PRACTICA 1

¿Qué es GNU?

Es un Sistema Operativo tipo Unix (Unix like), pero libre.

* S.O. diseñado por miles de programadores
* S.O. gratuito y de libre distribución (se baja desde la Web, CD, etc.)
* Existen diversas distribuciones (customizaciones)
* Es código abierto, lo que nos permite estudiarlo, personalizarlo, auditarlo, aprovecharnos de la documentación, etc...
* Permite a los usuarios tener la libertad de usar, estudiar, distribuir y mejorar los programas

GNU/Linux al ser un software libre posee las siguientes características:

* Puede ser usado, copiado, estudiado, modificado y redistribuido libremente.
* Generalmente es de costo nulo ← Es un gran error asociar el software libre con el software gratuito ← Pensar en software gratis que se distribuye con restricciones
* Es común que se distribuya junto con su código fuente (código abierto)
* Corrección más rápida ante fallas
* Características que se refieren a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software

*Breve historia*

*Fue iniciado por Richard Stallman en 1983 con el fin de crear un Unix libre (el sistema GNU). En 1985 Richard Stallman creó la Free Software Foundation (FSF o Fundación para el Software Libre) para proveer soportes logísticos, legales y financieros al proyecto GNU. Esta fundación contrato programadores, aunque una porción sustancial del desarrollo fue (y continúa siendo) producida por voluntarios.*

*En 1990, GNU ya contaba con un editor de textos (Emacs), un compilador (GCC) y gran cantidad de bibliotecas que componen un Unix típico pero faltaba el componente principal, el núcleo (Kernel).*

*Linus Torvalds ya venía trabajando desde 1991 en un Kernel denominado Linux, el cual se distribuía bajo licencia GPL. Múltiples programadores se unieron a Linus en el desarrollo, colaborando a través de Internet y consiguiendo paulatinamente que Linux llegase a ser un núcleo compatible con UNIX. En 1992, el núcleo Linux fue combinado con el sistema GNU, resultando en un SO libre y completamente funcional. El SO formado por esta combinación es usualmente conocido como "GNU/Linux" o como una "distribución Linux" y existen diversas variantes.*

Características más importantes de GNU/Linux:

* Es multiusuario
* Es multitarea y multiprocesador
* Es altamente portable
* Posee diversos intérpretes de comandos, de los cuales algunos son programables
* Permite el manejo de usuarios y permisos
* Todo es un archivo (hasta los dispositivos y directorios)
* Cada directorio puede estar en una partición diferente (/tmp, /home, etc.)
* Es case sensitive
* Es código abierto

¿Qué es POSIX?

POSIX consiste en una familia de estándares especificadas por la IEEE con el objetivo de facilitar la interoperabilidad de sistemas operativos. Además, POSIX establece las reglas para la portabilidad de programas. Por ejemplo, cuando se desarrolla software que cumple con los estándares POSIX existe una gran probabilidad de que se podrá utilizar en sistemas operativos del tipo Unix. Si se ignoran tales reglas, es muy posible que el programa o librería funcione bien en un sistema dado pero que no lo haga en otro.

Es un conjunto de estándares que define varias interfaces de herramientas, comandos y API para Sistemas Operativos similares a UNIX y otros; una norma escrita por la IEEE, que define una interfaz estándar del sistema operativo y el entorno, incluyendo un intérprete de comandos (shell)

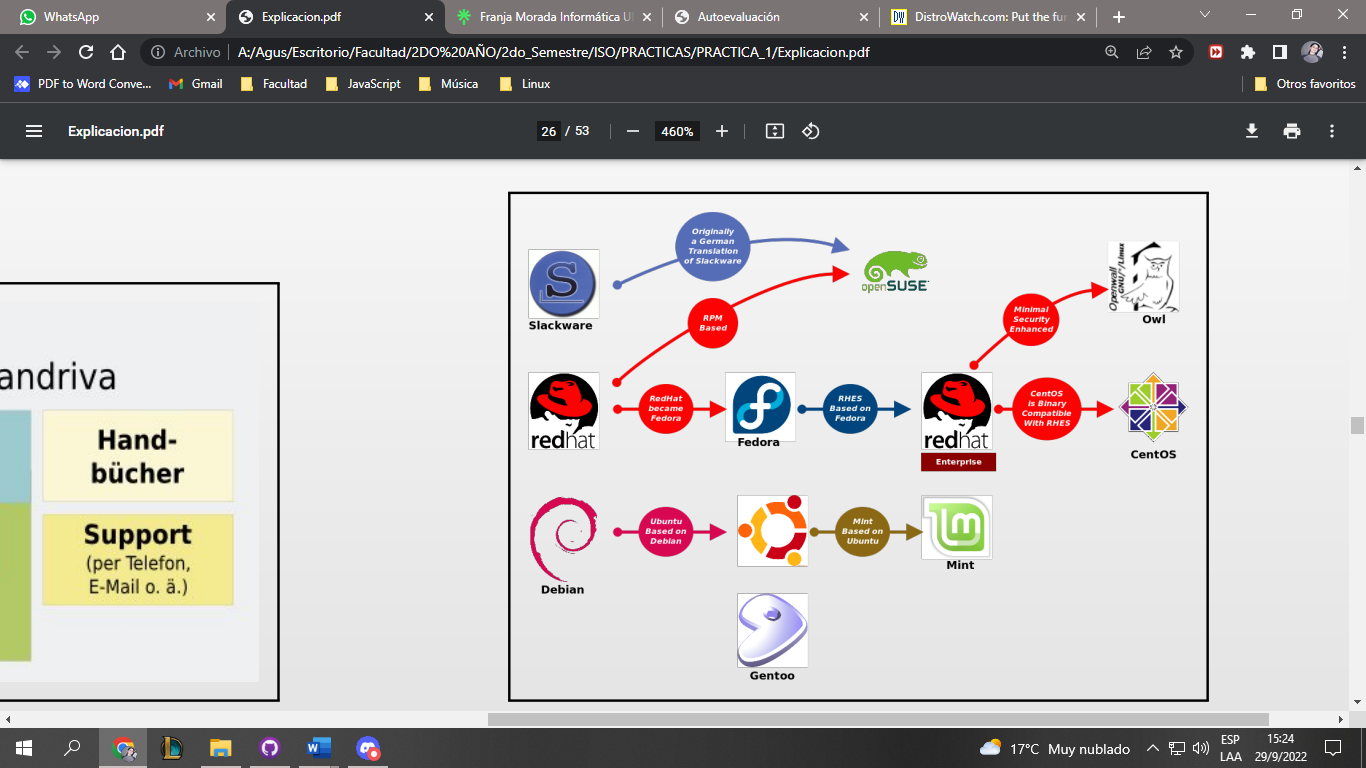
Distribuciones

Son un conjunto de aplicaciones reunidas que permiten brindar mejoras para instalar fácilmente un sistema operativo basado en GNU/Linux. Son "sabores" de GNU/Linux que, en general, se diferencian entre sí por las herramientas para configuración y sistemas de administración de paquetes de software para instalar. La elección de una distribución depende de las necesidades del usuario y de gustos personales.

