

RESUMEN ENTORNO (Unidades 2, 4, 5, 6, 7.1 y 7.2).

2) ARQUITECTURA DE UNA PC (hardware).

- ❖ **GABINETE / CASE:** carcasa que cubre y protege los componentes de la pc. Protege, organiza y refrigera.
- ❖ **PLACA MADRE / MAINBOARD:** pieza central de una computadora, encargada de conectar y comunicar todos los componentes esenciales del sistema. A través de sus conectores une la CPU, placa de video, etc
- ❖ **PSU / FUENTE DE ALIMENTACION:** se encarga de convertir la corriente que recibe de la toma eléctrica en los diferentes voltajes necesarios para alimentar los componentes internos de la computadora. Además, protege los componentes de la computadora de sobretensiones, cortocircuitos y otros problemas eléctricos que pueden ocurrir.
- ❖ **PROCESADOR / CPU:** es el componente principal de una computadora. Es un chip integrado que se coloca en el zócalo de la placa madre y está compuesto por varios núcleos (o Cores) que trabajan en conjunto. Es responsable de procesar y ejecutar los programas, manejar la entrada y salida de datos, y controlar los componentes del sistema.
- ❖ **MEMORIA RAM (Random Acces Memory):** tipo de memoria que se utiliza para almacenar temporalmente los datos y programas que están en uso. Es un componente clave en el rendimiento de una computadora, ya que proporciona un acceso rápido a los datos y programas que el procesador necesita para operar. Es una memoria volátil, lo que significa que pierde todos los datos almacenados en ella cuando se apaga la computadora.
- ❖ **MEMORIA SECUNDARIA:** HDD, SSD, M.2. Piezas encargadas de almacenar gran cantidad de datos, en discos magnéticos o en memorias de tipo flash, de esta ultima característica depende la velocidad de lectura y escritura del disco.



- ❖ **PLACA DE VIDEO / GPU** : componente de hardware que tiene como objetivo procesar y generar imágenes en la pantalla. Su función es liberar a la CPU de la tarea del procesamiento gráfico, permitiéndole concentrarse en otras tareas.
- ❖ **PERIFERICOS VARIOS.**

4) CONCEPTOS DE PROGRAMACION.

La programación es el proceso de diseñar, escribir, probar y mantener el código informático. En términos más simples, la programación es la forma en que los desarrolladores crean software, aplicaciones y sistemas.

