1. Linux

GNU/Linux es el resultado de combinar el kernel Linux, el cual es el núcleo encargado de gestionar el hardware, con las herramientas y utilidades del proyecto GNU, las cuales proporcionan funcionalidades esenciales para que este sistema operativo sea utilizable, Por esta razón, se le denomina GNU/Linux.
Las distribuciones de GNU/Linux son sistemas operativos completos que se basan en el kernel de Linux. Estas incluyen herramientas de GNU y otros componentes adicionales, como gestores de paquetes, entornos gráficos y aplicaciones específicas. Estas personalizaciones permiten optimizar cada distribución para diferentes nichos o usos. Dentro de las distribuciones se encuentran:

- Debian: Conocida por su estabilidad y uso en servidores.
- Slackware: Orientada a usuarios avanzados que buscan entornos minimalistas
- Arch Linux: Enfocada en la simplicidad y personalización.
- Red Hat Enterprise Linux (RHEL): Orientada a entornos empresariales y servidores.
- Ubuntu: Es una distribución basada en Debian.

Estructura de Linux

El sistema de ficheros en Linux es una estructura jerárquica que organiza los datos y recursos del sistema. Dando inicio desde la raíz, que es representada como `/`. Desde la raíz se despliegan varios directorios que alojan distintos tipos de datos y configuraciones. Estos ficheros pueden estar contenidos en particiones, básicamente son secciones dentro de la unidad de almacenamiento donde se instaló el sistema que están separadas una de la otra lo que permite que se pueda organizar mucho más la información.
Linux soporta varios tipos de sistemas de ficheros a nivel de formato como:

- EXT4(Extended Filesystem 4): Es uno de los más utilizados por fiabilidad y rendimiento.
- XFS: Optimizados para manejar grandes volúmenes de datos, principalmente ideal para entornos empresariales.
- BTRFS: Esta diseñado para snapshots, alta escalabilidad y características avanzadas como compresión integrada.
  Los elementos clave de un sistema de ficheros son:
- Inodos: Almacena información de metadatos (permisos, propietario, tamaño, etc) sobre un fichero o directorio.
- Bloques: Son unidades de almacenamiento donde se guardan los datos de los ficheros.

- Super Bloque: Contiene información crucial sobre el sistema de ficheros. Si el superbloque se daña, el sistema puede ser difícil de recuperar.
  En Linux existen 2 tipos de enlaces:
- Enlaces duros: Son referencias directas al mismo inodo, esto significa que apuntan al mismo contenido del fichero. Si el fichero es eliminado, el contenido seguirá accesible mientras existan enlaces duros.
- Enlaces blandos o simbólicos: Apunta a la ruta del fichero en lugar del contenido, Si el fichero original es eliminado, el enlace simbólico se rompe.

¿Qué es un comando?

Un comando es una instrucción que el usuario ingresa a través de la terminal para que le sistema realice una tarea específica. Los comandos son interpretados por la shell, que traduce estas ordenes en acciones ejecutables por el sistema. Cada comando puede estar asociado con un conjunto de instrucciones predefinidas que al ser ejecutadas, permiten al usuario interactuar con el sistema y realizar operaciones como gestionar archivos, consultar información del sistema o ejecutar programas.

Shell vs Emulador de terminal

Una terminal es simplemente el software que provee a la interfaz que permite la entrada y salida de datos dentro de la cual se puede ejecutar una Shell. En el pasado la única manera de poder darle instrucciones al sistema operativo era por medio de una terminal, en la actualidad las cosas han cambiado, ahora todos los sistemas operativos de uso común cuentan con una interfaz gráfica que les permite ser más aptos para todo público ya que entre más fácil sea un sistema operativo más personas pueden llegar a usarlo. Por otro lado, los emuladores de terminales básicamente emulan una terminal física y dentro de esta es donde se puede ejecutar una Shell.
Una Shell es un programa cuya función principal es la interpretación de comandos, a través de dicha interpretación de comandos usa los recursos del sistema para poder lograr la acción deseada por el usuario, se podría decir que la Shell es un puente entre el usuario y el sistema operativo, para que a través de ella se le pueda dar órdenes al sistema, las más comunes son:

- sh: Es la shell básica, generalmente incluida en la gran mayoría de sistemas UNIX.
- bash: Es una mejora de la sh, con características adicionales como historial de comandos, autocompletado y scripting avanzado.
- zsh: Es una shell moderna que combina características de bash y otras shells como ksh, con mejoras en la interactividad, autocompletado avanzado y plugins.

Introducción al prompt en Linux

Comúnmente se puede referir a él como el indicador de comandos o el indicador que estamos en una Shell dentro del emulador de terminal. La estructura básica de bash es la siguiente: usuario@host:ruta$

- usuario El nombre del usuario activo.
- @: Separa el nombre del usuario del nombre del host.
- host: El nombre del sistema o máquina.
- : : Separa el nombre del host de la ruta.
- ruta: El directorio actual en el que se encuentra el usuario.
- $ o # : El carácter final que indica el nivel de permisos:
  - $ para usuarios normales.
  - # para el superusuario (root).

