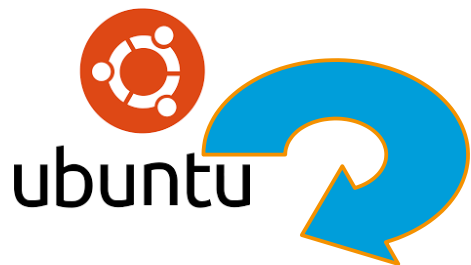


# Instalación y configuración de Linux

**Sistemas Informáticos**

# Instalación y configuración de Linux



distribución de GNU/Linux basada en Debian, que incluye principalmente software libre y de código abierto. Puede utilizarse en ordenadores y servidores.

## Requisitos mínimos:

- Un ordenador con un procesador de doble núcleo de 2 GHz.
- Al menos 25 GB de espacio en el disco duro.
- Al menos 1 GB de RAM para crear un servidor.

<https://ubuntu.com/download/desktop>

## Ubuntu 22.04.1 LTS

The latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years of free security and maintenance updates, guaranteed until April 2027.

### Recommended system requirements:

- |                                       |  |
|---------------------------------------|--|
| ✓ 2 GHz dual-core processor or better | ✓ Internet access is helpful                               |
| ✓ 4 GB system memory                  | ✓ Either a DVD drive or a USB port for the installer media |
| ✓ 25 GB of free hard drive space      |  |

# Gestor de arranque de Linux

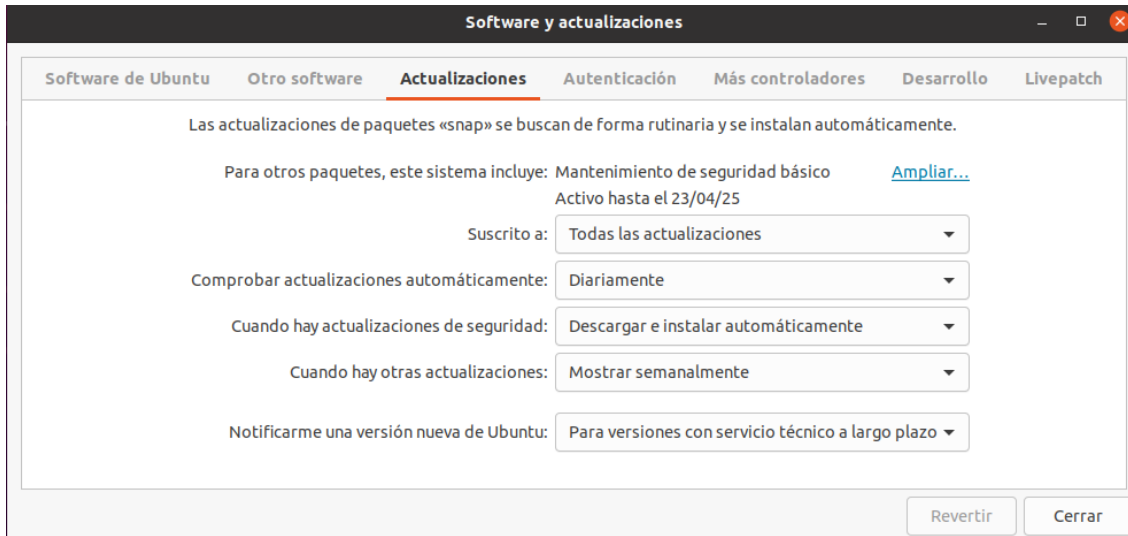
Linux normalmente emplea el gestor de arranque **GRUB2 (GRUB Legacy)**. Es muy potente y flexible, puede lanzar la mayoría de sistemas operativos.