Una distribución es una customización de GNU/Linux formada por una versión de kernel y determinados programas con sus configuraciones

Existen distribuciones que incluyen paquetes propietarios y otras que son 100% libres.

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.



Estructura GNU/Linux

3 componentes fundamentales de GNU/Linux.

* El kernel (núcleo)
* El Shell (interprete de comandos)
* El FileSystem (sistema de archivos)

**La estructura básica del S.O es el Kernel que:**

* Ejecuta programas y gestiona dispositivos de hardware
* Es el encargado de que el software y el hardware puedan trabajar juntos
* Sus funciones más importantes son la administración de memoria, CPU y la E/S
* En sí, y en un sentido estricto, es el sistema operativo
* Es un núcleo monolítico híbrido:
  + 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.
* Esta licenciado bajo la licencia GPL v2

**Kernel**

El kernel (también conocido como núcleo) es la parte fundamental de un sistema operativo. El kernel o núcleo de linux se podría definir como el corazón de este sistema operativo. Es, a grandes rasgos, el encargado de que el software y el hardware de una computadora puedan trabajar juntos.

La versión estable actual (según la practica) es 5.8.X.

Antes de la serie de Linux 2.6.x, los números pares indicaban la versión “estable” lanzada. Por ejemplo, una para uso de fabricación, como el 1.2, 2.4 ó 2.6. Los números impares, en cambio, como la serie 2.5.x, son versiones de desarrollo, es decir que no son consideradas de producción.

Comenzando con la serie Linux 2.6.x, no hay gran diferencia entre los números pares o impares con respecto a las nuevas herramientas desarrolladas en la misma serie del kernel. Linus Torvalds dictaminó que este será el modelo en el futuro.

Nomenclatura versionado:

A.B.C[.D]

A: Denota versión. Cambia con menor frecuencia.

B: Denota mayor revisión.

C: Denota menor revisión. 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

**¿Es posible tener más de un Kernel de GNU/Linux instalado en la misma máquina?**

Se pueden tener varios instalados (Versión anterior o algún kernel virtual), pero en realidad estaría funcionando solo uno, ya que es la parte de un sistema operativo que administra y controla los recursos y procesos

**¿Dónde se encuentra ubicado dentro del File System?**

En /boot. El primer sector del disco se llama boot sector. Contiene información general de donde se almacena el Kernel y como se arranca

**Shell**

El Shell (intérprete de comandos) es el programa que recibe lo que se escribe en la terminal y lo convierte en instrucciones para el sistema operativo. Un intérprete de comandos es un programa que lee las entradas del usuario y las traduce a instrucciones que el sistema es capaz de entender y utilizar.

* También conocido como CLI (Command Line Interface).
* 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. Esta se define por defecto (si no se especifica) pero puede ser personzalida.
  + Un usuario puede cambiar su propio shell a cualquier cosa: la cual, sin embargo, debe estar listada en el archivo/etc/shells.
  + Solo root puede ejecutar un shell que no esté incluido en el archivo/etc/shells.
  + Si una cuenta tiene un shell de inicio de sesión restringido, solo root puede cambiar el shell de ese usuario.
  + Al crear cuentas de usuario con las utilidades useradd o adduser, la marca --shell se puede usar para especificar el nombre del shell de inicio de sesión de un usuario que no sea el especificado en los archivos de configuración respectivos.
* Se pueden personalizar.
* Son programables.
* Bourne Shell (sh), Korn Shell (ksh), Bourne Again Shell (bash)(autocompletado, history, alias).

La shell no forma parte del kernel básico del SO; sino que la misma “dialoga” con el kernel.

No es muy difícil darse cuenta de por qué el shell no es parte del kernel. Como el shell se usa para interpretar las órdenes del usuario y ejecutarlas, si el mismo estuviese en el kernel tendría acceso a instrucciones propias que usa el SO para la gestión de los distintos dispositivos del hardware. Razón por la cual se abstrae al usuario del manejo de dispositivos hardware, dejándolo al kernel.

**¿Dónde se ubican (path) los comandos propios y externos al Shell?**

El shell nos permite ejecutar:

Comandos externos, por ejemplo: ls, cat, mkdir, etc.

* son programas ajenos al Shell
* cuando se lanzan inician un nuevo proceso
* se buscan en los directorios indicados en la variable PATH

Comandos internos (builtin commands), por ejemplo: cd, bg, alias, eval, exec, pwd, etc.

* se ejecutan en el mismo proceso del shell, sin lanzar un nuevo proceso

La diferencia fundamental es que los internos están incorporados a la consola y se pueden ejecutar directamente, mientras que para los externos hay que indicar la ruta hasta la ubicación del comando.

Para los comandos externos puede ser que no tengamos que indicar la ruta hasta la ubicación del mismo de forma explícita, si esta ruta está incluida en la variable de entorno PATH.

También debemos tener precaución en el caso de que el comando exista tanto de forma interna y externa, ya que las dos versiones del comando pueden dar resultados distintos, por lo que si queremos estar seguros de que estamos ejecutando la versión externa debemos indicar la ruta del comando (p.e. pwd ó /bin/pwd)

Los comandos internos son nativos del shell de linux que estemos usando (bash por ejemplo). Estos se suelen encontrar en el directorio /usr/bin. Los externos se encuentran en la variables $PATH, no son nativos del shell de Linux.