Directorios importantes en Linux

El sistema de archivos en Linux sigue una estructura jerárquica con una raíz única representada por /. Desde esta raíz, se organizan los diferentes directorios del sistema.

```
/
├── boot/ - Archivos necesarios para el arranque del sistema (kernel, GRUB, etc.).
├── etc/ - Archivos de configuración del sistema y aplicaciones.
├── var/ - Archivos de datos variables (logs, bases de datos, colas de impresión, etc.).
│   ├── log/ - Archivos de registro del sistema.
│   ├── spool/ - Colas de impresión y correo.
│   └── tmp/ - Archivos temporales persistentes.
├── usr/ - Archivos compartidos por los usuarios y aplicaciones.
│   ├── bin/ - Ejecutables de aplicaciones comunes.
│   └── lib/ - Bibliotecas compartidas.
├── home/ - Directorios personales de los usuarios.
├── root/ - Directorio personal del administrador del sistema (root).
├── tmp/ - Archivos temporales que se eliminan tras reiniciar.
├── dev/ - Archivos especiales que representan dispositivos de hardware.
├── sbin/ - Ejecutables esenciales para la administración y recuperación del sistema.
├── proc/ - Sistema de archivos virtual con información sobre procesos y estado del sistema.
├── sys/ - Sistema de archivos virtual que representa el hardware del sistema.
├── opt/ - Software adicional o de terceros.
├── mnt/ - Puntos de montaje temporal para sistemas de archivos.
├── media/ - Puntos de montaje para dispositivos extraíbles como USB o CD/DVD.
└── lib/ - Bibliotecas compartidas esenciales para /bin/ y /sbin/.
```

Ficheros importantes en Linux

El sistema de archivos no solo organiza directorios, si no que también contiene ficheros específicos con roles clase para el correcto funcionamiento del sistema y la configuración.

/
├── etc/
│ ├── passwd - Información sobre los usuarios del sistema.
│ ├── group - Información sobre los grupos del sistema.
│ ├── shadow - Contraseñas de los usuarios almacenadas en forma cifrada.
│ ├── fstab - Configuración de montaje de sistemas de archivos. (Suele editarse manualmente)
│ ├── network/
│ │ └── interfaces - Configuración de las interfaces de red (distribuciones más antiguas).
│ ├── hostname - Nombre del host del sistema. (Suele editarse manualmente)
│ ├── resolv.conf - Configuración de servidores DNS. (Suele editarse manualmente)
│ └── crontab - Configuración de tareas programadas del sistema.
├── var/
│ ├── log/
│ │ ├── syslog - Registro principal del sistema.
│ │ ├── auth.log - Registros de autenticación y acceso.
│ │ └── dmesg - Registros del kernel durante el arranque.
│ └── spool/
│ └── cron/ - Tareas programadas pendientes de ejecución.
├── boot/
│ └── grub/
│ └── grub.cfg - Configuración del cargador de arranque GRUB.
├── proc/
│ ├── cpuinfo - Información sobre la CPU del sistema.
│ ├── cmdline - Contiene toda la información que le fue pasada al kernel cuando inicia.
│ ├── meminfo - Información sobre la memoria del sistema.
│ ├── modules - Contiene una lista de los módulos es están cargados actualmente.
│ └── uptime - Tiempo de actividad del sistema.
├── home/
│ └── nom_user/
│ ├── .bashrc - Configuración del shell bash para un usuario.
│ └── .ssh/
│ ├── id_rsa - Clave privada SSH del usuario.
│ └── authorized_keys - Claves públicas autorizadas para conexión SSH.
├── root/
│ └── .bashrc - Configuración del shell bash para el usuario root.
├── usr/
│ ├── bin/
│ │ └── top - Ejecutable del comando `top` para monitoreo.

│ └── share/
│ └── man/ - Páginas de manual del sistema.
└── dev/
├── null - Dispositivo especial para descartar datos.
├── sda - Disco duro principal (bloque).
└── tty - Dispositivos de terminal.

Variables de entorno en Linux

Son valores dinámicos que influyen en cómo se ejecutan los procesos dentro del sistema. Estas variables permiten a las aplicaciones obtener información sobre el entorno o establecer configuraciones predeterminadas para su ejecución. Estas pueden ser consultas, definidas y manipuladas mediante comandos específicos:

- Visualizar variables de entorno:
  - env: Muestra todas las variables de entorno disponibles.
  - echo $NOMBRE_VARIABLE: Muestra el valor de una variable específica.
- Definir una nueva variable local
  - Variable=ruta del directorio:$Variable: Añade una nueva ruta a la variable seleccionada.
- Definir una nueva variable de entorno:
  - export NOMBRE_VARIABLE=valor: Crea o modifica una variable de entorno en la sesión actual.
- Eliminar una variable de entorno:
  - unset NOMBRE_VARIABLE: Elimina la variable de entorno especificada.
  Variables de entorno importantes:
- PATH: Define las rutas donde el sistema buscará ejecutables cuando se ejecuten comandos.
- HOME: Especifica el directorio personal del usuario actual.
- USER: Contiene el nombre del usuario que inició sesión.
- SHELL: Indica la Shell que se esta utilizando en la sesión.
- TERM: Define el tipo de terminal en uso, útil para ajustar configuraciones de aplicaciones basadas en terminal.
- EDITOR: Estable el editor de texto predeterminado para programas como crontab o git.
- LANG: Especifica la configuración de idioma y localización del sistema.

Rutas absolutas y relativas

Las rutas se utilizan para ubicar archivos y directorios dentro del sistema de archivos. Estas se dividen en rutas absolutas y rutas relativas:

1. Rutas Absolutas:
   - Representan la ubicación de un archivo o directorio en relación con la raíz del sistema (/).
   - Siempre comienzan con el caráctes /.
   - Son independientes del directorio en el que se encuentre el usuario.
2. Rutas Relativas:
   - Indican la ubicación de un archivo o directorio en relación con el directorio actual de trabajo.
   - Nunca comienzan con el carácter /.
   - Son dependientes del directorio donde esté trabajando el usuario.
3. Caracteres Especiales en la navegación:
   - . (Punto):
     - Representa el directorio actual.
     - Se utiliza para referirse al directorio donde se encuentra el usuario.
     - Cuando un archivo o directorio comienza con (.), este se considera oculto
   - .. (Punto Punto):
     - Representa el directorio padre, en otras palabras, el directorio inmediatamente superior al actual.

Comprensión de estándares

La entrada y salida de datos entre programas, comandos y usuarios está gestionada mediante tres flujos principales, conocidos como Stdin, Stdout y Stderr. Estos estándares son esenciales para la comunicación eficiente entre procesos y la interacción con el usuario.