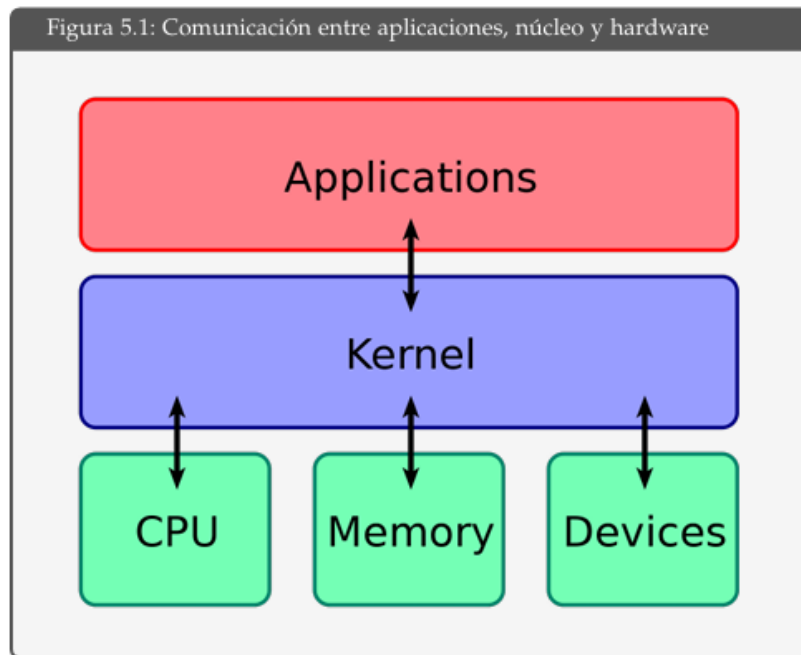
- ❖ **LENGUAJE DE PROGRAMACION:** Un lenguaje de programación es un conjunto de reglas, símbolos y palabras clave que se utilizan para comunicar instrucciones a una computadora. Es una herramienta para crear software, permitiendo transformar ideas en código ejecutable.
- ❖ **NIVELES DE LENGUAJES:** existen de 3 tipos, ALTO, BAJO Y BINARIO.
 - **ALTO:** son más cercanos al lenguaje natural humano y se utilizan para escribir programas complejos de manera más fácil y rápida, abstrayéndose de muchos detalles. (PYTHON, JAVA, C++).
 - **BAJO:** son cercanos al lenguaje de la máquina y se utilizan para escribir programas que interactúan directamente con el hardware de la computadora. Requiere mayor conocimiento en el hardware y memoria.
 - **BINARIO:** utilizado por computadoras para ejecutar programas. Instrucciones escritas en 0 y 1 para que el procesador entienda y ejecute directamente.
- ❖ **COMPILADORES E INTERPRETES:** Un intérprete y un compilador son dos tipos de programas que se utilizan para convertir el código fuente escrito por un programador en instrucciones ejecutables por una computadora.
 - **INTERPRETE:** programa que lee el código fuente de un programa y lo traduce en instrucciones ejecutables en tiempo real. El intérprete lee una línea de código fuente, la traduce a lenguaje de máquina y la ejecuta antes de pasar a la siguiente línea.
 - **COMPILADOR:** programa que traduce todo el código fuente a lenguaje de máquina en un solo paso, generando un archivo ejecutable que puede ser utilizado por la computadora sin necesidad de leer el código original nuevamente. El compilador lee todo el código fuente del programa y lo traduce en un archivo ejecutable que puede ser utilizado por la computadora sin necesidad de leer el código fuente original nuevamente.
- ❖ **OTROS CONCEPTOS:**
 - **CODIGO FUENTE:** el código fuente de un programa es el conjunto de instrucciones escritas por un programador en un lenguaje de programación determinado
 - **CODIGO ABIERTO:** refiere a un tipo de software en el cual el código fuente está disponible públicamente y puede ser modificado por cualquier persona.
 - **SOFTWARE LIBRE:** El software libre es aquel que se distribuye con una licencia que permite a los usuarios tener acceso al código fuente del programa y redistribuir versiones modificadas del mismo sin tener que pagar regalías o cargos adicionales.

En general, dicha licencia establece los términos y condiciones bajo los cuales se puede usar, modificar y distribuir el software.

- **IDE:** conjunto de herramientas (editores de textos y depuradores) y características que se combinan en una única aplicación para hacer más fácil y eficiente el proceso de desarrollo de software.

5) SISTEMAS OPERATIVOS.

Conjunto de programas y herramientas que controlan y coordinan las actividades de una computadora o dispositivo electrónico, y permiten a los usuarios interactuar con el hardware y el software de manera sencilla y eficiente. Está compuesto por un núcleo (o kernel) que tiene control total sobre el hardware en el que corre, y una serie de programas utilitarios que se comunican con él.



❖ FUNCIONALIDADES PRINCIPALES DE UN SISTEMA OPERATIVO.

- **ABSTRACCION DEL HARDWARE:** el sistema operativo oculta los detalles específicos del hardware de la computadora a las aplicaciones y los usuarios. Esto permite a las aplicaciones interactuar con el hardware de manera consistente y simplificada, sin tener que preocuparse por los detalles internos del hardware subyacente.
- **GESTION DEL HARDWARE:** el sistema operativo es responsable de gestionar el hardware. Controla cómo se utilizan estos recursos (componentes de la pc) y asigna la cantidad adecuada de memoria y procesador a cada aplicación.
- **INTERFAZ DE USUARIO:** proporciona una interfaz de usuario que permite a los usuarios interactuar con la computadora y ejecutar aplicaciones.
- **GESTION DE ARCHIVOS:** se encarga de gestionar los archivos y carpetas, permitiendo a los usuarios crearlos, modificarlos, copiarlos y eliminarlos.

- **MULTITAREA:** permite que varias aplicaciones se ejecuten al mismo tiempo y asigna los recursos necesarios para que funcionen correctamente, dando la ilusión de paralelismo.
- **MULTIUSUARIO:** un sistema operativo puede ser utilizado por varios grupos al mismo tiempo y garantiza que cada usuario tenga sus propios archivos.

