1. **Características de GNU/Linux:**
   * + **Multitarea**: Se pueden realizar varias actividades a la vez.
     + **Multiusuario**: Varios usuarios pueden trabajar concurrentemente en una computadora con varias terminales de forma que tengan la sensación de que es el único que está trabajando en el sistema.
     + **Conectividad**: Permite las comunicaciones en red y el acceso a recursos remotamente. Por ejemplo, podemos acceder a nuestros datos situados en una máquina a través de otro equipo, conectados ambos a Internet.
     + **Multiplataforma**: Se puede instalar en multitud de dispositivos.
     + **Libre**: Su código fuente está disponible. Cualquiera puede usarlo, modificarlo y distribuir.
     + Posee diversos intérpretes de comandos, de los cuales algunos son programables.
     + Todo es un archivo (hasta los dispositivos y directorios)
     + Cada directorio puede estar en una partición diferente
   1. ..
   2. GNU es un sistema operativo de tipo Unix desarrollado por y para el Proyecto GNU.  
      \* **GNU no es Unix** (nombre elegido debido a que GNU sigue un diseño **tipo Unix** y se mantiene compatible con éste, pero se distingue de Unix por ser **software libre** y por **no contener código de Unix**).
   3. * Fue iniciado por Richard Stallman en 1983 con el fin de crear un Unix libre.
      * Para asegurar que el mismo fuera libre, se necesitó crear un marco regulatorio conocido como GPL.
      * En 1985, Stallman crea la FSF (Free Software Foundation), con el fin de financiar el proyecto GNU.
      * En 1990, GNU ya contaba con un editor de textos (**Emacs**), un compilador (**GCC**) y gran cantidad de bibliotecas que componen un Unix t ́ıpico. **Pero faltaba el Kernel..**
      * Se venía trabajando en un kernel conocido como **TRIX**, es en 1988 que se decide abandonarlo debido a su complejidad (corrí́ıa en hardware muy costoso). Entonces se decide adoptar como base el kernel **MACH** para crear **GNU Hurd**, el cual tampoco prosperó
      * Linus Torvalds ya venía trabajando desde 1991 en un Kernel denominado **Linux**, el cual se distribuiríıa bajo licencia GPL.  
        **Hasta que en el año 1992, Torvalds y Stallman deciden fusionar ambos proyectos, y es allí donde nace GNU/Linux.**
   4. La multitarea es la característica de los sistemas operativos modernos de permitir que varios procesos o aplicaciones se ejecuten aparentemente al mismo tiempo, compartiendo uno o más procesadores. GNU/Linux hace uso de ella.
   5. Es una norma escrita por la IEEE. Dicha norma define una interfaz estándar del sistema operativo y el entorno, incluyendo un intérprete de comandos (o "shell"), y programas de utilidades comunes para apoyar la portabilidad de las aplicaciones a nivel de código fuente.
2. **Distribuciones de GNU/Linux:**
   1. Una distribución es una customización de GNU/Linux formada por una versión de kernel y determinados programas con sus configuraciones.  
      Por ejemplo Fedora, Ubuntu, Debian y Slackware.
      * Diferencias:
        1. **Ubuntu** y **Fedora** ofrecen soporte comercial, en cambio **Debian** está mantenida por la comunidad, y **Slackware** no está relacionada con ninguna comunidad ni empresa.
        2. **Ubuntu** está orientado al usuario promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia del usuario.  
           **Fedora** no busca sólo incluir software libre y de código abierto, sino ser el líder en ese ámbito tecnológico.  
           **Slackware** se asemeja en alto grado a los sistemas operativos Unix. A tal efecto, incluye software que normalmente no se encuentra en otras distribuciones Linux, tal como la última versión del entorno de comandos Korn shell.  
           **Debian** es una comunidad conformada por desarrolladores y usuarios, que mantiene un sistema operativo GNU basado en software libre.
        3. **Ubuntu**, **Fedora** y **Slackware** optan por un kernel monolítico. **Debian** se encuentra precompilado, empaquetado y en formato deb para múltiples arquitecturas y para varios kernels.
   2. Las distribuciones de Linux tienen en común el kernel, pero el resto de componentes (las herramientas, la shell, el Display Server, la GUI) varían entre sí, se personalizan o se crean desde cero, por eso las distribuciones son tan diferentes entre sí. Aunque en la mayoría de los casos la principal diferencia es la GUI, o los programas y herramientas que vienen incluidos.