**FileSystem**

El Filesystem es la forma en que dentro de un SO se organizan y se administran los archivos. Esa administración comprende:

* Métodos de acceso: cómo se acceden los datos contenidos en el archivo.
* Manejo de archivos: cómo actúan los mecanismos para almacenar, referenciar, compartir y proteger los archivos.
* Manejo de la memoria secundaria: Cómo se administra el espacio para los archivos en memoria secundaria.
* Mecanismos de integridad: con qué métodos se garantiza la incorruptibilidad del archivo.

El adoptado por GNU/Linux es el Extended (v2, v3, v4).

Linux no permite ver por defecto el contenido de las particiones de Windows. Para poder hacerlo será necesario que montemos la partición NTFS o FAT en la que está Windows. Aunque en estos momentos existen distribuciones de GNU-Linux que pueden realizar operaciones de lectura y escritura sobre ellas.

La estructura de archivos es una estructura jerárquica en forma de árbol invertido, donde el directorio principal (raíz) es el directorio "/", del que cuelga toda la estructura del sistema. Este sistema de archivos permite al usuario crear, borrar y acceder a los archivos sin necesidad de saber el lugar exacto en el que se encuentran. No existen unidades físicas, sino archivos que hacen referencia a ellas.

/bin: En donde se residen los comandos principales de Linux como ls o mv. Estan los comandos que pueden usar todos los usuarios (incluido el root).

/boot: Aquí se encuentran los cargadores de inicio y los archivos de inicio del sistema.

/dev: En esta ruta se encuentran montados todos los dispositivos físicos como el USBs o el DVDs.

/etc: Contiene la configuración de los paquetes instalados.

/home: Los usuarios del sistema tendrán su carpeta personal para colocar todas sus carpetas adicionales con su nombre como se muestra a continuación: /home/likegeeks.

/lib: Aquí se guardan las librerías de los paquetes instalados ya que estas librerías son compartidas por todos los paquetes. A diferencia de Windows, puedes encontrar duplicados en diferentes carpetas.

/media: En esa ruta se encuentran los dispositivos externos como los DVDs ylos pendrives USB, desde aquí puedes acceder a sus archivos desde aquí.

/mnt: Aquí se montan otras cosas como localizaciones de red y algunas distribuciones que puedas tener montadas en un pendrive o DVD.

/opt: Algunos paquetes opcionales se encuentran aquí y esta ruta es administrada por el administrador de paquetes.

/proc: Debido a que todo en Linux es un archivo, esta es una carpeta que tiene los procesos ejecutándose en el sistema, y puedes acceder a ellos para obtener información acerca de los procesos actuales.

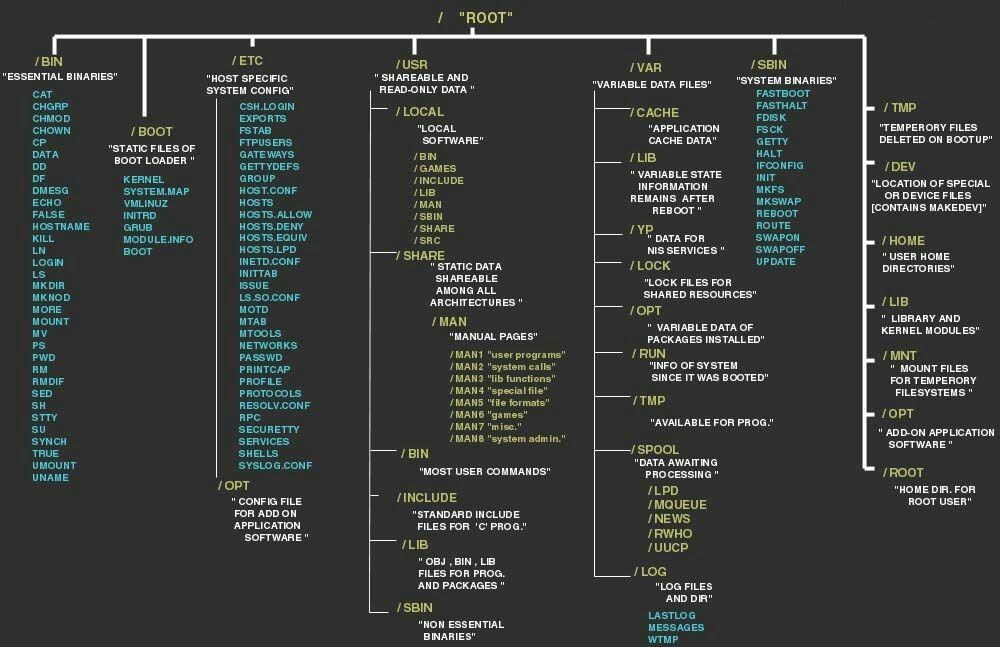
/root: La carpeta home para el usuario root.

/sbin: Como /bin, pero con archivos binarios solo para el usuario root.