❖ **NUCLEO.**

El kernel del sistema (o núcleo), es la parte central y más fundamental del mismo. Responsable de controlar el acceso a los recursos del hardware, gestionar los procesos, etc. Proporciona una interfaz para que las aplicaciones interactúen con el hardware del sistema.

Se ejecuta en modo privilegiado, lo que significa que tiene acceso directo al hardware y puede ejecutar instrucciones que otros programas no pueden.

Para casi cualquier tarea las aplicaciones de usuario necesitan pedirle permiso al kernel, mediante una instrucción denominada «*llamada a sistema*». Cuando ocurre una llamada a sistema, el CPU detiene la ejecución del programa y comienza a ejecutar la funcionalidad del núcleo requerida, luego de la cual se reanuda el flujo original del programa.

❖ **PROCESO DE ARRANQUE.**

Durante el arranque, ocurren varios pasos importantes que permiten la correcta ejecución del sistema operativo y el pc esté listo para su uso.

Al presionar el botón de arranque se enciende la fuente de alimentación, suministrando la energía necesaria para que la placa madre comience a funcionar.

- **POST (Power on Self Test):** autodiagnóstico realizado por la placa madre para corroborar que todos los componentes estén funcionando correctamente. Si esto ultimo no sucede, se emite un mensaje de error y se detiene el proceso de arranque.
- **BOOT LOADER (Cargador de Arranque):** programa ejecutado con el fin de cargar el kernel del sistema (*GRUB* y *bootmgr*).
- **KERNEL:** toma el control. Inicialmente se encarga de identificar el hardware disponible, cargar los drivers necesarios y montar el sistema de archivos del sistema. Finalmente da inicio al primer programa de usuario, a partir del cual se ejecutan todos los demás.
- **INIT:** programa inicial de los sistemas de *LINUX*. Encargado de cargar los scripts de arranque del sistema, ejecutar los servicios esenciales para el funcionamiento de este y proveer al usuario de un entorno grafico o de línea de comandos.

❖ **INTERFAZ DEL USUARIO (UI).**

Punto de interacción y comunicación entre un ser humano y un programa. Una buena interfaz es intuitiva y facilita la interacción.

Existen 3 tipos:

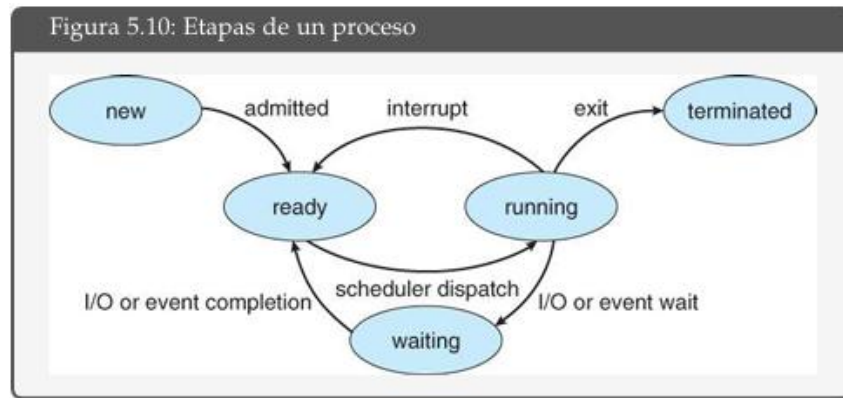
- **GUI:** forma de interactuar con un programa o un sistema operativo mediante el uso de elementos gráficos. Faciles de aprender y utilizar.

- **CLI:** interfaz basada en texto, donde el usuario interactúa con el sistema escribiendo comandos específicos en una línea de comandos. Requieren mayor conocimiento técnico, pero consumen menos recursos del sistema.
- **TUI:** interfaz híbrida que combina las 2 anteriores. Se ejecuta en un entorno de texto, pero utiliza caracteres y colores para crear una apariencia más visual y estructurada.
- ❖ **TERMINALES/ CONSOLA/ EMULADOR/ INTERPRETE.**
 - **TERMINAL:** refiere a él o los dispositivos físicos que se utilizan para interactuar con un ordenador mediante la entrada y salida de texto. En la actualidad está compuesta principalmente por el teclado y el monitor.
 - **CONSOLA VIRTUAL:** aplicación implementada dentro del kernel que provee acceso al sistema simulando una terminal de teletipo.
 - **EMULADOR DE TERMINAL:** programa de usuario que permite interactuar con un sistema operativo a través de una ventana en un entorno gráfico. Comúnmente usados para acceder a sistemas remotos o para ejecutar aplicaciones en línea de comandos en sistemas operativos.
 - **SHELL (interprete de línea de comandos):** programa que permite al usuario interactuar con el sistema operativos mediante la ejecución de comandos a través de una interfaz de línea de comandos.