**Cada distribución Linux tiene un objetivo, que justifica su existencia.** Por ejemplo, distros como Ubuntu se centran en ser lo más amigables posible a la hora de instalarse o descargar programas. Linux Mint aprovecha el hardware potente para competir con Windows o MacOS. Si tienes un ordenador viejo, hay distros ligeras como Puppy Linux. ¿Quieres instalar Linux en un servidor? Usa Debian. ¿Deseas jugar a videojuegos en Linux? SteamOS es la mejor.

* 1. Debian es una comunidad conformada por desarrolladores y usuarios, que mantiene un sistema operativo GNU basado en software libre.  
     **Historia:**
     + Ian Murdock fundó oficialmente el proyecto Debian el 16 de agosto de 1993. Una distribución realizada de forma abierta, siguiendo el espíritu de Linux y GNU.
     + Por un año fue patrocinada por el proyecto GNU de la Free Software Foundation, la organización fundada por Richard Stallman y asociada con la Licencia Pública General (GPL).
     + Debian 0.91 fue publicada en enero de 1994. Tenía un primitivo sistema de empaquetamiento que permitía a los usuarios manipular paquetes pero que no hacía mucho más. Había pocas personas trabajando en ella.
     + Debian 0.93 Release 5 tuvo lugar en marzo de 1995 y fue la primera versión «moderna» de Debian: Esta tuvo muchos más desarrolladores y se usaba dpkg  para instalar y mantener todos estos paquetes después de que el sistema base esté instalado.
     + Desde 1995 Desde aquel tiempo, el proyecto Debian estuvo creciendo para incluir varias migraciones a otras arquitecturas, y otra migración al nuevo kernel (no linux), el microkernel GNU Hurd.
     + Debian 2.2 (potato) se publicó el 15 de agosto del 2000, es la primera versión en incluir las arquitecturas PowerPC y ARM. Mostró cómo un esfuerzo de software libre podría llevar a un moderno sistema operativo a pesar de todas las cuestiones alrededor de él.
     + Debian 3.0 (woody) fué publicada el 19 de julio de 2002 convirtiéndose en la primera versión en ser publicada en formato DVD además de los ya acostumbrados CD-ROMs.
     + El proyecto Debian continúa trabajando en la distribución «inestable» . La mayoría de los paquetes nuevos ó las actualizaciones son cargadas en esta distribución..

1. Estructura de GNU/Linux:
   1. Kernel, shell y file system.
   2. Paquete de software que permite diferenciar una distribución de otra.  
      **Editores de texto:** vi, emacs, joe

**Herramientas de networking:** wireshark, tcpdump  
**Paquetes de oficina:** OpenOffice  
**Interfaces gráficas:** GNOME / CINNAMON, KDE, LXDE