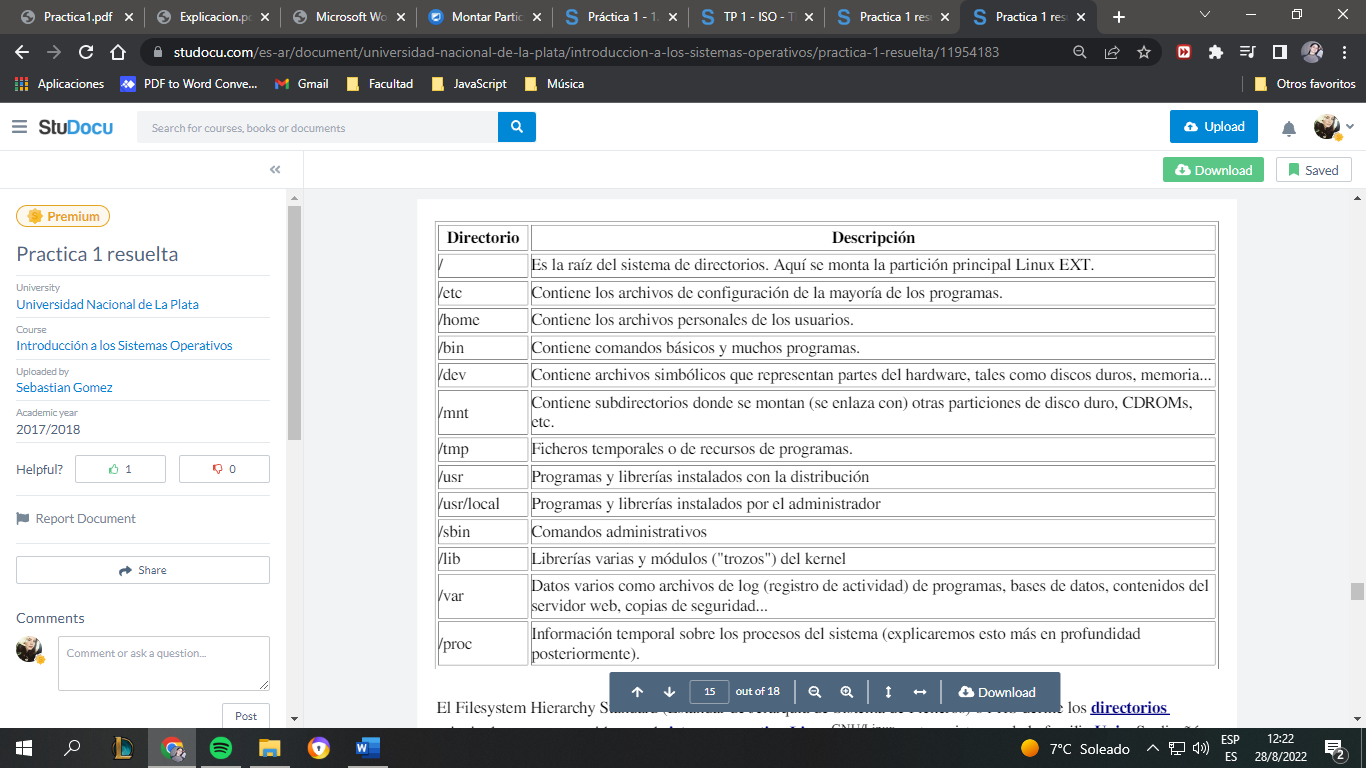
/tmp: Contiene los archivos temporales.

/usr: Aquí es donde las utilidades y los archivos se comparten entre usuarios en Linux.

/var: Contiene registros del sistema y otros datos variables



**ANOTACION: /dev/null es un archivo especial que descarta toda la información que se escribe en o se redirige hacia él. A su vez, no proporciona ningún dato a cualquier proceso que intente leer de él, devolviendo simplemente un EOF o fin de archivo.**



PARTICIONES

Una partición de disco, en mantenimiento, es el nombre genérico que recibe cada división presente en una sola unidad física de almacenamiento de datos. Toda partición tiene su propio sistema de archivos (formato); generalmente, casi cualquier sistema operativo interpreta, utiliza y manipula cada partición como un disco físico independiente, a pesar de que dichas particiones estén en un solo disco físico.

Ventajas y deventajas:

-Es una buena práctica separar los datos del usuario de las aplicaciones y/o Sistema Operativo instalado

-Tener una partición de Restore de todo el sistema

-Poder ubicar el Kernel en una partición de solo lectura, o una que nisiquiera se monta (no está disponible a los usuarios).

-Particionar demasiado un disco puede tener desventajas. (Al achicar el tamaño del disco rígido físico, puede ocurrir que archivos grandes no entren en el tamaño de una de las particiones nuevas).

Debido al tamaño acotado en el MBR para la tabla de particiones:

• Se restringe a 4 la cantidad de particiones primarias

• 3 primarias y una extendida con sus respectivas particiones lógicas

• Una de las 4 particiones puede ser extendida, la cual se subdivide en volúmenes lógicos

Partición primaria: división cruda del disco (puede haber 4 por disco). Se almacena información de la misma en el MBR.

Partición extendida: sirve para contener unidades lógicas en su interior. Solo puede existir una partición de este tipo por disco. No se define un tipo de FS directamente sobre ella.

Partición lógica: ocupa la totalidad o parte de la partición extendida y se le define un tipo de FS. Las particiones de este tipo se conectan como una lista enlazada.

**¿Cómo se identifican las particiones en GNU/Linux? (Considere discos IDE, SCSI y SATA).**

* Disqueteras
  + Primera disquetera: **/dev/fd0** (en Windows sería la disquetera A:)
  + Segunda disquetera: **/dev/fd1**
* Discos duros (en general: **/dev/hdx#,** donde x es el disco y # es la partición)
  + Primer disco duro: (todo el disco) **/dev/hda**
    - Particiones primarias
      * Primera partición primaria: **/dev/hda1**
      * Segunda partición primaria: **/dev/hda2**
      * Tercera partición primaria: **/dev/hda3**
      * Cuarta partición primaria: **/dev/hda4**
    - Particiones lógicas
      * Primera partición lógica: **/dev/hda5**
      * Sucesivamente: **/dev/hda#**
  + Segundo disco duro: (todo el disco**) /dev/hdb**
    - Particiones primarias
      * Primera partición primaria: **/dev/hdb1**
      * Segunda partición primaria: **/dev/hdb2**
      * Tercera partición primaria: **/dev/hdb3**
      * Cuarta partición primaria: **/dev/hdb4**
    - Particiones lógicas
      * Primera partición lógica: **/dev/hdb5**
      * Sucesivamente: **/dev/hdb#**
* Discos SCSI
  + Primer disco SCSI: **/dev/sda**
  + Segundo disco SCSI: **/dev/sdb**
    - Sucesivamente ...

Primer CD-ROM SCSI: /dev/scd0, también conocido como **/dev/sr0**

Para instalar GNU/Linux como mínimo es necesario una partición (para el /). Es recomendable crear al menos 2 (/ y SWAP). Usualmente se suelen tener tres, una para el sistema/programas (/), otra para los datos (/home) y otra para swap.

Si no has separado /home del resto del sistema, cuando vayas a instalar una nueva versión, perderás toda esa documentación.

Punto de montaje / 🡪 Primaria. Sistema de ficheros ext4 transaccional.

**¿Qué tipo de software para particionar existe? Menciónelos y compare.**

Existen 2 tipos:

* Destructivos: permiten crear y eliminar particiones (fdisk)
* No destructivo: permiten crear, eliminar y modificar particiones (fips, gparted) ← generalmente las distribuciones permiten hacerlo desde la interfaz de instalación

Arranque (bootstrap) de un Sistema Operativo:

En las arquitecturas x86 la BIOS de la motherboardes un chip especial que guarda configuración inicial de la computadora.