❖ **PROCESOS.**

Un proceso es una instancia de un programa en ejecución. Durante su ciclo de vida, atraviesa diferentes etapas las cuales incluyen:

- **CREACION:** el sistema operativo crea un nuevo proceso cuando se inicia un programa o se solicita mediante una llamada al sistema. Durante la creación se asignan recursos como espacio de memoria, identificadores de procesos, etc.
- **LISTO:** el proceso está cargado en memoria la espera de ser seleccionado por el planificador del sistema operativo para su ejecución
- **EJECUCION:** es seleccionado por el planificador del sistema. El procesador ejecuta las instrucciones del programa y realiza las tareas definidas por el proceso. El proceso continúa en este estado hasta que le toque ejecutar a otro programa o se bloquee esperando alguna operación de entrada/salida.
- **BLOQUEADO:** un proceso se encuentra en este estado si necesita esperar por alguna operación de entrada/salida. Cuando esto ocurre el proceso se suspende temporalmente y se libera el uso del procesador.
- **FINALIZACION:** cuando un proceso ha terminado de ejecutar todas sus instrucciones (aunque sea de manera forzada), entra en estado de finalización, liberando los recursos asignados realizando la limpieza necesaria.



❖ DISTRIBUCIONES.

Las distribuciones de Linux son sistemas operativos basados en el kernel de este, compuestos por una combinación de software libre y de código abierto (*Debian, Ubuntu, Red Hat, etc*). Existen porque el software de código abierto permite a los usuarios y desarrolladores acceder, modificar y distribuir el código fuente del software. Además, al ser un sistema operativo altamente personalizable y adaptable, cada distribución puede estar diseñada para satisfacer las necesidades específicas de diferentes usuarios, como por ejemplo para usuarios de servidores, programadores, usuarios de escritorio, entre otros.

❖ PROCESO DE APAGADO.

Antes de finalizar la ejecución del sistema operativo, se inicia un proceso que cierra todos los programas y servicios que se están ejecutando. Luego, se guardan todos los datos pendientes y se asegura que todos los dispositivos de almacenamiento estén en un estado seguro antes de apagar la alimentación. Finalmente, se envía una señal al hardware para que se apague por completo y se desconecte la alimentación en caso de ser necesario.

Hay varios métodos para “apagar” una computadora: apagar, reiniciar, suspender e hibernar.

6) SISTEMAS DE ARCHIVOS.

❖ ACCESO SECUENCIAL Y ALEATORIO.

- Un dispositivo de **acceso aleatorio** permite el acceso directo a cualquier ubicación en el medio de almacenamiento.
- Por otro lado, un dispositivo de **acceso secuencial** requiere que los datos se lean en secuencia, desde el principio del medio de almacenamiento hasta el final. Esto significa que, para acceder a un archivo o dato específico, el dispositivo debe leer los datos almacenados antes de llegar al archivo o dato deseado.

❖ DESCRIPCION.

Un sistema de archivos es una estructura lógica y organizada de datos que se utiliza para almacenar y recuperar información en un medio de almacenamiento. Sin un sistema de

archivos, los datos colocados en un medio de almacenamiento serían un gran cuerpo de datos sin manera de saber dónde termina un dato y comienza el siguiente (ext4, NTFS).

Sus principales funciones son:

- **GESTION DEL ESPACIO.**
- **ORGANIZACIÓN.**
- **PERMISOS.**
- **METADATOS.**
- **INTEGRIDAD.**

❖ **PARTICIONES.**

Sección lógica de un disco duro u otro dispositivo de almacenamiento que se trata como si fuera un disco separado.

Las particiones se utilizan para varios fines, como separar el sistema operativo y los archivos del usuario, o para crear diferentes áreas de almacenamiento para diferentes propósitos.

❖ **ESTRUCTURA DE DIRECTORIOS.**

La jerarquía de directorios en un sistema de archivos se llama árbol de directorios. El nivel superior del árbol de directorios se llama directorio raíz, que contiene todos los demás directorios y archivos en el sistema de archivos. A medida que se desciende por el árbol de directorios, los nombres de los directorios y los archivos se combinan para formar rutas de acceso a los archivos y directorios individuales.