- Stdin o Standar input: Es el flujo estándar utilizado para recibir datos que un programa necesita procesar. Por defecto, está asociado al teclado, aunque puede redirigirse desde otros orígenes, como un archivo o la salida de otro comando. Esta representado por el número 0 en el sistema.
- Stdout o Standar output: Es el flujo estándar donde un programa envía los resultados de su ejecución. Por defecto, está asociado al emulador de terminal, pero se puede redirigir a archivos o a la entrada estándar de otro programa. Esta representado por el número 1 en el sistema.
- Stderr o Standar error: Es el flujo estándar utilizado para mensajes de error o diagnóstico generados por un programa. Por defecto, también está asociado al emulador de terminal, aunque puede redirigirse a un archivo separado o combinarse con la salida estándar. Esta representado por el número 2 en el sistema.

¿Qué es un descriptor de archivo?

Es un identificar numérico que utiliza el sistema para representar recursos de entrada y salida, como archivos, dispositivos o flujo de datos. En los sistemas UNIX y Linux, cada proceso que se ejecuta tiene tres descriptores de archivos abiertos por defecto:

- **0 (Stdin)**: Entrada estándar.
- **1 (Stdout)**: Salida estándar.
- **2 (Stderr)**: Salida de error estándar.
  Dentro de sus funciones principales están:
- Permiten a los programas interactuar con recursos externos de manera eficiente y unificada.
- Los identificadores se pueden manipular para redirigir la entrada y salida a diferentes destinos, como archivos, dispositivos o incluso a otros procesos.
  Los usos en el control de comandos:
- **Depurar programas**: Separa la salida normal de los errores para facilitar la identificación de problemas.
- **Automatización**: Hace posible almacenar datos para análisis posteriores o enviar información a otros procesos.
- **Flexibilidad**: Mediante comandos como grep, awk o sed. Los descriptores de archivo potencian el procesamiento avanzado de datos.

Redirecciones en Linux

Son herramientas que permiten controlar el flujo de entrada y salida de comandos, enviando o capturando datos hacia archivos o entre comandos
Operadores de redirección:

1. *(>) Sobreescritura de salida estándar*: Redirige la salida estándar de un comando hacia un archivo, sobrescribiendo el contenido existente.
2. *(>>) Añadir a un archivo*: Añade la salida estándar al final del archivo
3. *(|) Pipe*: Permite encadenar varios comandos, pasando la salida estándar de un comando como entrada estándar al siguiente.

Diferencias entre argumentos y parámetros

Los comandos pueden ser personalizados o configurados mediante parámetros y argumentos. Aunque a menudo se usan indistintamente, tienen roles específicos:

1. **Parámetros**:
   - Son opciones o indicadores que modifican el comportamiento de un comando.
   - Se especifican generalmente con:
     - Guion simple (-): Seguido de una letra.
     - Doble guion (--): Seguido de una palabra completa.

- Permiten activar funciones adicionales o cambiar cómo se ejecuta el comando.

2. **Argumentos**:
   - Son valores o datos que un comando necesita para realizar una acción específica.
   - Indican los elementos en los que debe actuar el comando.
   - A menudo aparecen después de los parámetros.
3. **Diferencias clave**:
   - Los parámetros alteran el comportamiento del comando.
   - Los argumentos especifican sobre qué datos debe actuar el comando.

Operadores lógicos

- **&&** (AND): Permite ejecutar un segundo comandos solo si el primero se ejecuta con éxito.
- **||** (OR): Permite ejecutar un segundo comandos solo si el primero falla.
- **!** (NOT): Invierte el resultado de la ejecución de un comando.
- **;** (Secuencia): Permite ejecutar varios comandos de forma secuencial, sin importar si los anteriores tienen éxito o fallan.

Caracteres glob

Son una característica de la shell, no es algo propio de algún comando específico. Como resultado a esto, se pueden usar con cualquier comando existente en Linux.

- **Asterisco (*)**: Coincide con cero o más caracteres en un nombre de archivo o directorio.
- **Interrogación (?)**: Coincide con exactamente un carácter en un nombre de archivo.
- **Corchetes ([ ])**: Coincide con cualquier carácter dentro del conjunto especifico.
- **Comillas simples (')**: Evitan que la shell interprete todos los caracteres especiales. "Protege" una cadena de ser cambiada por la shell.
- **Comillas dobles (")**: Permiten la expansión de variables y la sustitución de comandos, pero protegen otros caracteres especiales como espacios o glob.
- **Comilla invertida (`` ` ``)**: Permite la sustitución de comandos, ejecutando el comando dentro de la shell y reemplazándolo con su salida.
- **Barra inclinada (\ )**: Escapa un carácter especial para tratarlo literalmente en lugar de interpretarlo.

Expresiones Regulares básicas

Es una colección de caracteres "normales" y "especiales" que se utilizan para que coincida con un patrón simple o complejo. Las expresiones regulares básicas son las siguientes:

Expresión Regular	Coincidencias
Punto (.)	Cualquier carácter individual.
Corchetes ([ ])	Una lista o rango de caracteres que coinciden con un carácter.
Asterisco (*)	El carácter previo que se repite cero o más veces
Acento circunflejo (^)	El texto siguiente debe aparecer al principio de la línea.
Dolar ($)	El texto anterior debe aparecer al final de la línea

Comandos para Linux

La sintaxis para los comandos en Linux es: [Comando (Opciones) (Argumentos)]
Opciones: Las opciones se pueden utilizar para modificar el comportamiento de un comando.
Argumentos: Un argumento se puede usar para especificar algo sobre lo que el comando debe actuar.

