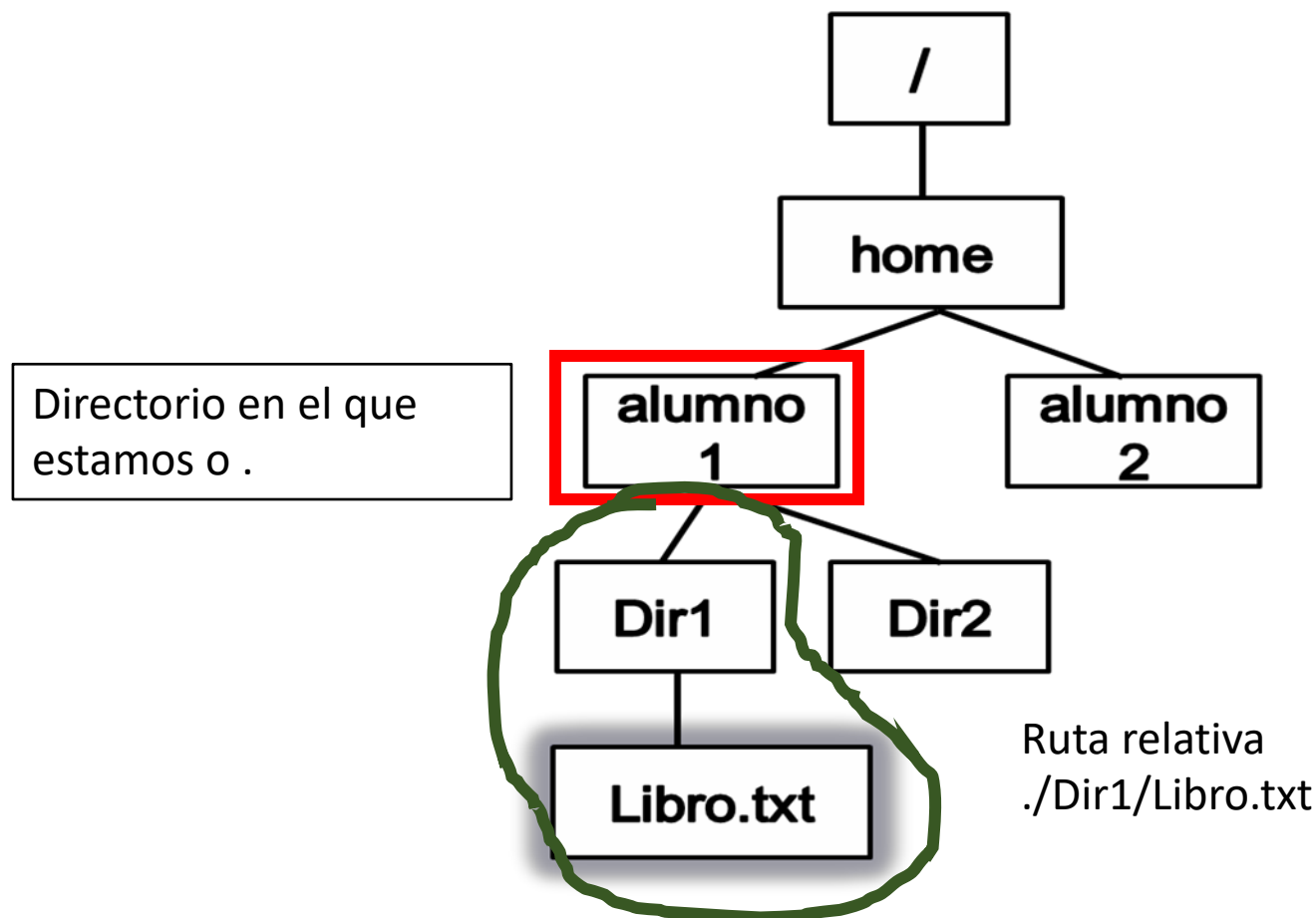


¿Cuál es la ruta absoluta a Libro.txt? ¿Y la ruta relativa?