Para navegar a través de los directorios del sistema en un entorno gráfico, utilizamos un programa denominado «administrador de archivos».

Existen dos tipos de rutas que se utilizan para referirse a la ubicación de un archivo o directorio en un sistema de archivos: *rutas absolutas* y *rutas relativas*.

- **RUTAS ABSOLUTAS:** especifica la ubicación completa de un archivo o directorio, desde el directorio raíz hasta el archivo o directorio en cuestión. Describe la ubicación exacta de un archivo o directorio. Siempre comienza con “/”.
- **RUTAS RELATIVAS:** describe la ubicación de un archivo o directorio en relación con el directorio de trabajo actual. En otras palabras, una ruta relativa es una ruta que se especifica desde el directorio actual hasta el archivo o directorio en cuestión.

❖ **EL SISTEMA DE ARCHIVOS DE LINUX.**

➤ **ESQUEMA DE PARTICIONADO.**

Linux puede instalarse en una variedad de esquemas de particionado, pero uno de los más comunes es el esquema de particionado convencional, que utiliza tres particiones para el sistema:

- Una partición en donde se instala el gestor de arranque y el núcleo del sistema operativo.
- Una partición destinada a almacenar los programas de usuario y documentos.
- Una partición de intercambio, reservada para ser utilizada cuando la memoria RAM se agota.

➤ **NOMECLATURA DE DISPOSITIVOS:**

En Linux, los discos duros y particiones se nombran usando un esquema de nomenclatura estandarizado. Cada dispositivo se identifica por un nombre que se genera automáticamente y que puede estar compuesto de diferentes elementos.

El nombre asignado a un dispositivo de almacenamiento consta de tres partes:

- Una sigla identificadora del tipo de dispositivo. En general, los discos rígidos, unidades de estado sólido y dispositivos USB empiezan con las letras «sd».
- Una letra secuencial que identifica el dispositivo en relación con otros dispositivos del mismo tipo. El primer dispositivo se denominará «sda», el segundo «sdb» y así sucesivamente.
- Un número adicional que indica la partición específica del dispositivo. La primera partición en el primer dispositivo será «sda1», la segunda será «sda2», y así sucesivamente.

➤ JERARQUIA DE ARCHIVOS.

La jerarquía de archivos en el sistema de archivos se organiza en un árbol con una estructura específica. El objetivo de una estructura estandarizada es que los desarrolladores de software y los administradores de sistemas puedan predecir con precisión dónde se encuentran los archivos importantes del sistema y cómo acceder a ellos. A continuación, se presentan algunos de los directorios más importantes de la jerarquía de archivos:

- **/** Es la raíz del sistema de archivos. Cualquier otro directorio debe encontrarse dentro de él, independientemente de en cuál dispositivo se encuentre.
- **/bin** Este directorio contiene los programas del sistema, como comandos básicos de GNU/Linux.
- **/boot** Contiene los archivos necesarios para el proceso de arranque del sistema, como el gestor de arranque o el núcleo.
- **/dev** En Linux, los dispositivos del hardware se representan como archivos especiales. En esta carpeta se encuentran esos archivos.
- **/etc** Este directorio contiene los archivos de configuración del sistema y de las aplicaciones instaladas en el mismo.
- **/home** Este directorio se utiliza para almacenar los directorios personales de los usuarios.
- **/lib** Contiene las bibliotecas compartidas necesarias para el funcionamiento del sistema.
- **/mnt** Este directorio se utiliza para montar sistemas de archivos externos, como unidades flash USB, discos duros externos, etc. En algunas distribuciones esto también se hace en el directorio «/media».
- **/proc** El directorio «/proc» es un directorio virtual que contiene información sobre el sistema y los procesos en ejecución. Es generado en tiempo real por el kernel del sistema operativo.
- **/root** Este directorio es el directorio personal del usuario «root». El usuario root es el único que tiene acceso completo al sistema.
- **/sbin** Contiene los programas utilizados principalmente para la administración del sistema.