De información y ayuda

- man: Es la primera línea de ayuda para entender cómo funcionan otros comandos.
  -k: Muestra un resumen de todas las páginas del manual sobre la palabra clave asignadas.
  -f: Muestra las páginas que coinciden con el nombre especificado.
- info: Es similar al comando man, pero con un formato diferente ya que a veces suele dar información más detallada que man.
- whatis: Brinda una descripción bastante breve del comando a consultar.
- apropos: Ayuda a encontrar comandos relacionados con una palabra clave.
- whereis: Busca los comandos, archivos de código fuente y las páginas man en las ubicaciones especificas donde estos archivos se almacenan.
- which: Muestra la ruta del archivo ejecutable asociado a un comando.
- type: Se utiliza para determinar información acerca de varios comandos, también puede identificar comandos integrados en bash u otra shell. También puede identificar los aliases para otros comandos.
  -a: Revela la ruta de otro comando.
  -t: Indica el tipo de comando (ejecutable, función, alias, etc)
- help: Proporciona información solamente de comandos incorporados en la Shell que se esté usando.
- alias: Permite ver los alias que están definidos en la shell.

De búsqueda de ficheros o directorios

- **find**: Busca archivo en el sistema de archivos, ya sea por nombre o incluso caracteres comodines si no se sabe el nombre exacto. Además, puede buscar archivos en función de los metadatos de archivos, como: tipo de archivo, tamaño de archivo y propiedad de archivo. Dentro de sus parámetros están:
  -name: Busca un archivo por el nombre.
  -ls: Muestra los detalles de los archivos.
  -size (+/- Tamaño que se quiere buscar): Permite buscar los archivos por tamaño.
  -iname: Busca por nombre de archivo, pero no es sensible a las mayúsculas y minúsculas.
  -user: Devuelve los archivos que son propiedad de un usuario especifico.
- **grep**: Filtra las líneas en un archivo o en la salida de otro comando basado en la coincidencia de un patrón, puede ser por medio de texto exacto o por uso de expresiones regulares. Existen varios parámetros:
  -i: hace que a la búsqueda no le importen las mayúsculas o minúsculas.
  -c: permite contar el número de líneas que contiene el patrón de búsqueda.
  -n: Muestra los números de la línea originales.
- **locate**: Principalmente funciona para encontrar la ubicación de ficheros dentro del sistema. Para actualizar la base de datos de "locate" se utiliza sudo updatedb.

Para navegar en la terminal

- **cd**: Sus siglas significan Change Directory y funciona para cambiar el directorio actual, recibe tanto rutas relativas como rutas absolutas.
  Doble punto (..): Representa al directorio arriba del directorio actual.
  Punto (.): Se utiliza para referirse al directorio actual.
- **pwd**: Sus siglas significan Print Working Directory, básicamente lo que hace es imprimir el directorio actual de trabajo.
- **ls**: Sus siglas significan List y su función es listar los directorios y ficheros, posee parámetros importantes como:
  -a: Muestra todos los ficheros, incluyendo los ocultos.
  -l: Muestra todo en un formato más largo.
  -R: Entra en cada directorio enlista y enlista su interior.
  -h: Muestra los detalles, pero con tamaño legible para el humano. como KB,MB,GB.
  -d: Hace referencia al directorio actual y no al contenido dentro de él.
  -S: Ordena los archivos de en función del tamaño.
  -t: Muestra los archivos según el momento en el que se modificaron.
- **history**: Almacena un historial de comandos ejecutados. Se puede utilizar un comando desde el historial con un signo de exclamación (!) y el número de donde este el comando deseado.

De gestión de ficheros

- touch: Su principal función es crear ficheros vacíos.
- cp: Es el acrónimo de copy y permite copiar tanto ficheros como directorios, su sintaxis es: cp (origen) (destino). Sus parámetros son:
  -i: Solicita confirmación antes de copiar un fichero.
  -r: Funciona para copiar directorios de forma recursiva.
  -v: Reproduce la salida del comando.
- mv: Es el acrónimo de move y funciona para mover y renombrar ficheros y directorios su sintaxis es: mv (origen) (destino). Sus parámetros son:
  -i: Solicita confirmación antes de mover un fichero.
  -u: Solamente sobrescribe el archivo si el archivo de destino es más antiguo.
- rm: Es el acrónimo de remove y funciona para eliminar ficheros y directorios, su sintaxis es: rm (activo que se desea eliminar). Sus parámetros son:
  -i: Solicita confirmación antes de eliminar un fichero.
  -r: Funciona para eliminar directorios de forma recursiva.

De visualización ficheros

- cat: Es el acrónimo de concatenate, se originó para concatenar y mostrar el contenido de ficheros, pero actualmente muy pocas veces se le da ese uso, ya que principalmente se utiliza para leer ficheros, no muy grandes, dentro de la terminal, su sintaxis es: cat (ruta al fichero). Su parámetro es:
  -n: Funciona para enumerar las líneas mientras se lee un fichero.
- head: Muestra las primeras 10 líneas de un fichero, la diferencia con "cat" es que mostraría todo el fichero. Con el parámetro
  -n: Se le puede indicar el número de líneas que se desean leer.
- tail: Muestra el fichero, pero partiendo de las ultimas 10 líneas. Con el parámetro
  -n: Se le puede indicar el número de líneas que se desean leer.
  -f: Permite seguir el contenido en tiempo real de un fichero.
- less: Muestra el contenido de un fichero de forma paginada, en otras palabras, permite visualizar el fichero página por página. Sus parámetros son:
  -n: Muestra el número de líneas al visualizar un fichero.
  -f: Sigue el contenido del fichero en tiempo real.
- sort: Es un comando que se utiliza para ordenar líneas de texto en fichero, ya sea alfabética o numéricamente y también soporta la ordenación basada en columnas específicas. Es especialmente útil para procesar y organizar grandes cantidades de datos. Parámetros que se pueden llegar a usar
  -f: ignora el case sensitive al ordenar.
  -o: funciona para guardar el resultado en un fichero.
  -n: para ordenar de forma numérica.
  -t: para ordenarlos separados por un campo en específico.