1. Kernel:
   1. Es el encargado de ejecutar programas y gestionar dispositivos de hardware y que el software y el hardware puedan trabajar juntos.
      * En 1991 Linus Torvalds inicia la programación de un Kernel Linux basado en Minix (clon de Unix desarrollado por Tanenbaum en 1987 con el n de crear un S.O. de uso didáctico)
      * El 5 de octubre de 1991, se anuncia la primera versión oficial de Linux
      * En 1992 se combina su desarrollo con GNU, formando GNU/Linux
      * La versión 1.0 apareció el 14 de marzo de 1994
      * Desarrolló continuado por miles de programadores alrededor del mundo
      * En julio de 1996 se lanza la versión 2.0 y se define la nomenclatura de versionado. Se desarrolló hasta febrero de 2004 y terminó con la 2.0.40
      * En enero de 1999 se lanza la versión 2.2, que provee mejoras de portabilidad entre otras y se desarrolla hasta febrero de 2004 terminando en la versión 2.2.26
      * En 2001 se lanza la versión 2.4 y se deja de desarrollar a fines del 2010 con la 2.4.37.11. Fue la que catapultó a GNU/Linux como un SO estable y robusto. Durante este periodo es que se comienza a utilizar Linux más asiduamente.
      * A fines del 2003 se lanza la versión 2.6. Que ha tenido muchas mejoras para el SO dentro de las que se destacan soporte de hilos, mejoras en la planificación y soporte de nuevo hardware.
      * El 3 de agosto de 2011 se lanza la version 2.6.39.4 anunciandose la misma desde meses previos como la última en su revisión
      * La última versión estable es la 4.18.6  (septiembre de 2018)
   2. Sus funciones más importantes son la administración de memoria, CPU y la E/S.
   3. La versión actual es la 4.18.6.  
      Esquema de versionado:  
      **A**: Denota version. Cambia con menor frecuencia.

**B**: Denota mayor revision. **Antes de la version 2.6, los**

**números impares indicaban desarrollo, los pares producción**

**C**: Denota menor revision. Solo cambia cuando hay nuevos drivers o características

**D**: Cambia cuando se corrige un grave error sin agregar nueva

funcionalidad  ( Casi no se usa en las ramas 3.x y 4.x, viéndose reflejado en C)

* 1. ..
  2. Se encuentra en el directorio /boot/
  3. Es un núcleo monolítico híbrido ya que los drivers y código del Kernel se ejecutan en modo privilegiado. Lo que lo hace híbrido es la capacidad de cargar y descargar funcionalidad a través de módulos.

1. Intérprete de comandos (Shell)):
   1. Es un software que provee una interfaz de usuario para acceder a los servicios del sistema operativo.
      * Modo de comunicación entre el usuario y el SO
      * Ejecuta programas a partir del ingreso de comandos
      * Cada usuario puede tener una interfaz o shell
      * Se pueden personalizar
      * Son programables
   2. Bourne Shell (sh), Korn Shell (ksh), Bourne Again Shell (bash)  
      La sintaxis de Bash tiene muchas extensiones que no proporciona el intérprete Bourne:
      * Operaciones matemáticas con enteros
      * Acceso a los argumentos

Pero Bourne fue el intérprete usado en las primeras versiones de Unix y se convirtió en un estándar de facto.

La principal ventaja de ksh sobre otros intérpretes de comandos tradicionales de Unix, es el uso como lenguaje de programación. desde su concepción, se le agregaron gradualmente muchas capacidades.

Entre las principales diferencias entre KornShell y la Bourne Shell tradicional se incluyen:

* + - Control de tareas, definición de alias de comandos, e historial de comandos
    - La opción de tres estilos de edición de línea de comandos basadas en vi, Emacs, y XEmacs.
    - Vectores asociativos y aritmética de coma flotante integrada
    - Extensibilidad dinámica de comandos integrados
  1. Cada Shell se encuentra en /bin/<shell>.   
     Los externos se buscan en los directorios indicados en la variable PATH.  
     Y los propios se encuentran en el mismo proceso del shell.
  2. El Shell no es parte del kernel ya que es un **programa,** que actúa como interfaz entre el usuario y el kernel. Por lo tanto se ejecuta en modo usuario.
  3. Si, si se quiere definir una nueva shell hay introducir el siguiente comando: sudo usermod -s <shell> <username>  
     Por ejemplo, si se le quiere definir zsh como shell al usuario Pastorsin