Su función principal es la de iniciar los componentes de hardware y lanzar el sistema operativo de un ordenador cuando lo encendemos (a través del MBC). También carga las funciones de gestión de energía y temperatura del ordenador.

Cuando enciendes tu ordenador lo primero que se carga en él es el BIOS. Este firmware entonces se encarga de iniciar, configurar y comprobar que se encuentre en buen estado el hardware del ordenador, incluyendo la memoria RAM, los discos duros, la placa base o la tarjeta gráfica. Cuando termina selecciona el dispositivo de arranque (disco duro, CD, USB etcétera) y procede a iniciar el sistema operativo, y le cede a él el control de tu ordenador.

**EFI**

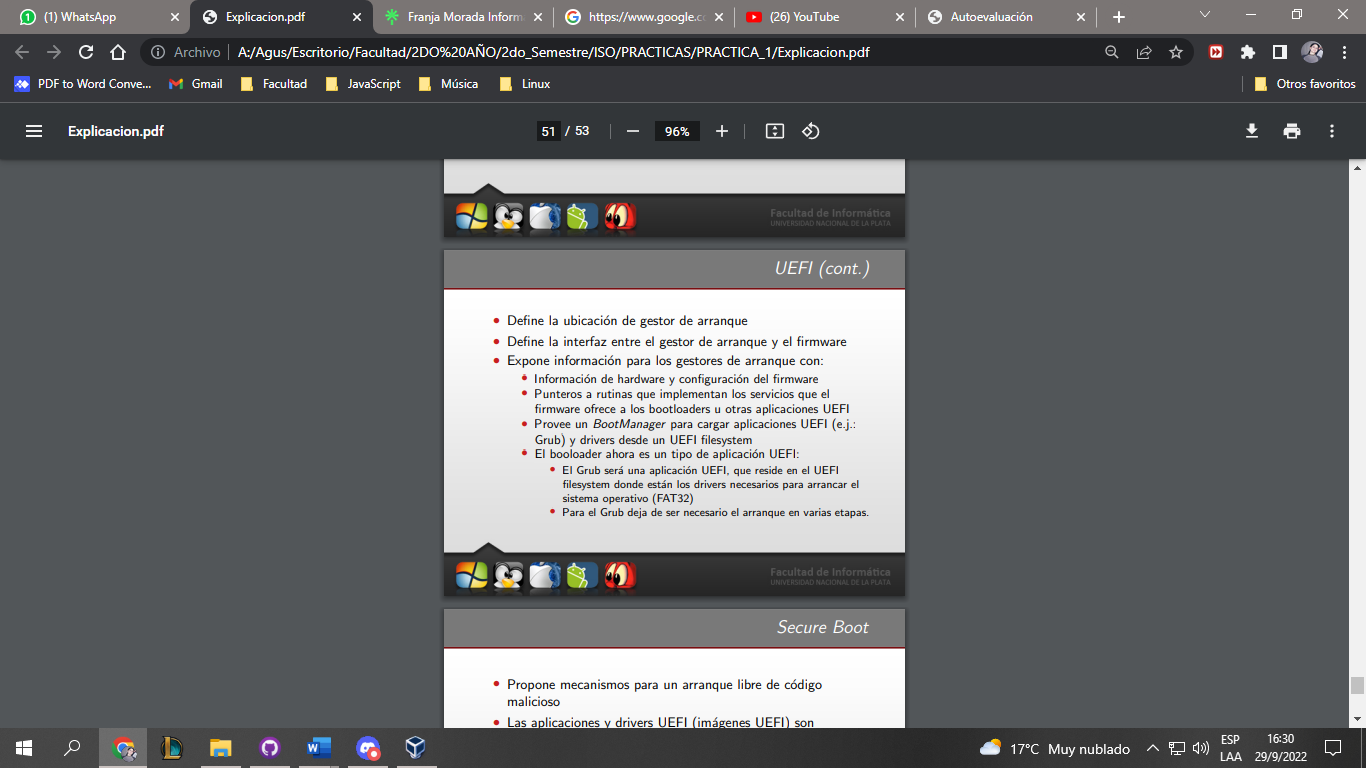
**EFI** es nexo entre el SO y el firmware:

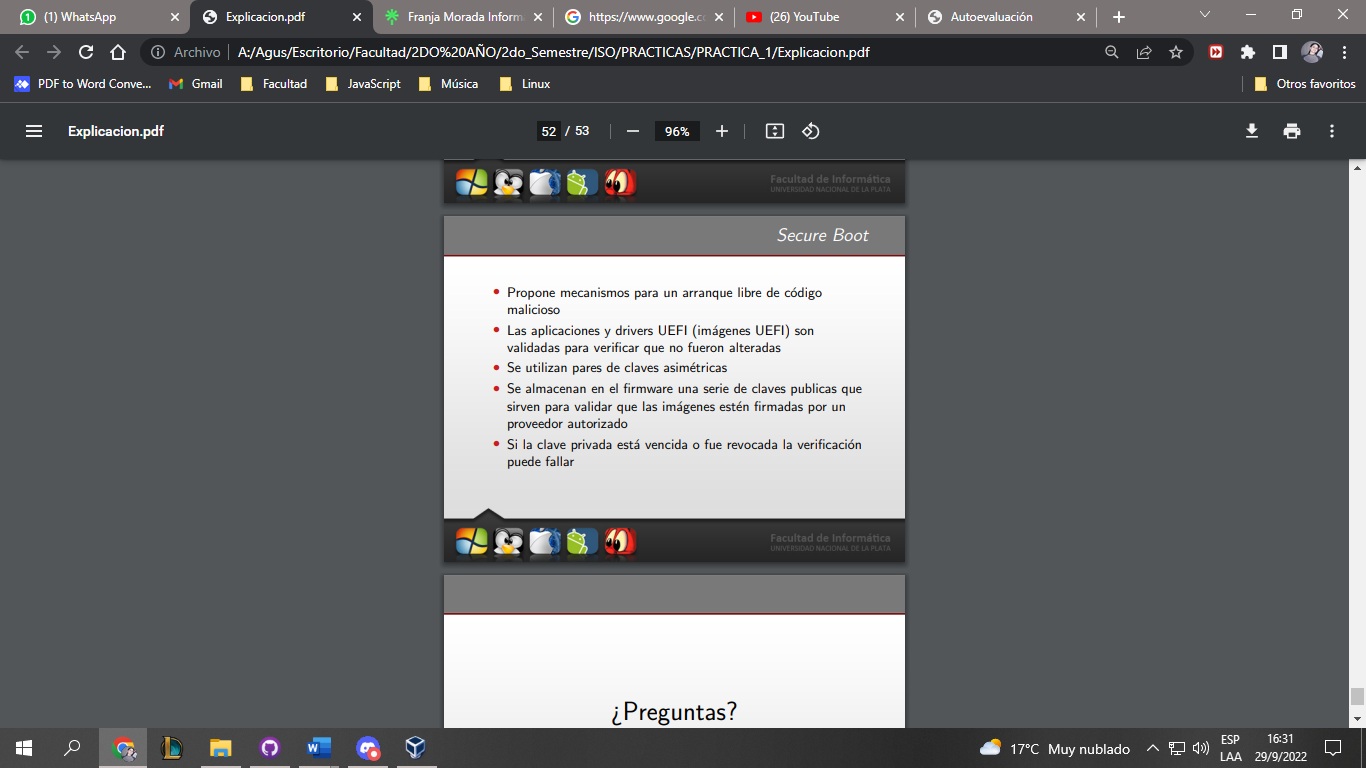
* Utiliza el sistema GPT (GUID partition table) para solucionar limitaciones del MBR, como la cantidad de particiones.
* GPT especifica la ubicación y formato de la tabla de particiones en un disco duro.
* Es parte de EFI. Puede verse como una sustitución del MBR.
* La especificación EFI es propiedad de Intel
* Alternativa para reemplazar la BIOS
* Se pueden crear hasta 128 particiones
* Se mantiene un MBR para tener compatibilidad con el esquema BIOS
* GPT usa un modo de direccionamiento lógico (logical block addressing LBA) en lugar de cylinder-header-sector. GPT puede verse como una sustitución del MBR.
* El MBR “heredado” se almacena en el LBA 0.
* En el LBA 1 está la cabecera GPT. La tabla de particiones en sí está en los bloques sucesivos
* La cabecera GPT y la tabla de particiones están escritas al principio y al final del disco (redundancia)

**UEFI**

A mediados de la década pasada las empresas tecnológicas se dieron cuenta de que el BIOS estaba quedándose obsoleto, y 140 de ellas se unieron en la fundación **UEFI** para renovarla y reemplazarla por un sistema más moderno. En esencia, todo lo que hace el BIOS lo hace también la UEFI. Pero también tiene otras funciones adicionales y mejoras sustanciales, como una interfaz gráfica mucho más moderna, un sistema de inicio seguro, una mayor velocidad de arranque o el soporte para discos duros de más de 2 TB

* Alianza entre varias compañías con el objetivo de modernizar el proceso de arranque.
* UEFI es propiedad del UEFI Forum.
* Define la interfaz entre el gestor de arranque y el firmware más sencilla y fácil de comprender.
* Aporta mayor velocidad en el arranque de los equipos
* UEFI mejora la seguridad con su funcionalidad Secure Boot





**El MBR** (master boot record) es el primer sector del disco (cilindro 0, cabeza 0, sector 1). Esto se carga a memoria y se ejecuta. Es un registro de arranque principal que puede contener un código de arranque denominado MBC (master boot code) y una marca de 2 bytes que indica su presencia o puede solamente contener la tabla de particiones. En el último caso el BIOS ignora este MBR.

Existe un MBR en todos los discos y si existiese más de un disco rígido en la máquina, sólo uno es designado como Primary Master Disk. El tamaño del MBR coincide con el tamaño estándar de sector: 512 bytes.

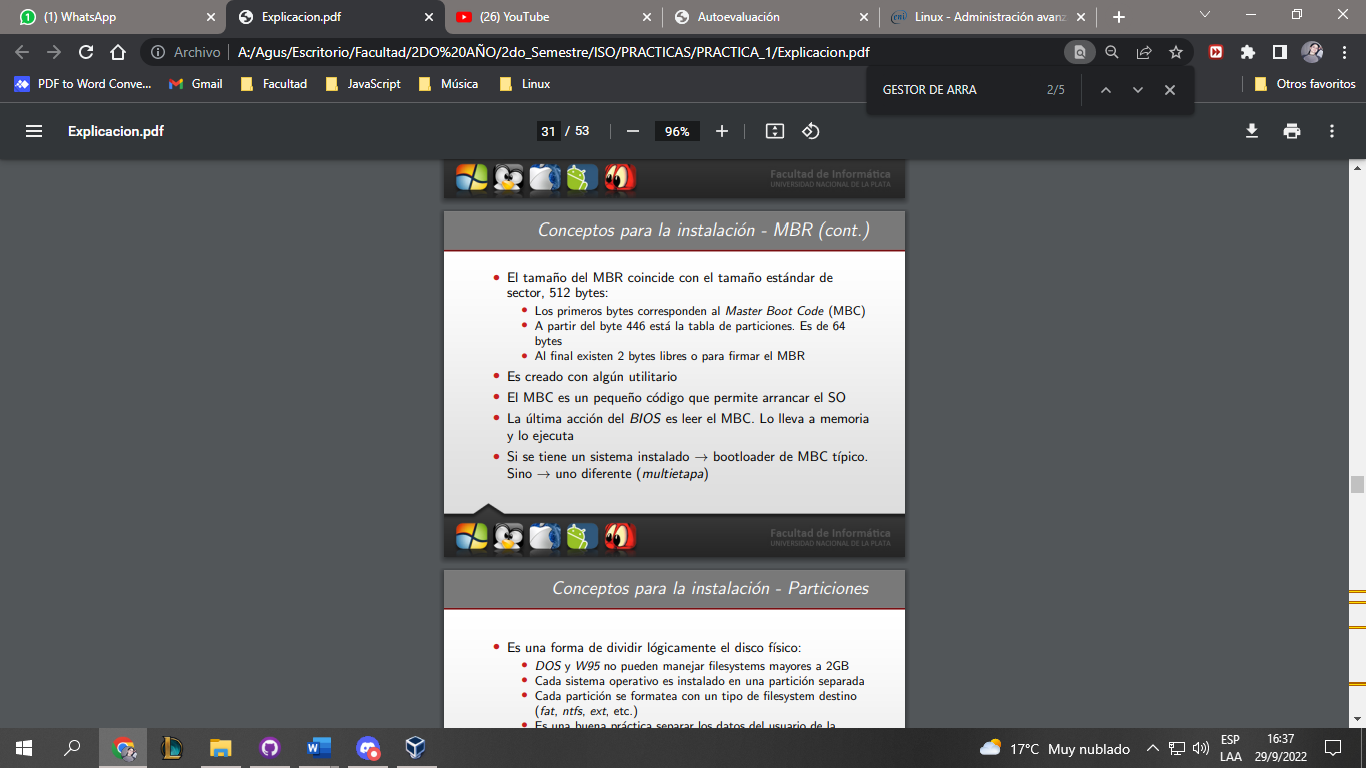
El MBC es un pequeño código que permite arrancar el SO. La última acción del BIOS es leer el MBC, lo lleva a memoria y lo ejecuta.