Rutas de acceso



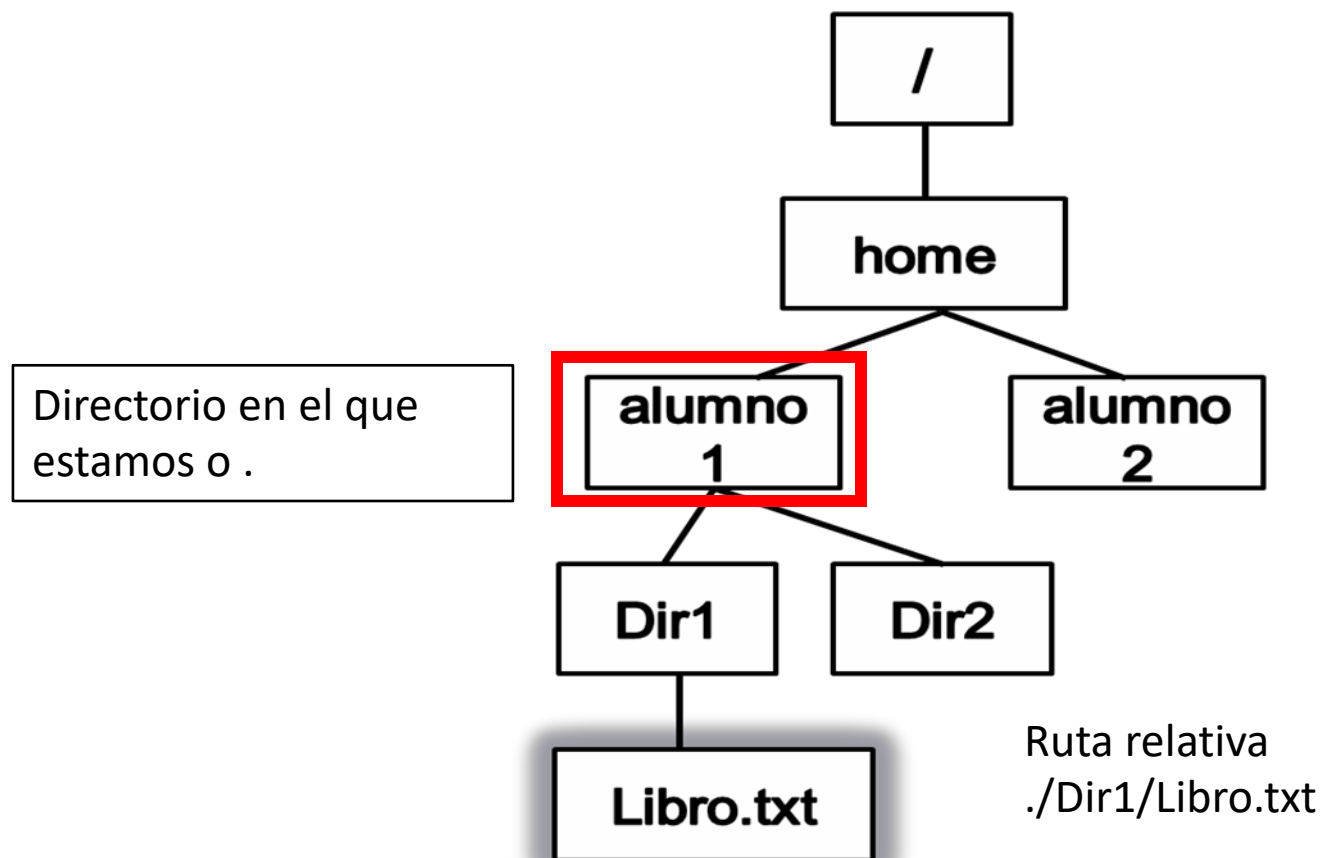
Rutas de acceso



Comandos

- Los comandos hay que teclearlos exactamente
- Las letras mayúsculas y minúsculas se consideran como diferentes (case sensitive)
- Cuando se introduce un comando o ruta de fichero o directorio, *la Shell* permite escribir las primeras letras del mismo y realiza un autorellenado al presionar la tecla del **tabulador**. Si no puede distinguir entre diversos casos rellenará hasta el punto en el que se diferencien

Comandos: Ejemplo



Línea de comandos

- El Shell espera instrucciones por el teclado en una línea llamada **línea de comandos** o **prompt**
- Esta línea de comandos nos ofrece cierta información fácilmente reconocible:

```
profesor@VM-NGS01:~/Documentos$
```

→ \$ sin privilegios
→ # con privilegios

↓ Hostname (Nombre de la máquina)
↓ Ubicación actual en el árbol de directorios
↓ Usuario conectado

- Esta línea de comandos nos informa de que el usuario profesor sin privilegios de administración esta conectado a la maquina lógica VM-NGS01 en la posición del sistema de ficheros Documentos

Sintaxis de los comandos

- En general la sintaxis de los comandos es:

Comando [-o | --opción] [argumentos]

- Los **parámetros** son opciones del comando y normalmente se escriben mediante un guion y una letra (-r por ejemplo). Se puede ejecutar mas de un parámetro por instrucción escribiendo guion y letra varias veces (-r -a) o uniendo las letras detrás del guion (-ra)
- Algunos parámetros o instrucciones requieren un **argumento**, por ejemplo el nombre de un fichero, que se colocara detrás del parámetro separado por un espacio
- Para escribir más de un comando en la misma línea los separaremos con el carácter “;”
 - Ejemplo: `$ mkdir dir1` (este comando creara la carpeta dir1 en la ubicación actual de el árbol de directorios)

Sintaxis de los comandos

- Añaden infinita flexibilidad a la invocación de comandos
- La shell los sustituye por otra expresión

?	Único carácter
*	Cadena de caracteres
~	Abreviatura del directorio home
[x-y]	Algún carácter dentro del rango

- Por ejemplo:
 - *.txt representa todos los ficheros que terminen en “.txt”

Ayuda de los comandos

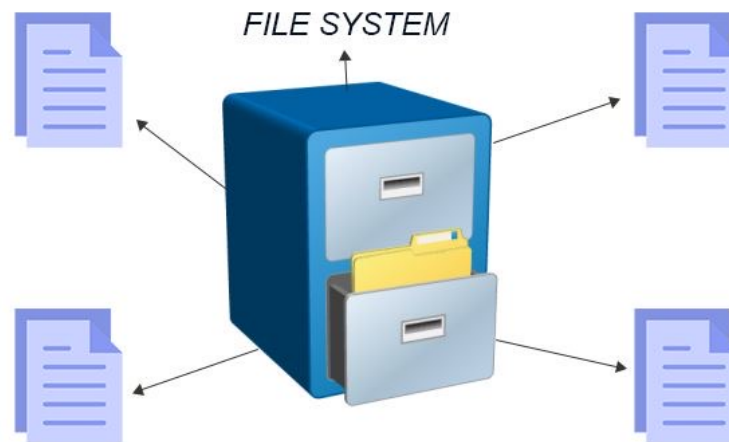
- Es imposible conocer cada uno de los parámetros y argumentos de todas las instrucciones (comandos)
- Todos los manuales de Linux están dentro del propio sistema operativo
- Existen paginas con información completa sobre los comandos. Para acceder a ellas se usa el comando **man** o el argumento **help**. Sintaxis:
 - **man comando** (\$ *man ls*)
 - **comando --help** (\$ *ls --help*)
- Se puede navegar a través de estas páginas con los cursores del teclado.
- Presionando letra **q** (quit) se sale del manual