sudo usermod -s /bin/zsh Pastorsin

**\* No se necesitan permisos de root, así que cualquier usuario puede hacerlo**

1. Sistemas de Archivos:
   1. Organiza la forma en que se almacenan los archivos en los dispositivos de almacenamiento. Por lo tanto es el encargado de administrar y facilitar el uso de las memorias periféricas, ya sean secundarias o terciarias.
   2. GNU/Linux soporta desde sistemas basados en discos, como pueden ser ext2, ext3, ReiserFS, XFS, JFS, UFS, ISO9660, FAT, FAT32 o NTFS, a sistemas de ficheros que sirven para comunicar equipos en la red de diferentes sistemas operativos, como NFS (utilizado para compartir recursos entre equipos Linux) o SMB (para compartir recursos entre máquinas Linux y Windows).  
      **El file system usado por GNU/Linux es el Extended (ext2, ext3, ext4)**
   3. Para **FAT** GNU/Linux tiene soporte completo y estable de escritura y lectura, mientras que para **NTFS**, y con las últimas versiones del kernel, solo se puede acceder de manera estable en modo lectura.
   4. La estructura básica es un árbol jerárquico de archivos, bastante parecido a como se estructuran los sistemas Unix. Usa como file system a Extended (ext2, ext3, ext4). Dicho árbol rige bajo el estándar Filesystem Hierarchy Standard (**FHS**).  
      **Directorios más importantes:**
      * **/** el equivalente al C: de windows
      * **/bin** Binarios y ejecutables del sistema.
      * **/boot** Se encuentra lo más importante del sistema; el kernel y los archivos necesarios para que el sistema funcione correctamente.
      * **/dev** Archivos que se relacionan con los diferentes dispositivos que pueden estar funcionando en una PC (discos duros, terminales, sonido, video, lectores dvd/cd, etc)
      * **/etc** Uno de los más importantes, porque se encuentran todas las configuraciones del sistema.
      * **/home** Aquí cada usuario posee un directorio donde se guarda toda su configuración, los programas instalados y documentos personales.
      * **/usr** Jerarquía secundaria para datos compartidos de solo lectura (Unix system resources).
      * **/root** Similar al /home pero para el usuario root (solo puede ser visto por el administrador)
      * **/var** Archivos variables, como son logs, bases de datos, directorio raíz de servidores HTTP y FTP, colas de correo, archivos temporales, etc.
2. Particiones:
   1. Es una forma de dividir lógicamente el disco físico.
      * Partición primaria: Divisón cruda del disco. Se almacena la información en el MBR.
      * Partición extendida: Sirve para contener unidades lógicas en su interior. Solo 1 por disco. Y no se define ningún FS en ella.
      * Partición lógica: Ocupa la totalidad o parte de la partición extendida y se le define un FS. Son conectadas como una lista enlazada.
   2. hda (primer disco duro IDE).
      * hda1 (primera partición del primer disco duro IDE).
      * hda2 (segunda partición del primer disco duro IDE).
      * hda5 (primera partición lógica de una partición extendida del primer disco duro IDE)

hdb (segundo disco duro IDE).

* + - hdb1 (primera partición del segundo disco duro IDE).
    - hdb2 (segunda partición del segundo disco duro IDE).
    - hdb5 (primera partición lógica de una partición extendida del segundo disco duro IDE)

**Discos SCSI ó SATA:**

sda (primer disco duro SCSI)

* + - sda1 (primera partición del primer disco SCSI).
    - sda2 (segunda partición del primer disco SCSI).
    - sda5 (primera partición lógica del primer disco duro SCSI)
  1. Como mínimo es necesario 1 partición
     + Punto de montaje: /
     + Tipo de partición: Primaria
     + File System: ext4
     + Identificación: /dev/sda1
  2. ..
  3. Tipos de particionador:
     + Destructivos: permiten crear y eliminar particiones (**fdisk**)
     + No destructivos: permiten crear, modificar y eliminar particiones (**fips**)

La principal diferencia es que los destructivos si es que ya hay un sistema operativo antes instalado lo borra, en cambio los no destructivos te dejan reducir el tamaño de la partición en donde está el sistema operativo instalado.