-c: verifica si el fichero ya está ordenado.
-k: Se le indica un argumento para indicar el número de campos.
-r: Realiza un sort inverso.

De gestión de directorios

- **mkdir**: Es el acrónimo de make directory y básicamente su función es crear directorios dentro del sistema de archivos, su sintaxis es mkdir (ruta del directorio). Su parámetro es:
  -p: Funciona para crear directorios padres.
- **rmdir**: Su significado es remove directory y únicamente funciona para eliminar directorios que están vacíos. Su parámetro es:
  -r: Elimina directorios no vacíos.
- **ln**: Permite crear enlaces físico a un fichero o directorio. Entre sus parámetros esta:
  -s: Permite crear enlaces simbólicos.

De gestión de usuarios

- **adduser**: Es un comando utilizado para crear nuevos usuarios en el sistema.
- **deluser**: Es un comando utilizado para eliminar un usuario del sistema. Su parámetro más común es:
  –remove-home: Permite eliminar el usuario y a su vez el directorio home asignado a dicho usuario.
- **usermod**: permite modificar las propiedades de un usuario como el nombre del usuario, su directorio home o el grupo al que este asignado. Sus principales parámetros son:
  -l : Permite cambiar el nombre login del usuario.
  -d: Permite cambiar el directorio home del usuario.
  -L: Bloquea la cuenta del usuario.
  -U: Desbloquea la cuenta del usuario.
- **passwd**: utilizado para cambiar la contraseña de un usuario. Permite ciertos parámetros
  -l bloquea la contraseña del usuario.
  -u: desbloquea la contraseña del usuario.
- **id**: Es el acrónimo de identification y permite identificar la información del usuario actual o del usuario que se le indique. Sus parámetros son
  -u: solamente muestra el UID del usuario (id del usuario).
  -g: que muestra el GID del usuario (grupo del usuario).
- **who**: Permite saber quién está conectado al sistema. Permite varios parámetros como:
  -u: Muestra los usuarios conectados junto a la hora de conexión.
- **w**: Proporciona una lista más detallada sobre los usuarios que actualmente están en el sistema.
- **su:** Es el acrónimo de switch user y permite cambiar de usuario, si no se especifica un nombre de usuario siempre va a intentar cambiar al usuario root. Posee una serie de

parámetros como:
-l o *--login*: Proporciona un entorno similar al que tendría el usuario si se conectara.
-c: Permite ejecutar un comando como el usuario que se le indique sin necesidad de tener que iniciar sesión como ese usuario.

- **sudo**: Permite ejecutar comandos con privilegios de otro usuario. Posee varios parámetros como:
  -u: Permite especificar el usuario con el que se quiere ejecutar el comando
  -i: Proporciona un entorno similar al que tendría el usuario si se conectará
  -s: Se le indica al comando que proporcione una Shell del usuario que se le indique.
- **whoami**: Proporciona el nombre del usuario asociado con el user ID actual en la terminal.
- **finger**: Proporciona información sobre un usuario en particular.

De gestión de grupos

- **groupadd**: Permite crear grupos dentro del sistema, es necesario que se ejecute por el usuario root. Dentro de sus parámetros están:
  -g: Se utiliza para especificar un id al nuevo grupo.
- **groupdel**: Permite eliminar grupos, si le grupo posee a un usuario dentro primero hay que sacarlo.
- **groupmod**: Permite modificar el nombre de un grupo, posee varios parámetros
  -n: Permite cambiar el nombre de grupo.
  -g: Permite cambiar el GID del grupo.
  -aG: Sirve para añadir un usuario a un grupo
  -G: establece el grupo suplementario del usuario
- **gpasswd**: Se utiliza para administrar las contraseñas de grupos y miembros de cada grupo. Posee varios parámetros como:
  -a: con el que se puede añadir un usuario a un grupo.
  -d: con el que se puede eliminar un usuario de un grupo.
- **groups**: Permite ver los grupos a los que pertenece un usuario.
- **newgrp**: Permite cambiar el grupo primario al que pertenece el usuario.

De gestión de redes

En Linux de utilizan 2 tipos de nomenclatura, los nombres tradiciones (No son persistentes y poco descriptivos) y nombres predictivos (Son persistentes y descriptivos, se basan en la ubicación física o topología de hardware de la nic).

- **ifconfig**: Enlista todas las interfaces de red activas mostrando detalles como la dirección IP, mascara, estadísticas de paquete, entre otras. Entre sus parámetros están:
  up: Para encender una interfaz de red.
  down: Para apagar una interfaz de red.

- **ip**: Es una herramienta que muestra y manipula rutas, dispositivos, políticas de enrutamiento y túneles. Entre sus parámetros están:
  a o *addr*: Muestras todas las interfaces de red con sus direcciones ip.
- **ip route**: Muestra la tabla de rutas del kernel, básicamente la tabla de enrutamiento por las que viaja los paquetes. Entre sus parámetros están:
  add + *vía* : sirve para añadir una nueva ruta.
- **netstat**: Permite visualizar tablas de enrutamiento, estadísticas de interface, conexiones masquerales, miembros multicast. Posee varios parámetros:
  -t: Muestra las conexiones por TCP.
  -u: Muestra las conexiones por UDP.
  -l: Muestras solo los puertos en escucha.
  -p: Muestra el pid.
  -e: Muestra la información extendida.
  -i: Muestra estadísticas del tráfico de red.
  -r: Muestra la información de enrutamiento.
  -n: Muestra el número, no el nombre el servicio.
- **ping**: Permite verificar la conectividad que se posee con otros hosts. Posee varios parámetros:
  -c: Especifica el número de paquetes que se desea enviar.
- **traceroute**: Muestra la ruta que toman los paquetes para llegar a un host de destino. Sus parámetros son:
  -m: especifica el número de saltos antes de que el comando detenga el rastreo.
- **nslookup**: Permite realizar consultas al servidor DNS. Existen varios parámetros:
  -type: Sirve para especificar tipo de registro que se desea buscar.
- **hostname**: Funciona para mostrar y configurar el nombre del sistema operativo.
- **nc**: Significa netcat, es una herramienta principalmente usada para abrir y cerrar puertos. Posee varios parámetros como:
  -l: Permite ponerse a la escucha en un puerto.
  -v: Activa el modo verboso para que muestre más detalles.
  -p: Con él se le puede indicar un puerto.
  -n: Para que no realice la resolución DNS.
  -u: para utilizar el protocolo UDP en vez de TCP que viene por defecto.
- **dig**: Prueba la funcionalidad del servidor DNS del host que se esta utilizando, para esto realiza consultas en el servidor DNS para determinar si la información necesaria está disponible.