Historial

- La Shell conserva un historial de los comandos que se ejecutan.
- Se pueden consultar con el comando ***history***
- También se puede navegar por este historial pulsando la flecha para arriba o la flecha para debajo de los cursores del teclado
- Es de gran utilidad ya que permite no tener que reescribir los comandos utilizados hace poco
 - **Sintaxis:** `$ history [opciones]`
 - **Opciones:**
 - **-c** Borra el historial

Manejo y gestión de ficheros

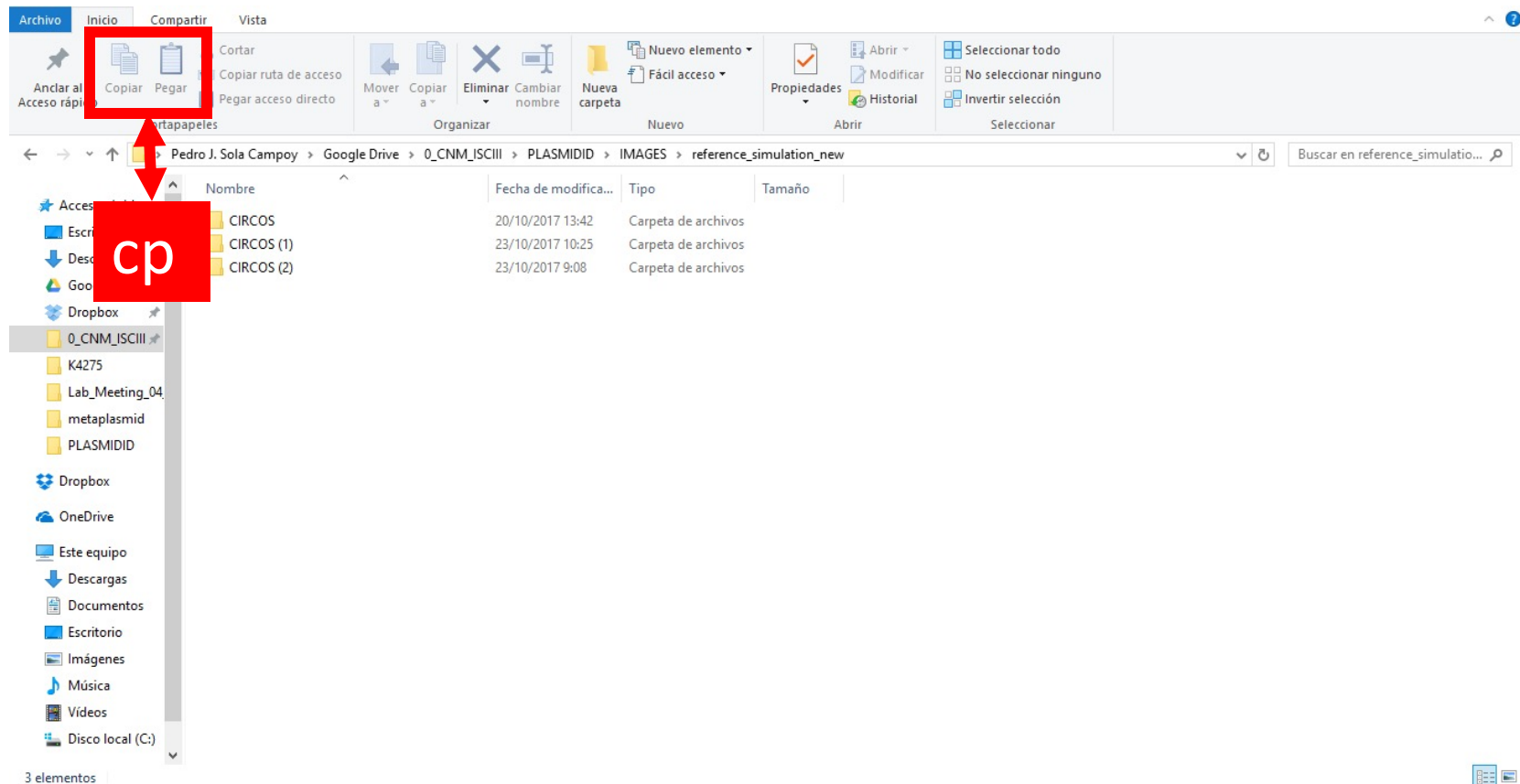
- Comandos procesamiento de archivos
- Administración ficheros y directorios