- **/tmp** Este directorio se utiliza para almacenar archivos temporales creados por el sistema y los usuarios.
- **/usr** Contiene la mayoría de las utilidades y aplicaciones multiusuario, es decir, accesibles para todos los usuarios. En otras palabras, contiene los archivos compartidos, pero que no obstante son de sólo lectura.
- **/var** Este directorio contiene archivos que cambian frecuentemente durante la operación del sistema, como archivos de registro y archivos de bases de datos.
- **/opt** Se utiliza para almacenar «paquetes adicionales» o software adicional que no es parte de la distribución principal del sistema operativo.

Todos aquellos archivos y carpetas que en su nombre comiencen con un punto, son considerados archivos ocultos.

➤ **CARPETAS ESPECIALES.**

Existen tres tipos de carpetas especiales en los sistemas Linux:

- El punto (.) es una carpeta especial que representa el directorio actual en el que se encuentra el usuario.
- Los dos puntos (..) representan el directorio padre del directorio actual.
- El símbolo de tilde (~) representa el directorio principal del usuario.

➤ **MONTAJE.**

«Montar» significa hacer que un sistema de archivos esté disponible en un punto específico del sistema de archivos raíz. En otras palabras, montar un sistema de archivos implica hacer que el sistema operativo sea consciente de la existencia de un sistema de archivos y asignarlo a un directorio específico en la jerarquía de archivos del sistema.

El punto de montaje es un directorio en el sistema de archivos raíz, que se utiliza para acceder al contenido del sistema de archivos que se va a montar. Una vez montado, el contenido del sistema de archivos se vuelve accesible en el punto de montaje especificado.

➤ **ENLACES.**

En Linux, existen dos tipos de enlaces para archivos:

- **ENLACES SUAVES:** archivo especial que apunta a otro archivo en el sistema de archivos. El enlace suave no contiene los datos del archivo al que apunta, sino simplemente la ruta del archivo. Si se elimina el archivo original, el enlace suave se vuelve inútil. Se utilizan comúnmente para hacer referencia a archivos en diferentes ubicaciones y para crear accesos directos en el sistema de archivos.
- **ENLACES DUROS:** entrada de directorio adicional que apunta al mismo archivo en el sistema de archivos. A diferencia de los enlaces suaves, los enlaces duros no son archivos independientes y no pueden apuntar a archivos en diferentes sistemas de archivos o particiones. Si se elimina el archivo original, el enlace duro sigue siendo válido y no se elimina hasta que todos los enlaces se eliminan.

7) COMANDOS BASICOS.

- **man:** herramienta de línea de comandos que proporciona información detallada sobre otros comandos, funciones y bibliotecas.

```
man [opciones] comando
```

- «-f» para buscar descripciones cortas de comandos relacionados con una palabra clave.
- «-k» para buscar páginas del manual que contienen una palabra clave específica.
- **clear:** se utiliza para limpiar la pantalla del terminal, eliminando todos los comandos y resultados previos que se hayan mostrado.

```
clear [-x]
```

- **echo:** se utiliza para imprimir texto en la pantalla del terminal.

```
echo [opciones] texto
```

También admite varias opciones y parámetros adicionales, que permiten ajustar el formato de la salida y realizar otras tareas:

- «-n»: evita que el comando agregue un carácter de nueva línea al final de la salida. En otras palabras, la siguiente línea del terminal comenzará en la misma línea en la que termina la salida del comando.
- «-e»: activa el soporte de secuencias de escape en la salida. Por ejemplo, se pueden utilizar secuencias de escape para agregar saltos de línea al texto.
- **pwd:** muestra la ruta completa del directorio de trabajo actual en el sistema de archivos.

```
pwd [-P]
```

Puede agregarse la opción «-P» para mostrar el directorio sin enlaces simbólicos.

- **cd:** se utiliza para cambiar el directorio de trabajo actual en el sistema de archivos.

```
cd [ruta]
```

- Si no se especifica una ruta, el comando es equivalente a «cd ~».
- Si se utiliza el argumento «-», el comando «cd» vuelve al directorio de trabajo anterior.