Los archivos/directorios que intervienen en su funcionamiento y gestión son:

- 1) El archivo de configuración principal se encuentra en **/boot** o **/boot/default/** y es **grub.cfg**. (**NO MODIFICAR DIRECTAMENTE**). Para que las modificaciones tengan efecto se hace uso de la orden **/usr/sbin/update-grub** o mediante la actualización de Kernel.
- 2) La configuración del menú gráfico durante el arranque se gestiona mediante el archivo **/etc/default/grub**. Se puede modificar el tiempo de espera del menú, la selección por defecto de este, establecimiento de password, etc., y se edita con privilegios de root.
- 3) En el directorio **/etc/grub.d/** se encuentran los siguientes scripts numerados (que se leen en ese orden:
  - a) **00\_header**: contiene y carga la información básica del GRUB desde **/etc/default/grub**
  - b) **05\_debian\_theme**: configuración imagen de fondo, color de texto, etc.
  - c) **10\_Linux**: localiza el kernel
  - d) **20\_memtest86+**: localiza y añade al menú el programa de testeo de memoria **/boot/memtest86+.bin**
  - e) **30\_os-prober**: busca otros sistemas operativos instalados en el disco y los añade al menú.
  - f) **40\_custom**: archivo donde el usuario puede agregar entradas nuevas.

# Actualizaciones en Ubuntu Desktop

Ubuntu actualiza tanto el SW del sistema (drivers incluidos) como las aplicaciones. Ubuntu utiliza “Software y actualizaciones”



Tres tipos de actualizaciones:

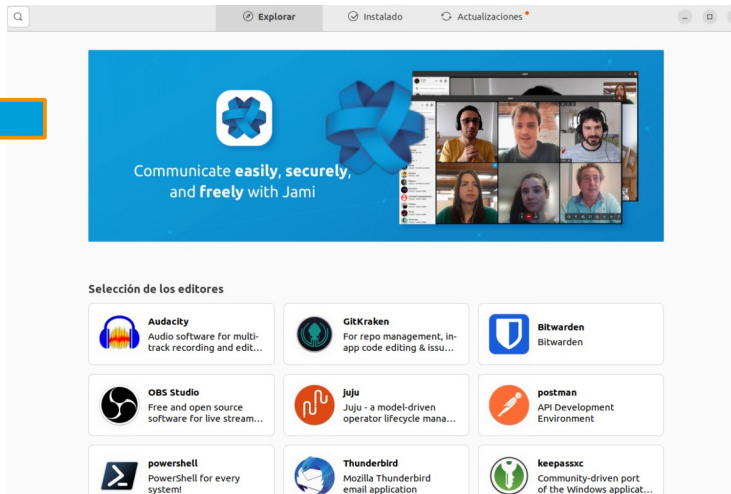
- **De seguridad:** Ubuntu recomienda tener siempre activas.
- **Recomendadas:** Los desarrolladores de paquetes son partidarios de activarlas.
- **Sin asistencia técnica:** Actualizaciones de paquetes no soportadas por Ubuntu.

Se puede seleccionar la frecuencia de comprobación de actualizaciones y las acciones a realizar para cuando se detectan estas. Podemos habilitar Notificaciones de nuevas versiones, actualizaciones, etc.

# Software de Ubuntu

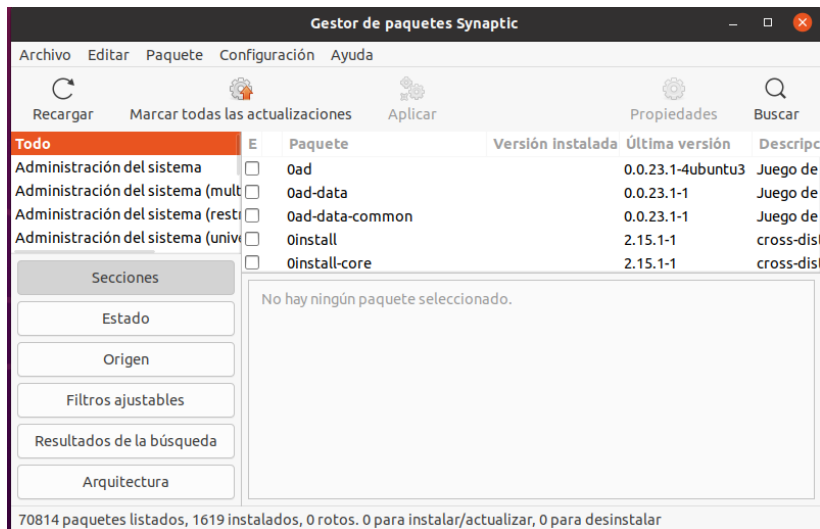
En software de Ubuntu se puede buscar software por categorías o realizar una búsqueda. Además, en la parte superior podemos encontrar el software instalado y las actualizaciones.