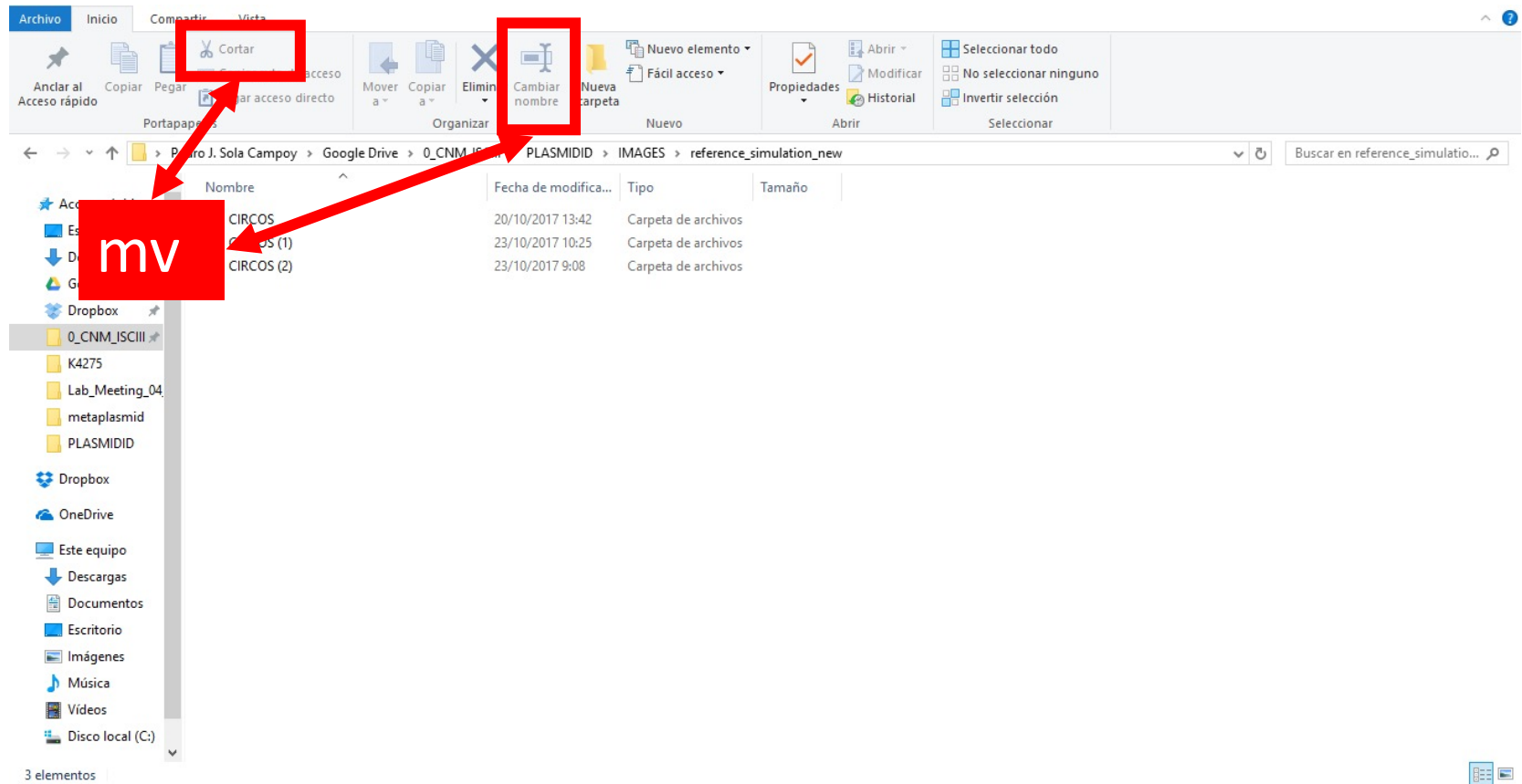
Ficheros

- No tienen extensión obligatoria y la extensión no tiene significado intrínseco
- Si el nombre comienza por “.” permanecen ocultos
- Poseen un propietario y pertenecen a un grupo
- Tipos:
 - Regulares (-)
 - Directorios (d)
 - Enlaces: Simbólicos o blandos (l), Físicos o duros (-)
 - De dispositivo (b, c)