1. Bootstrap:
   1. La BIOS es el sistema básico de entrada/salida (Basic Input-Output System) y ya viene incorporado a la placa base a través de la memoria flash. Es un firmware cuyo propósito es activar una máquina desde su encendido y preparar el entorno para cargar un sistema operativo en la memoria RAM y disco duro.
   2. La UEFI es una especificación **(alianza entre varias compañías con el objetivo de modernizar el proceso de arranque)** que define una interfaz entre el sistema operativo y el firmware.  La UEFI puede proporcionar menús gráficos adicionales e incluso proporcionar acceso remoto para la solución de problemas o mantenimiento.
   3. El **MBR** es el primer sector de un dispositivo de almacenamiento de datos, como un disco duro. A veces, se emplea para el arranque del sistema operativo con bootstrap, otras veces es usado para almacenar una tabla de particiones y, en ocasiones, se usa sólo para identificar un dispositivo de disco individual, aunque en algunas máquinas esto último no se usa y es ignorado.  
      El **MBC** es un pequeño código que permite arrancar el SO (La última acción de la BIOS es ejecutarlo)
   4. **GPT** especifica la ubicación y el formato de la tabla de particiones en el disco (es parte del EFI). Puede verse como una sustitución del **MBR**.
   5. La finalidad del bootloader es la de cargar una imagen de Kernel (sistema operativo) de alguna partición para su ejecución. Existen 2 modos de instalación:
      * En el MBR (puede llegar a utilizar MBR gap)
      * En el sector de arranque de la partición raíz o activa (Volume Boot Record)

**GRUB, LILO, NTLDR, GAG, YaST**

* + - La BIOS carga el programa de booteo (desde el MBR)
    - El bootloader lanzado desde el MBC carga el Kernel:
      1. Prueba y hace disponibles los dispositivos
      2. Luego pasa el control al proceso **init**
  1. .
     + 1. El BIOS realiza las tareas de inicio específicas de la plataforma de hardware.
       2. Una vez que el hardware es reconocido y se inicia correctamente, el BIOS carga y ejecuta el código de la partición de arranque del dispositivo de arranque designado (MBR),  que contiene la fase 1 de un gestor de arranque Linux. La fase 1 carga la fase 2 (la mayor parte del código del gestor de arranque). Algunos cargadores pueden utilizar una fase intermedia (conocida como la fase 1.5) para lograr esto, ya que los modernos discos de gran tamaño no pueden ser totalmente leídos sin código adicional.
       3. El gestor de arranque a menudo presenta al usuario un menú de opciones posibles de arranque. A continuación, carga el sistema operativo, que descomprime en la memoria, y establece las funciones del sistema como del hardware esencial y la paginación de memoria, antes de llamar a la función start\_kernel()
       4. La función start\_kernel() a continuación realiza la mayor parte de la configuración del sistema (interrupciones, el resto de la gestión de memoria, la inicialización del dispositivo, controladores, etc), antes de continuar por separado el proceso inactivo y planificador, y el proceso de Init (que se ejecuta en el espacio de usuario).
       5. El planificador toma control efectivo de la gestión del sistema, y el núcleo queda dormido (inactivo).
       6. El proceso Init ejecuta secuencias de comandos (Scripts) necesarios para configurar todos los servicios y estructuras que no sean del sistema operativo, a fin de permitir que el entorno de usuario sea creado y pueda presentarse al usuario con una pantalla de inicio de sesión.
  2. En el apagado, Init es llamado a cerrar toda las funcionalidades del espacio de usuario de una manera controlada, de nuevo a través de secuencias de comandos, tras lo cual el Init termina y el núcleo ejecuta el apagado.
  3. Si, ya que se pueden tener diversas imágenes del Kernel instaladas en diferentes particiones. Entonces cuando el bootloader arranque, va a cargar todas esas imágenes mostrando la opción con qué sistema operativo arrancar.