También encontramos los tipos de repositorios (almacenes de paquetes SW y aplicaciones): main, universo, restricted y multiverse . Se recomienda tener activos para la detección e instalación de nuevo SW y actualizaciones.

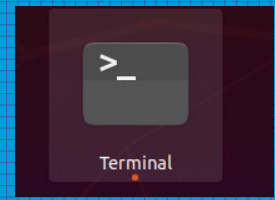


# Software : Synaptic

Una aplicación con entorno gráfico ampliamente utilizada para la gestión de paquetes es Synaptic, la cual podemos descargar desde el propio centro de “Software Ubuntu”. Resulta una herramienta muy potente y versátil.



# Software : Por comandos



Una de las herramientas más utilizadas en línea de texto es apt (Advanced Packaging Tool). Synaptic se basa en él. Su uso en terminal se realiza con privilegios de administrador (sudo). Las tareas más destacadas son:

- ***sudo apt update***: Actualización de paquetes disponibles. Actualiza el índice de repositorios de nuestro sistema.
- ***sudo apt install nombre\_paquete***: instalación de paquetes. Se puede dejar espacios entre paquetes si se instalan varios.
- ***sudo apt remove nombre\_paquete***: desinstalar paquetes.
- ***sudo apt upgrade***: actualización del sistema.

```
alex@alex-VirtualBox: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
alex@alex-VirtualBox:~$ sudo apt update  
[sudo] contraseña para alex:  
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease  
Des:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Des:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]  
Des:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]  
Descargados 336 kB en 1s (461 kB/s)  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se pueden actualizar 10 paquetes. Ejecute «apt list --upgradable» para verlos.  
alex@alex-VirtualBox:~$
```


# Gestión de Archivos. Línea de comandos

Los comandos Linux siguen esta sintaxis: **comando [opciones] [argumentos]**

Uno de los comando más utilizados es ls (list) → **ls [opciones][ficheros]**

Sus opciones más utilizadas son:

- l: muestra en formato largo
- t: ordena por fecha de modificación
- r: invierte el orden de salida
- R: lista recursivamente el contenido de cada directorio
- i: muestra el número de i-nodo
- a: muestra los archivos ocultos. En Linux son aquellos que empiezan por punto “.”
- h: muestra el tamaño de cada fichero en K, M, G etc.
- S: lista los archivos ordenados por tamaño.



```
alex@alex-VirtualBox:~/Documentos/prueba$ ls -l
total 16
drwxrwxr-x 2 alex alex 4096 mar 12 17:58 prueba1
drwxrwxr-x 2 alex alex 4096 mar 12 17:58 prueba2
drwxrwxr-x 2 alex alex 4096 mar 12 17:58 prueba3
drwxrwxr-x 2 alex alex 4096 mar 12 17:58 prueba4
alex@alex-VirtualBox:~/Documentos/prueba$
```

- 1 Tipo (d,l, -,c,b) más permisos por usuario grupos y otros
- 2 Número de enlaces duros
- 3 y 4 propietario y grupo
- 5 tamaño en bytes
- 6 fecha modificación/creación
- 7 nombre del fichero



# Gestión de Archivos. Características de los ficheros

Los nombres de los ficheros Linux:

- pueden tener una extensión entre 1 y 255 caracteres.
- No se puede emplear el carácter “/”
- No es recomendable usar los caracteres especiales como =, ^, ?, & <, >, etc... ya que suelen tener significado especial para el terminal.
- Puede contener espacios (si lo pasamos como argumento entre comillas para que lo considere dos argumentos).
- No tienen por qué usar extensiones (pero recomendable)
- Los archivos ocultos van tras un punto .oculto

Archivos y directorios se localizan dentro del árbol de directorios por su ruta.