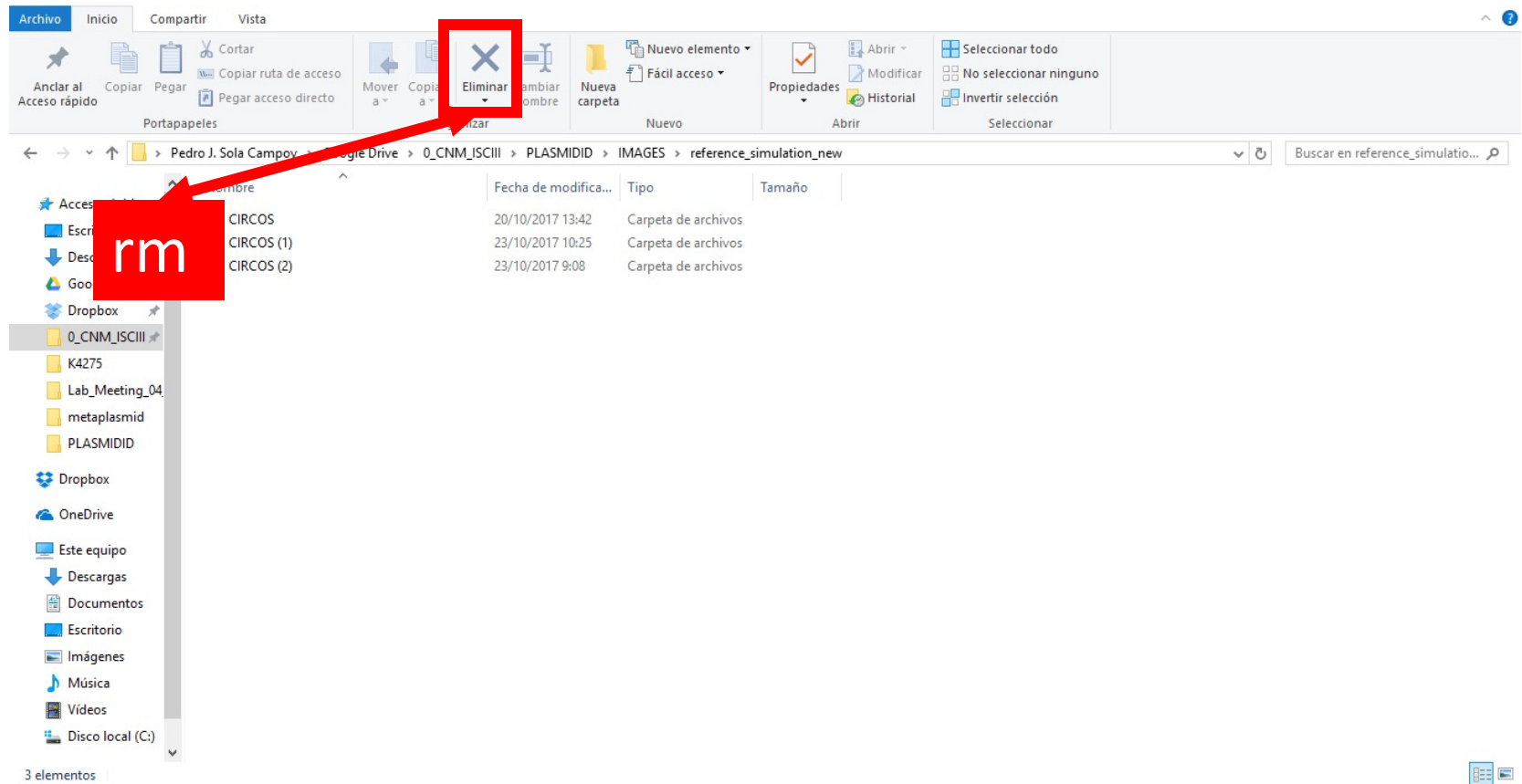
Nociones básicas



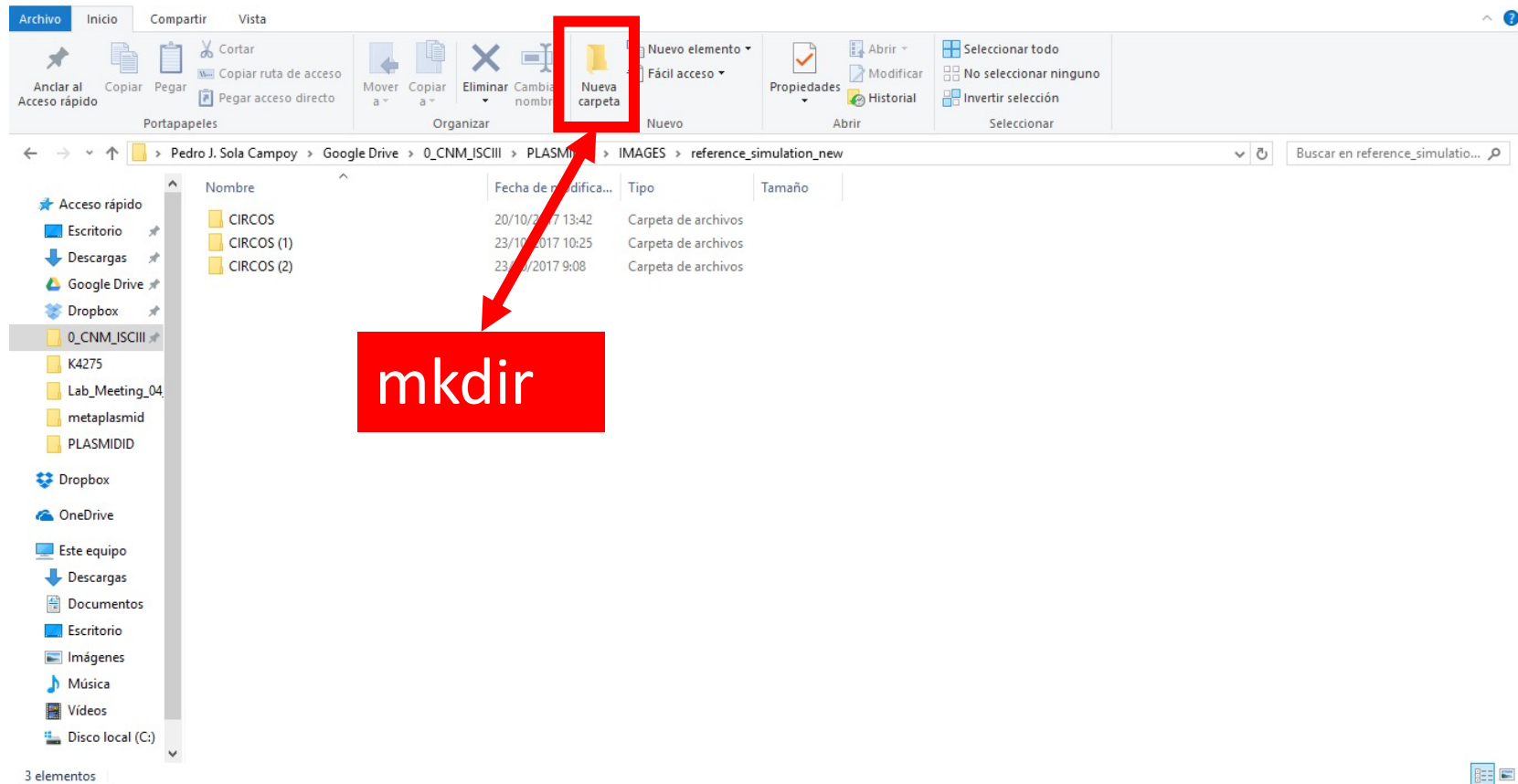
Nociones básicas



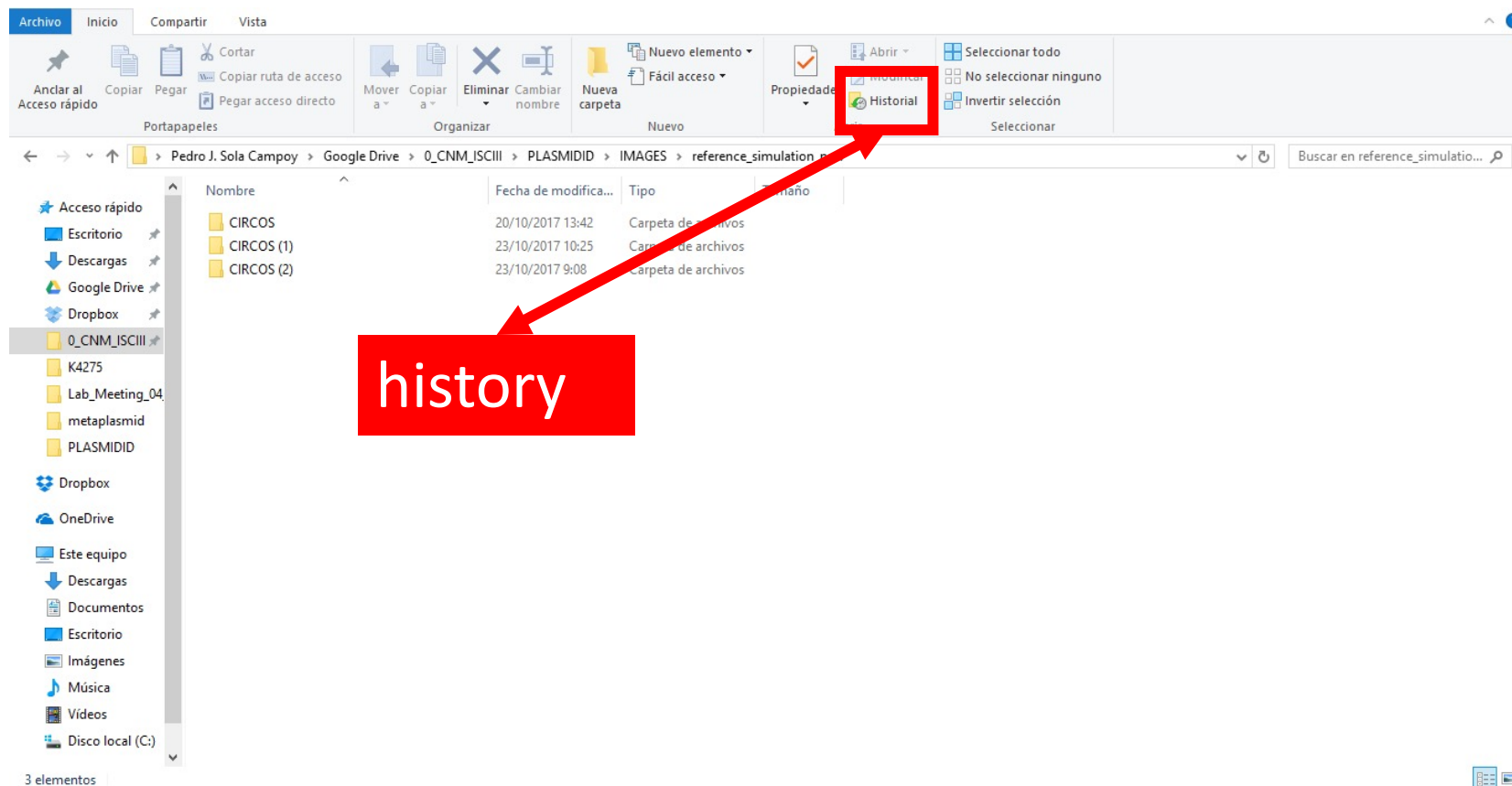
Nociones básicas



Nociones básicas



Nociones básicas

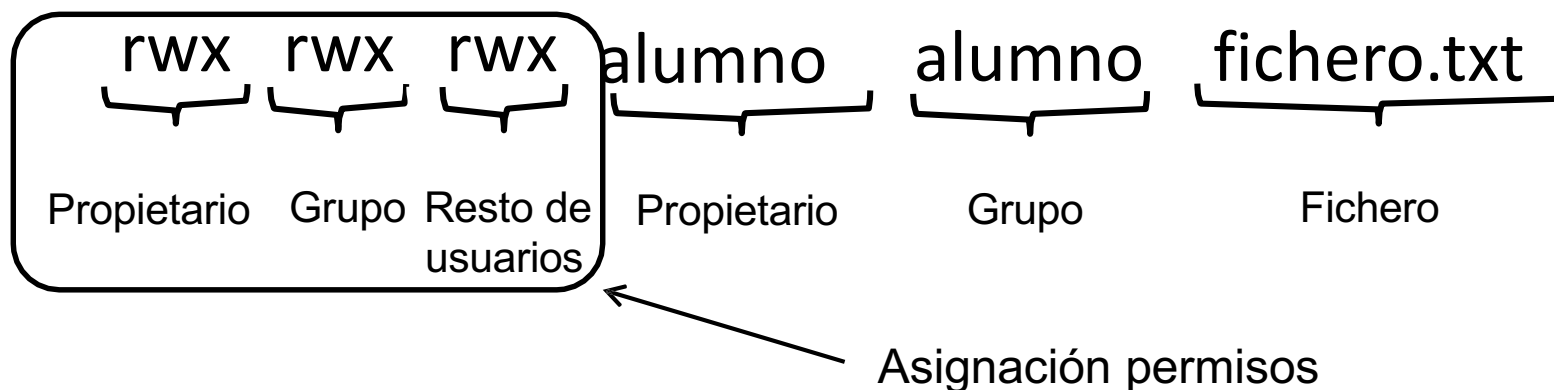


Permisos

- La forma de definir que puede y que no puede hacer un usuario es a través de los permisos de los ficheros.
- Cada fichero tiene asociados TRES tipos de permisos diferentes:
 - **Lectura (r):** permite ver el contenido del fichero
 - **Escritura (w):** permite alterar el contenido del fichero