- **ls:** se utiliza para listar los archivos y directorios en el directorio de trabajo actual o en una ubicación específica del sistema de archivos.

```
ls [ruta]
```

Si se omite la ruta, se asume que se desea listar los archivos y directorios en el directorio de trabajo actual. Además, se pueden utilizar opciones para personalizar la salida del comando «ls». Algunas opciones comunes incluyen:

- «-l»: muestra la lista en formato largo, que incluye información detallada sobre cada archivo o directorio.
 - «-h»: muestra el tamaño de los archivos en formato legible por humanos, como «KB» o «MB».
 - «-a»: muestra todos los archivos y directorios, incluyendo los que comienzan con un punto
- **tree:** en sistemas Unix y Linux es una herramienta de línea de comandos que muestra la estructura de directorios y archivos en forma de árbol. Su uso básico es bastante simple, y puede proporcionar una vista jerárquica clara de la organización de archivos en un directorio y sus subdirectorios.

```
tree [opciones] [directorio]
```

- «-d»: Muestra solo los directorios, no los archivos.
 - «-L nivel»: Limita la profundidad del árbol al nivel especificado.
 - «-a»: Muestra archivos y directorios ocultos.
 - «-f»: Muestra la ruta completa de cada archivo.
- **mkdir:** se utiliza para crear nuevos directorios en el sistema de archivos.

```
mkdir [opciones] directorio ...
```

- «-p»: crea todos los directorios en la ruta especificada, incluso si no existen. Si ya existe un directorio con el mismo nombre que uno de los directorios de la ruta, se ignora sin dar errores.
 - «-v»: muestra los detalles de la creación del directorio en la salida estándar de la terminal.
- **rmdir:** utilizado para eliminar directorios vacíos en el sistema de archivos.

```
rmdir [opciones] directorio...
```

- «-p»: dada una ruta, elimina un directorio y sus predecesores.
 - «-v»: muestra los detalles de la eliminación del directorio en la salida estándar de la terminal.
-
- **touch**: se utiliza para crear archivos vacíos o actualizar la fecha y hora de acceso o modificación de un archivo.

```
touch [opciones] archivo...
```

- «-a»: actualiza sólo la fecha y hora de acceso del archivo.
 - «-m»: actualiza sólo la fecha y hora de modificación del archivo.
-
- **rm**: utilizado para eliminar archivos y directorios en el sistema de archivos.

```
rm [opciones] archivo ...
```

«-r»: se utiliza para eliminar directorios y su contenido de forma recursiva.

«-f»: se utiliza para forzar la eliminación de un archivo si este no tiene permiso de escritura. Además, no da error si los archivos no existen.

«-i»: pide una confirmación antes de borrar cada archivo.

- **cp**: se utiliza para copiar uno o más archivos o directorios de una ubicación a otra en el sistema de archivos.

```
cp [opciones] origen destino
```

- «-r»: indica que se deben copiar los directorios y su contenido de forma recursiva.
- «-v»: muestra los detalles de la copia en la salida estándar de la terminal.
- «-i»: pide confirmación antes de sobrescribir archivos existentes.

- **mv:** se utiliza para mover o renombrar archivos o directorios.

```
mv [opciones] origen destino
```

- «-i»: Solicita confirmación antes de sobrescribir archivos existentes.
 - «-f»: Sobrescribe archivos existentes sin preguntar.
 - «-v»: Muestra información detallada sobre lo que se está haciendo.
-
- **ln:** se utiliza en sistemas Linux para crear enlaces simbólicos o duros a archivos y directorios.

```
ln [-s] [archivo original] [nombre del enlace]
```

Por defecto «ln» crea enlaces duros, a no ser que se utilice la opción «-s».

8) COMANDOS AVANZADOS.

- **sudo:** con «sudo», un usuario común puede obtener temporariamente los permisos de un usuario con otros privilegios, como el superusuario (root), para realizar tareas que requieren permisos especiales.

```
sudo [-u USUARIO] comando
```

Si no se especifica un USUARIO, se obtienen los privilegios del usuario root.

- **passwd:** utilizado para cambiar la contraseña de un usuario.

```
passwd [opciones] [usuario]
```

Si se ejecuta sin argumentos, se cambiará la contraseña del usuario actual. El comando pedirá al usuario que introduzca la contraseña actual y, a continuación, que introduzca la nueva contraseña dos veces para confirmarla.

- «-l»: Bloquea la cuenta del usuario. Esto significa que el usuario no podrá iniciar sesión hasta que la cuenta sea desbloqueada.
- «-u»: Desbloquea una cuenta que ha sido bloqueada.
- «-d»: Elimina la contraseña del usuario. Esto permite que el usuario inicie sesión sin una contraseña.
- «-e»: Hace que la contraseña del usuario expire. Esto significa que el usuario deberá cambiar su contraseña la próxima vez que inicie sesión.