- 1. Se ejecuta el código de la BIOS

- 2. El hardware lee el sector de arranque

- 3. Se carga el gestor de arranque

- 4. Se carga el kernel



Un gestor de arranque (en inglés «bootloader») es un programa sencillo que no tiene la totalidad de las funcionalidades de un sistema operativo, y que está diseñado exclusivamente para preparar todo lo que necesita el sistema operativo para funcionar. Normalmente se utilizan los cargadores de arranque multietapas, en los que varios programas pequeños se suman los unos a los otros, hasta que el último de ellos carga el sistema operativo.

En los ordenadores modernos, el proceso de arranque comienza cuando la unidad central de procesamiento ejecuta los programas contenidos en una memoria de sólo lectura en una dirección predefinida y se configura la unidad central para ejecutar este programa, sin ayuda externa, al encender el ordenador.

Habitualmente se instala en el MBR y asume el rol de MBC. En una computadora en la que hay sólo un sistema operativo, no hay referencias a pantalla generalmente. Si hay un gestor de arranque, este programa nos permitirá elegir el sistema operativo a arrancar. El código del MBC de Windows, por ejemplo, busca en la tabla de particiones cual es la primer partición primaria con el flag de “booteo” activo y transfiere el control al código que se encuentra al comienzo de dicha partición: el PBR (partition boot record).

En el caso del sistema operativo Linux, se puede optar por distintos gestores de arranque, por ejemplo, LILO (Linux Loader), GRUB (Grand Unified Bootloader) o GAG (Gestor de arranque Gráfico). LILO no se basa en un sistema de archivos específico. Funciona en una variedad de sistemas de archivos. GRUB en cambio, debe comprender el sistema de archivos y el formato de los directorios2. GRUB tiene algunas ventajas con respecto a LILO: tiene una línea de comandos interactiva como, permite arrancar desde una red, y podría considerarse más seguro

**¿Es posible tener en una PC GNU/Linux y otro Sistema Operativo instalado? Justifique.**

Si, ya que un disco rígido puede particionarse y en cada partición tener un sistema de archivos distinto (partición primaria), sería como tener varios discos distintos, uno con cada SO, por lo tanto, se necesitará un gestor de arranque como los descriptos arriba.

Archivos

Como en Windows, se puede emplear un cierto criterio de "tipo" para marcar las distintas clases de ficheros empleando una serie de caracteres al final del nombre que indiquen el tipo de fichero del que se trata. Así, los ficheros de texto, HTML, las imágenes PNG o JPEG tienen extensiones .txt, .htm (o .html), .png y .jpg (o .jpeg) respectivamente. Pese a esto Linux sólo distingue tres tipos de archivos:

* Archivos o ficheros ordinarios, son los mencionados anteriormente.
* Directorios (o carpetas), es un archivo especial que agrupa otros ficheros de una forma estructurada.
* Archivos especiales, son la base sobre la que se asienta Linux, puesto que representan los dispositivos conectados a un ordenador, como puede ser una impresora. De esta forma introducir información en ese archivo equivale a enviar información a la impresora. Para el usuario estos dispositivos tienen el mismo aspecto y uso que los archivos ordinarios

Comandos

Cat: El comando ‘cat’ imprimirá por pantalla el contenido del fichero sin ningún tipo de paginación ni posibilidad de modificarlo. Básicamente concatena archivos o la salida estándar en la salida estándar. Podemos pasarle parámetros como

More: Al igual que ‘cat’, ‘more’ permite visualizar por pantalla el contenido de un fichero de texto, con la diferencia con el anterior de que ‘more’ página 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.

Less: El comando ‘less’ es el más completo de los tres, pues puede hacer todo lo que hace ‘more’ añadiendo mayor capacidad de navegación por el fichero (avanzar y retroceder) además de que sus comandos están basados en el editor ‘vi’, del cual se diferencia en que no tiene que leer todo el contenido del fichero antes de ser abierto. Tiene una gran cantidad de opciones y parámetros, como siempre lo recomendable:

**(a) shutdown**

El comando shutdown se utiliza para apagar o reiniciar Linux desde la terminal.

Detener el sistema de forma segura.

shutdown [OPTIONS] [TIME] [MESSAGE]