De gestión de paquetes DEB

- **apt**: Viene de forma nativa en distribuciones basadas en Debian, por lo que en otras distros no se va a encontrar de forma nativa. Son las siglas de advanced packaging tool. Simplifica

la instalación, actualización y eliminación de paquetes, ya que resuelve automáticamente las dependencias necesarias para instalar un paquete. Para su correcto funcionamiento es necesario correr el comando con permisos de sudo. Posee varios parámetros:
update: Actualiza la lista de paquetes disponibles y sus versiones.
upgrade: Instalas las versiones más recientes de los paquetes instalados en el sistema.
full-upgrade: realiza una actualización completa.
install: Realiza la instalación de paquetes.
remove: Elimina los paquetes instalados en el sistema.
autoremove: Elimina los paquetes que fueron instalados como dependencias y que ya no son necesarios.
--purge remove: Elimina todos los archivos de un paquete de software, incluyendo archivos de configuración.

- **apt-get**: Es una versión antigua de **apt** pero es más estable y comparte sus mismos parámetros, existen otros como:
  dist-upgrade: Demás de hacer las funciones del parámetro upgrade también es capaz de manejar de forma inteligente los cambios en las dependencias con nuevas versiones a nivel de paquetes.
- **apt-cache**: Realiza búsqueda y obtiene información sobre los paquetes almacenados en la cache de apt. Para su correcto funcionamiento es necesario correr el comando con permisos de sudo. Alguno de sus parámetros son:
  depends: Muestras las dependencias del paquete mencionado.
  policy: Muestras la política instalada del paquete mencionado, incluyendo las versiones y de cual repositorio se instalará el paquete.
  search: Busca en los repositorios del sistema en paquete mencionado.
- **dpkg:** Es el acrónimo de Debian package, básicamente es el sistema de gestión a bajo nivel en distros basadas en Debian, que solamente gestiona paquetes en específicos los **.deb**. Sus parámetros son:
  -i: Para instalar.
  -l: Muestra una lista de los paquetes instalados en el sistema.
  -r: Sirve para eliminar un paquete; pero no es una eliminación completa ya que elimina el paquete, pero mantiene sus archivos de configuración en el sistema.
  -p: Elimina el paquete en su totalidad.
  -L: Enlista los archivos de componen un paquete en especial.
- **dpkg-query**: Se utiliza para realizar consultas a la base de dpkg.
- **dpkg-reconfigure**: Se utiliza para reconfigurar un paquete después de que este haga sido instalado.
- **dpkg-dpkg-trigger**: Permite activar eventos en el sistema de paquetes.

De gestión de paquetes RPM

- **rpm**: Es el sistema de administración de paquetes de bajo nivel en los paquetes Red Hat. Realiza sus consultas mediante la conexión a una base de datos local en la máquina. Dentro de sus parámetros están:
  -qa: Muestra una lista de todos los paquetes instalados en el sistema actualmente.
  -ql: Enlista los archivos que componen un paquete en especial.
  -qi: Consulta un paquete y obtiene información o el estado del mismo.
- **yum**: Automatiza el proceso de resolución de problemas con las dependencias, si lo que se quiere es agregar, actualizar o eliminar paquetes se necesitan permisos de root. Dentro de sus parámetros están:
  search: Busca paquetes desde los repositorios configurados.
  install: Instala paquetes, junto a sus dependencias.
  update: Actualiza todos los paquetes.
  remove: Elimina un paquete, resolviendo los problemas con las dependencias.

De gestión de paquetes web

- **wget**: Permite descargar fichero de internet soportando los principales protocolos HTTP, HTTPS o FTP. Sus parámetros son:
-c: Permite continuar una descarga que fue interrumpida.
--no-check-certificate: Ignora la verificación del certificado TLS.

- **curl**: Funciona para transferir datos desde o hacia un servidor usando diversos protocolos y principalmente se utiliza para la prueba de apis o meramente descargar ficheros o enviar datos. Entre sus parámetros están:
  -X: Se puede especificar el método de solicitud a HTTP
  -H: Se le pueden añadir encabezados a la solicitud.

De gestión de ficheros comprimidos

- **tar**: Permite crear archivos comprimidos. Posee varios parámetros
  -c: indica la creación de un fichero,
  -z: utiliza gzip para comprimir,
  -v: muestra los ficheros procesados,
  -f: especifica el nombre del fichero de salida,
  -x: indica la extracción de ficheros.
  -t: Lista los documentos en el archivo empaquetado.
  -j: Descomprime con bzip2 antes de la lectura.
- **gzip**: Es un comando que permite comprimir ficheros individuales a los que se les añade la extensión .gz. Posee los siguientes parámetros:
  -d: se pueden descomprimir archivos.
  -l: Muestra información sobre la compresión.

- zip: Permite comprimir ficheros, pero con la extensión .zip.
- unzip: Permite descomprimir archivos .zip.