- **chown:** se utiliza para cambiar el propietario y/o grupo de un archivo o directorio.

```
chown [-R] usuario[:grupo] archivo
```

Puede agregarse la opción «-R» para operar recursivamente sobre los directorios.

- **chmod:** se utiliza para cambiar los permisos de acceso a archivos y directorios.

```
chmod [-R] modo archivo
```

Puede agregarse la opción «-R» para operar recursivamente sobre los directorios.

El modo puede especificarse en forma numérica o usando caracteres especiales. La sintaxis básica del modo simbólico es:

```
[ugoa] [+ -=] [rwx]
```

- «u»: representa al usuario propietario del archivo/directorio.
- «g»: representa al grupo al que pertenece el archivo/directorio.
- «o»: representa a otros usuarios que no son el propietario ni pertenecen al grupo.
- «a»: representa a todos los usuarios.
- «+»: agrega los permisos especificados al archivo/directorio.
- «-»: elimina los permisos especificados del archivo/directorio.

- «=»: establece los permisos especificados y elimina cualquier otro permiso que no sea el especificado.
- **useradd**: utilizado para crear una nueva cuenta de usuario en el sistema.

```
useradd [opciones] usuario
```

- «-m»: Crea automáticamente el directorio personal del usuario en «/home/usuario».
- «-g GRUPO»: Setea el GRUPO al que pertenece el usuario.

- **userdel**: se utiliza para eliminar una cuenta de usuario existente.

```
userdel [opciones] usuario
```

- «-r»: esta opción se utiliza para eliminar la cuenta de usuario y sus archivos personales.
- «-f»: esta opción se utiliza para forzar la eliminación de la cuenta de usuario, incluso si está actualmente en uso o si tiene procesos en ejecución.
- **usermod**: El comando «usermod» se utiliza para realizar cambios en la información de un usuario existente en el sistema. Puede utilizarse para modificar el nombre de usuario, el directorio de inicio, el grupo principal, los grupos secundarios, la información del usuario, entre otros aspectos relacionados con la configuración del usuario.

```
usermod [opciones] usuario
```


- «-l NOMBRE»: Permite cambiar el NOMBRE de usuario.
- «-d DIRECTORIO»: Permite cambiar el DIRECTORIO de personal del usuario.
(La ruta al directorio debe ser una **ruta absoluta**)
- «-m»: Mueve el contenido del directorio personal a una nueva ubicación. Sólo válida si se usa con «-d».
- «-g GRUPO»: Permite cambiar el GRUPO principal del usuario.
- «-G GRUPOS»: Permite establecer los GRUPOS a los que pertenece el usuario.
- «-a GRUPO»: Permite añadir un GRUPO secundario al usuario. Sólo válida si se usa con «-G».
- «-s SHELL»: Permite cambiar el SHELL del usuario

PERMISOS

Los **permisos** en Linux son utilizados para proteger los archivos y directorios del sistema de accesos no autorizado. Cada archivo y directorio es propiedad de un usuario y un grupo; y el propietario del archivo o directorio puede especificar quién tiene acceso a ellos y en qué nivel.

Existen tres tipos de permisos: de usuario (**u**), de grupo (**g**) y otros (**o**), y cada uno de ellos puede tener tres permisos posibles: lectura (**r**), escritura (**w**) y ejecución (**x**).

- El permiso de lectura (**r**) permite al usuario ver el contenido de un archivo o listar los archivos de un directorio.
- El permiso de escritura (**w**) permite al usuario modificar el contenido de un archivo o crear, eliminar y renombrar archivos dentro de un directorio.
- El permiso de ejecución (**x**) permite al usuario ejecutar un archivo o entrar en un directorio.

 Salida en pantalla

```
-rwxr-xr-x 1 alumno wheel 15832 abr 12 19:33 dsh*  
-rw-r--r-- 1 alumno wheel 1130 abr 12 19:32 dsh.c
```

En ocasiones estos permisos se representan mediante un número de 3 dígitos en base 8, donde cada dígito representa los permisos para un grupo diferente de usuarios: el propietario del archivo, el grupo al que pertenece el archivo y otros usuarios.

Los valores numéricos correspondientes a cada permiso son:

- **4** para el permiso de lectura (r).
- **2** para el permiso de escritura (w).
- **1** para el permiso de ejecución (x).