 - **Ejecución (x):** permite ejecutar el fichero (en un directorio permite acceder a él)
- Se diferencian tres roles de usuarios respecto a ficheros y directorios:
 - Propietario
 - Miembro del grupo del fichero
 - Resto de usuarios

Permisos

- Los permisos de un fichero solo pueden ser alterados por su propietario, usuarios que pertenezcan al grupo y por el administrador
- El sistema almacena estos permisos como una secuencia de 9 bits
- Esta secuencia tiene una estructura en tres grupos: para el propietario, para el grupo del fichero y para los demás



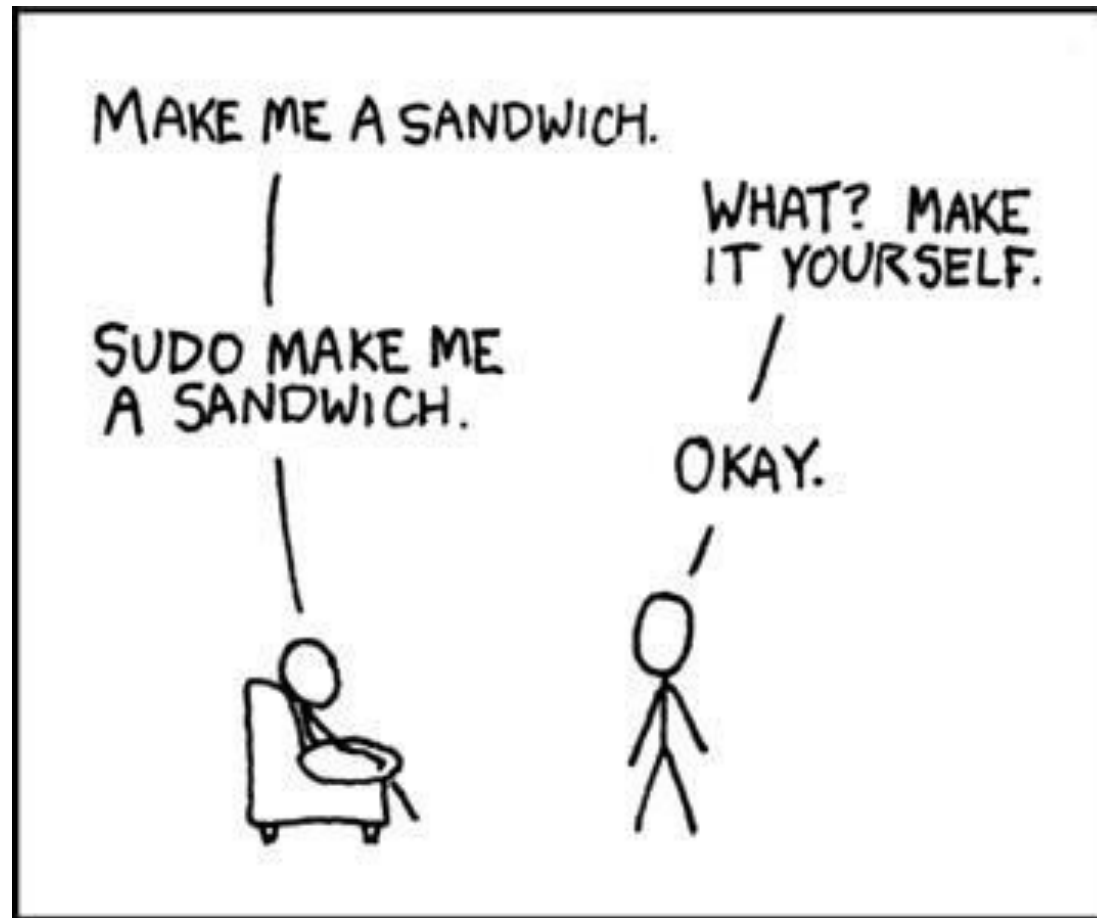
Permisos

- Ejemplo 1: **directorio opt**
 - **drwxr-xr-x root root opt**
 - El propietario puede leer, modificar y ejecutar el fichero.
 - Su grupo y el resto solo leerlo y ejecutarlo.
- Ejemplo 2: **directorio personal**
 - **drwx----- alumno clase alumno**
 - El propietario puede ver, modificar y acceder al contenido del directorio.
 - Su grupo y el resto no pueden hacer nada.
- Ejemplo 3: **directorio tmp**
 - **drwxrwxrwx root root tmp**
 - Todo el mundo puede ver, modificar y acceder al directorio.