De gestión de procesos

Cada proceso en Linux consta de varios puntos clave. Cada proceso cuanta con PID (Process Idetifier) es único para cada proceso, a su vez también existen tipos de procesos, como: procesos de usuarios que son los procesos iniciados y controlados por los usuarios y por otro lado los procesos del sistema que son los procesos iniciados y manejados por el sistema operativo y son necesarios para el funcionamiento de este. Los procesos tienen un estado asignado, ya sea running, sleeping, stoped o zombie.

- ps: Muestra información de los procesos activos. Posee parámetros como:
  a: Muestra los procesos de todos los usuarios.
  u: Muestra información detallada del usuario.
  x: Incluye los procesos que no tengan una terminal asociada.
  aux o *-ef*: Muestra todos los procesos del sistema.
- pstree: Muestra los procesos en formato de árbol familiar, mostrando a los padres e hijos.
- top: Muestra en tiempo real todos los procesos en ejecución y su consumo de recursos.
- free: Muestra la cantidad de memoria utilizada en el momento.
- kill: Se utiliza para enviar señales a un proceso y comúnmente se realiza para eliminar el proceso. Para este comando se utilizan las siguientes señales.
- killall: termina todos los procesos asociados en el nombre que se le brinde.
- - free: Muestra el estado de la memoria del sistema incluyendo la memoria física y swap.
    Su parámetro es:
    -b: Muestra la información en bits.
- fg: Es el acrónimo de ForeGround, permite traer un proceso en segundo plano a primer plano.
- bg: Es el acrónimo de BackGround, despierta un proceso que este suspendido, pero lo ejecuta en segundo plano.
- Jobs: Permite ver los procesos suspendidos o en segundo plano.
- disown: elimina toda la lista de trabajos y por consecuencia se puede separar de un proceso que ya se haga creado desde la terminal y dejarlo de forma independiente en el sistema.

Para la gestión de propietarios y permisos

- chown: Permite cambiar el propietario y el grupo de un fichero o directorio, se debe de utilizar por medio del usuario root. Un parámetro que se puede utilizar es:
  -R: Aplica los cambios a una estructura completa de directorios.

- chgrp: Es el acrónimo de Change Group, este únicamente permite cambiar el grupo de que va a pertenecer un fichero o directorios, se debe de utilizar por medio del usuario root.. Un parámetro que se puede utilizar es:
  -R: Aplica los cambios a una estructura completa de directorios.
- stat: Muestra información detallada sobre los permisos y la propiedad de los directorios
- chmod: Este comando permite modificar los permisos tanto de los directorios como de los ficheros y demás activos del sistema y permite realizar esto tanto de forma octal como de forma simbólica. Un parámetro que se puede utilizar es:
  -R: Que aplica los permisos de forma recursiva a nivel de directorios.

Comandos adicionales

- echo: Permite imprimir en pantalla diferentes tipos de valores. Sus parámetros son:
  -n: No imprime la línea nueva al final del texto
  -e: Habilita la interpretación de secuencias de escape como \n que es un salto de línea y \t que indica una tabulación.
- du: Permite mostrar el espacio utilizado por ficheros y directorios en disco. Se puede utilizar le parámetro:
  -a: Que muestra el uso de todos los ficheros.

Editores de ficheros desde la terminal

Vim/NVim

Es una versión mejorada del editor “vi” que es uno de los editores más utilizados en la comunidad de Linux. Para abrir un fichero con vim se utilizaría el comando vim seguido de la ruta del fichero y si el fichero no existe se creará uno con la ruta que se le haga asignado. Ofrece múltiples modos como:

Modo normal

Modo por defecto cuando se abre un fichero, se pueden utilizar teclas y comandos para navegar por el fichero y realizar acciones como borrar el texto, copiar y pegar, etc. Pero hay que tener en cuenta que en este modo no se puede insertar texto directamente. Teclas útiles para el modo normal:

1. Navegación básica:
   - h: moverse a la izquierda.
   - j: moverse hacia abajo.
   - k: moverse hacia arriba.
   - l: moverse a la derecha.
2. Navegación avanzada:

- w: salta al inicio de la siguiente palabra.
- e: salta al final de la palabra actual.
- b: Retrocede al inicio de la palabra anterior.

3. Manipulación de texto:
   - dd: elimina una línea.
   - yy: se puede copia una línea.
   - p: copiar lo que se haya pegado o eliminado
   - u: deshace la última acción.
   - Ctrl + r: Rehace la última acción deshecha.

Modo insertar

Permite añadir texto al archivo. Para entrar en este modo desde el modo normal se debe de:

- Presionar i para insertar texto antes del cursor.
- Presionar a para insertar texto después del cursor.
- Presionar o para insertar una nueva línea debajo del cursor.
  Para salir de este modo se debe de presionar Esc.

Modo comando

Para acceder a este modo se debe presionar : (dos puntos). Aquí se pueden ejecutar comandos avanzados para guardar, salir, buscar y realizar otras operaciones.

1. Guardar y salir:
   - :w: Guarda los cambios.
   - :q: Salir (solo si no hay cambios pendientes).
   - :q!: Salir sin guardar.
   - :wq o :x: Guarda y sale.
2. Búsqueda:
   - :/palabra: Busca "palabra" hacia adelante.
   - :?palabra: Busca "palabra" hacia atrás.
   - n: Salta al siguiente resultado.
   - N: Salta al resultado anterior.

Nano

Nano es un editor de texto como vim pero mucho más simple de utilizar. Se utiliza principalmente desde la terminal y es ideal para editar rápidamente archivos sin una curva de aprendizaje compleja.

Opciones comunes al abrir nano

Al usar nano se pueden usar ciertos parámetros para modificar su comportamiento. Dentro de los cuales están:
La sintaxis es: [nano (parámetro) nomb_archivo]

- -m: El uso del ratón para interactuar con el texto.
- -l: Activa la visualización del número de líneas del archivo.
- -v: Abre el archivo en modo lectura, evitando las modificaciones accidentales.
- -B: Crea una copia de seguridad del archivo antes de guardar los cambios.
- -E: Elimina espacios en blanco innecesarios al final de las líneas.
- -i: Agrega sangría automáticamente al comienzo de nuevas líneas.