1. Archivos:
   1. Los archivos se identifican a través de un pathname
   2. .
      * **Cat:** Escribirá a la salida estándar el contenido de cada uno de los archivos dados como argumentos, en el mismo orden en el que fueron dados y obliga el uso de una opción. **Concatena archivos** y los muestra en el salida estándar
      * **More:** Muestra también todo el contenido del archivo con la diferencia con el anterior de que ‘more’ pagina los resultados. Primero mostrará por pantalla todo lo que se pueda visualizar sin hacer scroll y después, pulsando la tecla espacio avanzará de igual modo por el fichero.
   3. Noto que muestra el tipo de los archivos y su codificación de caracteres leyendo su contenido, ya que los que no tiene permiso muestra que no los puede leer.
2. Comandos:

|  |
| --- |
| mkdir ISO2017  cd ISO2017  touch iso2017-1 iso2017-2  ls  pwd  find . -name ‘iso\*’  df  who  vim iso2017-1  +++ \*i  +++ Andrés Milla  +++ \*esc  +++ :wq  tail -f iso2017-1 iso2017-2 |

* 1. **mkdir**: Crea un directorio nuevo
  2. **cd:** Cambia de directorio.
     + Con **cd -** se va al último directorio donde estuvimos parados, ya que lo guarda en una variable de entorno.
  3. **touch:** Crea un nuevo archivo si no existe, y si existe le cambia el tiempo de modificación y de acceso por el actual
     + Con **-m** solo modificamos el tiempo de modificación
     + Con **-a** solo modificamos el tiempo de acceso
     + Con **-c** aseguramos que no cree un archivo nuevo
  4. **ls:** Lista los archivos del directorio actual
     + Con **-a** muestra todos los archivos/directorios (inclusive ocultos)
     + Con **-l** muestra el listado detallado
  5. **pwd:** Muestra el path del directorio actual
  6. **find:** Encuentra archivos o directorios
     + **-name ‘iso\*’**: Muestra todos los archivos/directorios que contengan dicho nombre. Se pueden hacer operaciones booleanas.
     + **-iname ‘iSo\*’**: No toma en cuenta mayúsculas ni minúsculas
     + Muchísimos más potentes: <https://www.linuxtotal.com.mx/index.php?cont=info_admon_022>
  7. **df:** Muestra la cantidad de espacio libre en el FS
     + $ df -h    # formato humano
     + $ df -m    # formato en megabytes
     + $ df -k    # formato en kb, este es el **default**
     + -Th muestra el tipo de FS
  8. **who:** Puede listar los nombres de los usuarios conectados actualmente, su terminal, el tiempo que han estado conectados, y el nombre del host desde el que se han conectado.
     + **-d** muestra los procesos muertos
     + **-b** muestra la hora del último arranque del sistema.
  9. ..
  10. **tail:** Sirve para mostrar en pantalla las últimas líneas de un archivo. Por defecto se muestran las últimas 10 líneas, pero este número puede variar dependiendo de las especificaciones del usuario.
      + **-n:** Permite especificar el número de líneas que hay que mostrar.
      + **-f:** Muestra las últimas líneas del archivo.