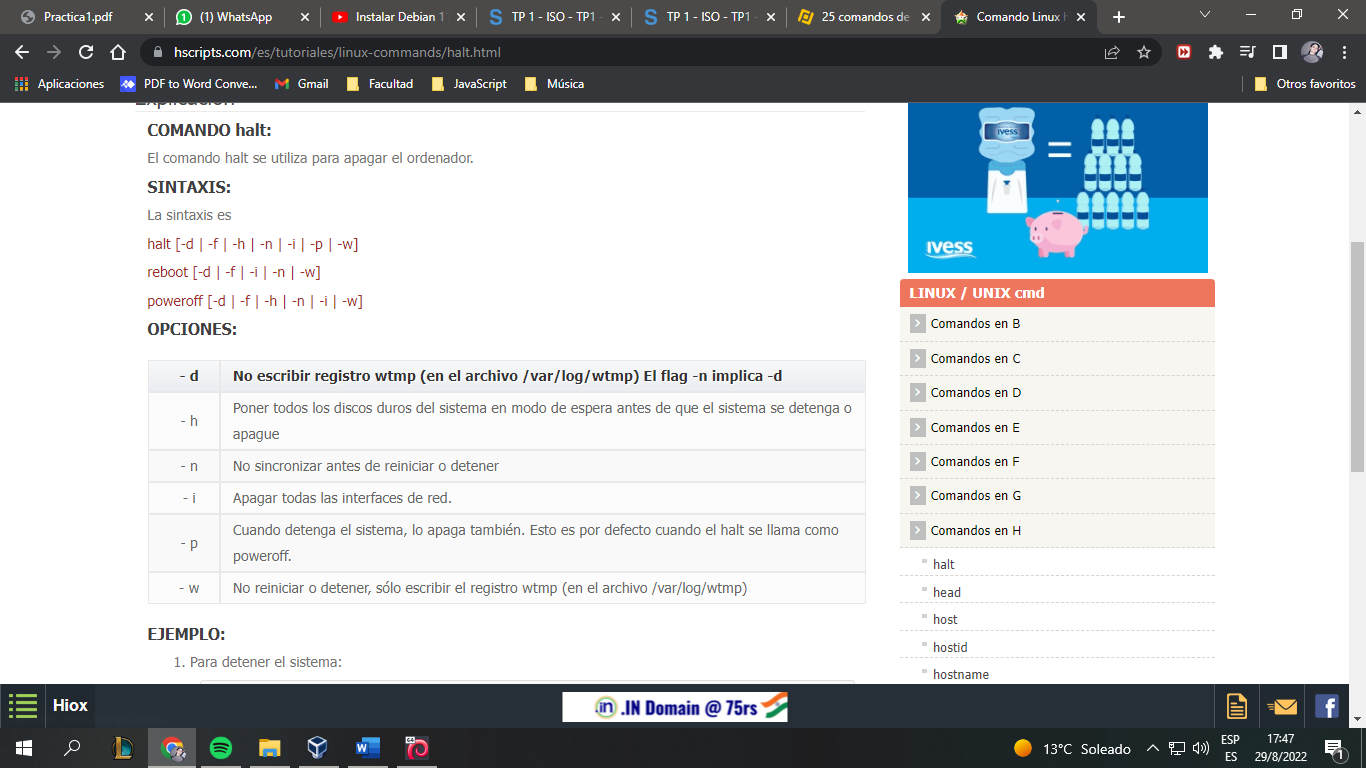
**(b) reboot**

Reinicia SO

-reboot, -r: reinicia el sistema

**(c) halt**

El comando halt se utiliza para apagar el ordenador



**(d) locate**

El comando locate es una alternativa útil, ya que es más rápido que find para realizar búsquedas. Eso se debe a que sólo escanea tu base de datos de Linux en lugar de todo el sistema

locate [my-file]

**(e) uname**

Si utilizamos el comando sin argumentos nos entregara la palabra Linux extraída de la información del Kernel.

$ uname

Si deseamos extraer la información de la versión del kernel utilizamos el parámetro -r.

$ uname -r

Si deseamos extraer la fecha de cuando la versión del kernel fue liberada utilizamos el parámetro -v.

$ uname -v

**(f) dmesg**

Se usa para examinar o controlar el ring buffer del kernel. La acción predeterminada es mostrar todos los mensajes del ring buffer.

Debido a toda la información desplegada, es difícil llevar a cabo alguna tarea de administración allí. Podemos hacer uso del parámetro “-H” con el fin de indicarle a dmesg que la salida sea legible para los usuarios, lo cual simplificará las tareas de soporte. Allí encontramos detalles mucho más claros sobre el anillo del kernel.

Otra alternativa para realizar un análisis con dmesg es con el parámetro “-w”, el cual nos permite escribir un script para analizar el resultado usando una expresión regular con el fin de filtrar los eventos para su posterior análisis:

**(g) lspci**

lspci es un comando para los sistemas operativos Unix-like que imprime listas con información detallada sobre todos los Buses y dispositivos del sistema

**(h) at**

Permite programar tareas únicas en nuestro sistemas GNU/Linux.

N os permite programar tareas para que se ejecuten a determinada fecha y hora .

at [hora] [fecha]

El comando at, nos puede ser útil para apagar el sistema a una hora especifica, realizar una copia de seguridad única, enviar un correo electrónico como recordatorio a la hora especificada, entre otras muchas cosas.

**(i) netstat**

netstat (estadísticas de red) es una herramienta de línea de comandos que muestra las conexiones de red (entrantes y salientes), tablas de enrutamiento y una serie de estadísticas de interfaz de red.

**(j) mount**

El mandato mount ordena al sistema operativo que haga que un sistema de archivos esté disponible para su utilización en una ubicación determinada (el punto de montaje).

**(k) umount**

Este comando permite desmontar un sistema de archivos montado previamente.El uso del comando umount garantiza que toda la información mantenida en memoria por el sistema operativo se escriba en el dispositivo antes de desmontarlo

Sintaxis: umount dispositivo | punto\_montaje

**(l) head**

El comando head muestra de modo predeterminado las diez primeras líneas de un archivo. Se puede modificar esta opción a las N primeras líneas del archivo con la sintaxis head -nN.

**(m) losetup**

se usa para asociar loop devices con archivos regulares o block devices, también para desacoplar loop devices, y para hacer queries del status de un loop device. (dispositivo iterador = loop device)

**(n) write**

El comando write permite mandar un mensaje a otro usuario del sistema especificando como parámetros el usuario al que enviar el mensaje y la TTY asociada:

$ write usuario tty.

Para finalizar la escritura del mensaje y enviarlo presionamos CTRL + D. La TTY se especificará cuando el usuario al que enviemos el mensaje tenga más de una sesión abierta. En el caso de que no le especificamos, se enviará automáticamente a la tty del usuario con actividad más reciente

**(ñ) mkfs**

se utiliza para dar formato a un dispositivo de almacenamiento de bloque con un determinado sistema de archivos.

**(o) fdisk (con cuidado)**

es una utilidad de línea de comandos basada en texto para ver y administrar particiones de disco duro en Linux. Con fdisk puedes ver, crear, cambiar el tamaño, eliminar, cambiar, copiar y mover particiones.más info de Fdisk: https://maslinux.es/comando-fdisk-para-administrar-particiones-de-disco-en-gnulinux/