Atajos en Linux

De línea de comandos

- Ctrl+a: Permite moverse al inicio de la línea.
- Ctrl+e: Permite moverse al final de la línea.
- Ctrl+b: Mueve el cursor un carácter hacia atrás.
- Ctrl+f: Mueve el cursor un carácter hacia adelante.
- Alt+b: Mueve el cursor una palabra hacia atrás.
- Alt+f: Mueve el cursor una palabra hacia adelante.
- Ctrl+xx: Alterna entre el inicio de la línea y la posición actual del cursor.

De procesos

- Ctrl+c: Termina el proceso actual.
- Ctrl+z: Suspende el proceso actual.
- Ctrl+d: Permite salir de la terminal si no hay texto

De historiales de comandos

- Ctrl+p: Con él se puede navegar hacia el comando anterior en el historial.
- Ctrl+n: Sirve para navegar al siguiente comando en el historial.
- Ctrl+r: Con él se puede buscar dentro del historial
- Ctrl+o: Dentro del historial de comandos se puede ejecutar el comando en cuestión. Es un complemento de Ctrl+r.

Del emulador de terminal

- Ctrl+l: Se limpia la terminal.

- Ctrl+s: Se detiene el desplazamiento de salida de la terminal.
- Ctrl+q: Reanuda el desplazamiento.
- Ctrl+ Alt + (F1-F6): Permite cambiar entre las terminales TTI

Alias en Linux

Es una forma de darle una palabra clave a una instrucción personalizada, en otras palabras, funcionan para poder ejecutar un comando con ciertos parámetros y argumentos, pero hacer solamente a través de una palabra clave. Con el comando alias se muestran los comandos que se hayan utilizado, para crear un alias se utiliza alias (nombre del alias = comando). Hay que tener en cuenta que los alias son temporales, si se reinicia o se apaga el sistema los alias creados se eliminan, para hacerlos permanentes hay que añadirlos al fichero .bashrc. Para eliminar un alias se utiliza unalias + nombre del alias.

Tipos de archivos

Los tipos de archivos se pueden identificar en el primer campo a la hora de enlistar los archivos con el comando ls con el parámetro -l.
Los tipos de archivos son:

Símbolo	Tipo de archivo	Descripción
d	Directorio	Un archivo usado para contener otros archivos.
-	Archivo ordinario	Incluye archivos leíbles, imágenes, archivos binarios, y archivos comprimidos.
l	Enlaces simbólicos	Apunta a otro archivo.
s	Socket	Permite la comunicación entre procesos.
p	Tubería (pipe)	Permite la comunicación entre procesos.
b	Archivo bloque	Usado para comunicaciones con el equipo (*hardware*).
c	Archivo carácter	Usado para comunicaciones con el equipo (*hardware*).

Permisos

Existen varios tipos de permisos que se le pueden asignar tanto a los ficheros como a los directorios, los cuales son los permisos de lectura o read, escritura o write y ejecución o execute. A nivel de asignación de permisos se puede realizar de 2 maneras, ya sea asignación octal o simbólica.

Asignación octal de permisos

Cada tipo de permiso tiene un valor numérico:

- r = 4, w = 2, x = 1
  Se asignan combinando los valores para cada categoría de usuarios:
- Propietario (u): Usuario que creó el archivo/directorio.
- Grupo (g): Usuarios que pertenecen al grupo del archivo/directorio.
- Otros (o): Todos los demás usuarios.
  Ejemplo:
  chmod 754 nomb_archivo
- **7 (propietario)**: r + w + x = 4 + 2 + 1 = 7
- **5 (grupo)**: r + x = 4 + 1 = 5
- **4 (otros)**: r = 4

Asignación simbólica de permisos

Se utiliza letras para definir permisos y operadores para asignarlos:

1. Permisos:
   - r = lectura, w = escritura, x = ejecución.
2. Categorías:
   - u = propietario, g = grupo, o = otros, a = todos.
3. Operadores:
   - +(agrega), - (remueve), = (establecer exacto)
   Ejemplos:
   chmod g+w nomb_archivo -> Agrega permisos de escritura al grupo.
   chmod g-w nomb_archivo -> Remueve permisos de escritura al grupo.

Otros permisos

SUID O Set User ID

- **Función**:
  - Permite ejecutar un archivo como si fuera el propietarios, sin importar quién lo ejecute.
  - Únicamente se puede asignar a archivos ejecutables.
- **Símbolo**: (s) reemplaza la (x) en los permisos del grupo.
- **Octal**: Se representa con un **4** *Ejemplo*: chmod **4**755 nomb_archivo

SGID o Set Group ID

- **Función**:
  - En archivos: Se ejecutan como si pertenecieran al grupo del archivo.

    - En directorios: Los archivos creados heredan el grupo del directorio.
- Símbolo: (s) reemplaza la (x) en los permisos del propietario.
- Octal: Se representa con un 2 *Ejemplo*: chmod 2755 nomb_archivo

Stickt Bit

- Función:
    - Restringe la manipulación de archivos en directorios compartidos, permitiendo solo al propietario o el administrador pueda eliminarlos o modificarlos.
    - Únicamente es aplicable a directorios.
- Símbolo: (t) en los permisos de otros.
- Octal: Se representa con un 1 *Ejemplo*: chmod 1755 nomb_archivo.

Resumen de permisos

Permiso	Octal	Símbolo
Lectura (r)	4	r
Escritura (w)	2	w
Ejecución (x)	1	x
SUID (propietario)	4	s
SGID (grupo)	2	s
Sticky Bit (otros)	1	t