1. / **12.** Comandos 2:
   1. **Shutdown:** Reinicia, detiene el sistema, apaga y manda mensajes a los usuarios.
      * -r : Reiniciar después del shutdown
      * -P: Apaga después del shutdown
      * -H: Halt después del shutdown
      * -k : Envía advertencias
      * Ejemplo:  shutdown -P 22:22 chau! (Apaga la pc a las 22:22 hs. Mostrando un mensaje “chau!”)
      * /sbin/shutdown
   2. **Reboot / Halt: Halt**comprueba  que el sistema  está siendo desconectado en el archivo /var/log/wtemp, y entonces llama al kernel para detener, reiniciar o apagar el sistema. Si se llama a **halt o reboot** cuando el sistema no está en los niveles 0 o 6, será invocado shutdown (con el flag -h o -r).
      * -f : Fuerza el halt o el reinicio, no llama a shutdown
      * /sbin/reboot
   3. **Halt:**
      * /sbin/halt
   4. **Locate:** Encuentra los archivos de lo que le mandamos como parámetro. La búsqueda la hace en una base de datos indexada para aumentar significativamente la velocidad de respuesta. Esto quiere decir, que locate realmente no busca en el disco del sistema, sino que en un archivo con la lista de todos los archivos que existen en el GNU/Linux.
      * -i : Ignora mayúsculas y minúsculas
      * /usr/bin/locate
   5. **Uname:** Muestra la información del sistema.
      * -a: Imprime toda la información
      * /bin/uname
   6. **Dmesg:** Se muestra los mensajes del kernel
      * /bin/dmesg
   7. **Lspci:**Lista todos los componentes tipo PCI
      * Ejemplo: lspci **| grep** -r Audio(detecta la tarjeta de audio)
      * /usr/bin/lspci
   8. **At:** Nos permite programar tareas que se ejecutarán una única vez.
      * -v : Nos devuelve la fecha en que programamos la tarea
      * -l : Lista todas las tareas junto con su ID
      * -d ID: Elimina la tarea
      * Ejemplo: at -v 3am May 18
      * /usr/bin/at
   9. **Netstat:** Muestra información estadística y estado actual de conexiones de red, protocolos, puertos, zócalos y dispositivos.
      * /bin/netstat
   10. **Mount:** Sin argumentos veremos un listado de lo que está montado en el sistema y la carpeta sobre la que lo hace.
       * -a : Monta todos los sistemas  ficheros que aparezcan en fstab,  los marcados como 'noauto', los excluidos por el parámetro -t y los que ya están montados.
       * Ejemplo: sudo mount /dev/sdb1 /media/usb (monta un usb en la carpeta usb de /media)
       * /bin/mount
   11. **Unmount:** Cuando un sistema de archivos ya no se necesita, hay que desmontarlo de la estructura de directorios utilizando el comando umount. Este comando permite desmontar un sistema de archivos montado previamente.
       * umount /dev/fd1 (desmonta la unidad de disquete B)
       * umount –t vfat (desmonta todas los sistemas montados del tipo vfat)
       * umount -a (desmonta todo)
       * /bin/umount
   12. **Head:** Muestra las primeras líneas de un archivo o lista de archivos
       * -n: Permite especificar el número de líneas que hay que mostrar.
       * -q: Evita que se muestren los títulos de los archivos especificados.
       * /usr/bin/head
   13. **Losetup:** Maneja loop devs (un dispositivo loop es un dispositivo de bloques virtual que apunta a un fichero normal y corriente de nuestro sistema de ficheros. Esto significa que podemos dar a un fichero la apariencia de ser un disco duro para el resto del sistema.  En Linux, estos dispositivos son /dev/loop0, /dev/loop1, etc… Su uso más habitual siempre ha sido montar ficheros ISO (imágenes de CD y DVD) mediante el comando mount -o loop, pero también se puede usar para manipular ficheros como si fuesen un disco duro; particionarlos, formatearlos y montar estos sistemas de ficheros entre otras cosas.)
       * /sbin/losetup
   14. **Write:** Permite mandar un mensaje a otro usuario del sistema especificando como parámetros el usuario al que enviar el mensaje y la TTY asociada.
       * /usr/bin/write
   15. **Mkfs:**Se utiliza para dar formato a un dispositivo de almacenamiento de bloque con un determinado sistema de archivos.
       * -v: Producir una salida detallada, incluyendo todas las órdenes específicas del sistema de archivos que se ejecutan.
       * -t: Especifica el tipo de sistema de archivos que se construirá. Si no se especifica, se utiliza el tipo de sistema de archivos por defecto (actualmente ext4).
       * Ejemplo: mkfs -t ext4 /dev/sdd1
       * /sbin/mkfs
   16. **Fdisk:** Te permite crear particiones en el disco duro (**destructivo**)
       * -d : Borra una partición
       * -l: Lista los tipos de partición
       * -m: Menú
       * -t: Cambia el tipo de la partición
       * /sbin/fdisk