Privilegios de Administrador

- Muchos de los comandos necesitan privilegios para poder ser ejecutados
- Para llamar o ejecutar un comando que requiere derechos de administrador tendremos que colocar antes de la instrucción el comando ***sudo*** (por ejemplo `$ sudo shutdown`). Después de ejecutar instrucciones con sudo el interprete nos pedirá la contraseña de administrador
- También podemos ejecutar el comando ***sudo su*** con el que conseguiremos derechos de root (administrador), evidentemente el interprete también nos pedirá la contraseña de administrador

Privilegios de Administrador

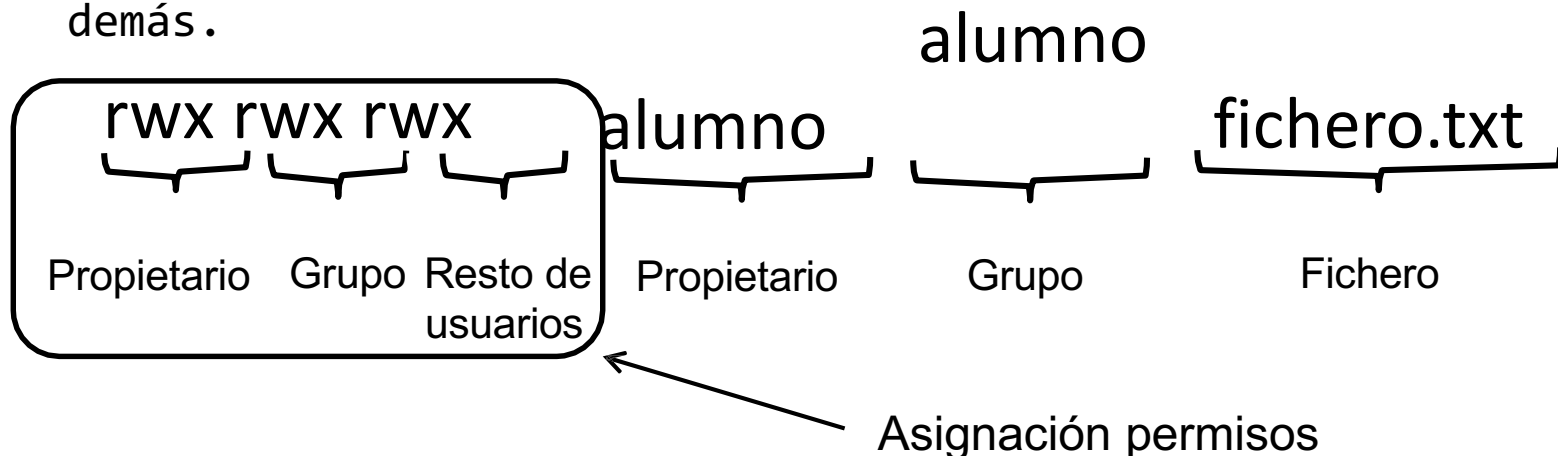


Administración ficheros y directorios

- **Asignación de permisos**
 - ☐ Comando `chmod`
- **Asignación de usuario y grupo propietario**
 - ☐ Comando `chown`
 - ☐ Comando `chgrp`

Asignación de permisos

- Los permisos de un fichero solo pueden ser alterados por su propietario, usuarios que pertenezcan al grupo y por el administrador
- El sistema almacena estos permisos como una secuencia de 9 bits.
- Esta secuencia tiene una estructura en tres grupos: para el propietario, para el grupo del fichero y para los demás.



Asignación de permisos

- Para la modificación de los permisos usamos **chmod**
- **Opciones:**
 - **-f** Forzar
 - **-R** Recursivo
- Suele usarse de dos modos:
 - **Modo textual.** Mas sencillo y lento.
 - **Modo octal.** Menos intuitivo pero mas rápido

Asignación de permisos

- Consiste en añadir(+) o eliminar(-) permisos
- A los diferentes actores: propietario(u), grupo(g) o los demás(o)
- **Sintaxis:** `chmod [opciones] [u|g|o|a] [+|=] [rwx] archivo`
- Ejemplos:
 - Quitar el permiso de lectura al grupo
 - `chmod g-r archivo`
 - Agregar permiso de ejecución al propietario y permiso de lectura para el grupo y otros
 - `chmod u+x,go+r archivo`
 - Agregar permiso de lectura y escritura a todos
 - `chmod a+rw archivo`

Asignación de usuario y grupo propietario

- **chown**
 - Permite cambiar el usuario o grupo propietario de un archivo o directorio
 - **Sintaxis:** `chown [opciones] usuario[:grupo] [archivo(s)]`
 - **Opciones:**
 - -R: Cambio recursivo
- **chgrp**
 - Permite cambiar el únicamente el grupo propietario de un archivo o
 - directorio
 - **Sintaxis:** `chgrp [opciones] grupo [archivo(s)]`
 - **Opciones:**
 - -R: Cambio recursivo

Gracias por su atención