Cada directorio incluye al menos dos entradas ocultas: “.” que es el directorio actual y “..” el directorio padre definidas automáticamente al crear el directorio.

# Gestión de Archivos. Características de los ficheros

Los tipos de ficheros son muy importantes. Cualquier elemento físico (impresora, disco, etc.) o lógico (directorio, enlace, etc.) se representa en Linux mediante un archivo. Distinguimos cuatro tipos:

- 1) Regulares
- 2) Directorios
- 3) Enlaces:
  - a) Enlaces duros
  - b) Enlaces simbólicos o blandos
- 4) Dispositivos:
  - a) Dispositivos por caracteres
  - b) Dispositivos por bloques

Al hacer `ls` normalmente muestra los siguientes colores por la opción “`--color`” que viene por defecto activada:

Blanco: Archivo regular.

Verde: ejecutable

Azul directorio

Cian: enlace simbólico.

Rojo: enlace roto.

# Gestión de Archivos. Eliminar archivos

## **rm [-irf] lista\_de\_ficheros**

Las opciones:

- i: solicita confirmación antes de realizar la acción.
- r o R: eliminación recursiva sobre directorios.
- f: fuerza la eliminación aun estando protegido el archivo contra escritura

Caracteres comodín:

- \* hace referencia a cualquier cadena de caracteres (Ejemplo: elimina todos los archivos del directorio actual: **rm -r \***)
- ? solo referencia a un único carácter cualquiera (Ejemplo: Elimina archivos que empiecen por ejer seguidos de cualquier número: **rm -i ejer?**)

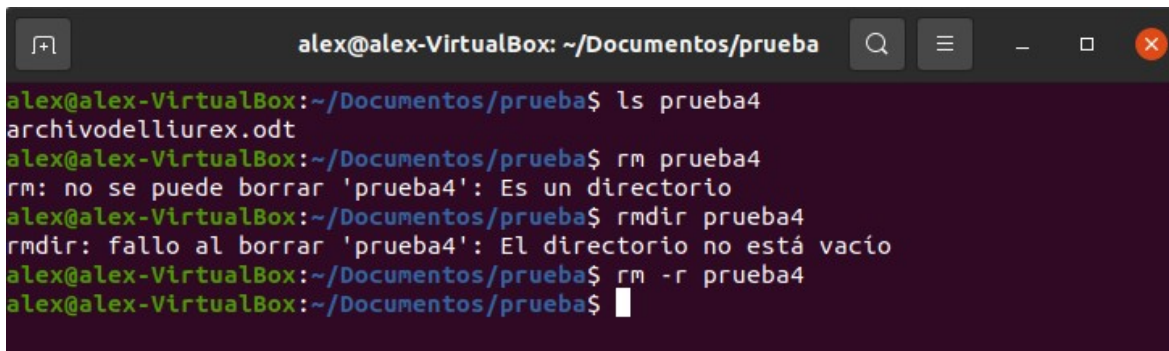
# Gestión de Archivos. Creación y eliminación de directorios

**mkdir lista\_de\_directorio**

**rmdir lista\_de\_directorio**

```
alex@alex-VirtualBox:~/Documentos/prueba$ rmdir prueba3  
alex@alex-VirtualBox:~/Documentos/prueba$
```

La eliminación de directorios se puede efectuar mediante la orden `rm -r`, si dispone de contenido, o empleando el comando `rmdir`, si el directorio se encuentra vacío.



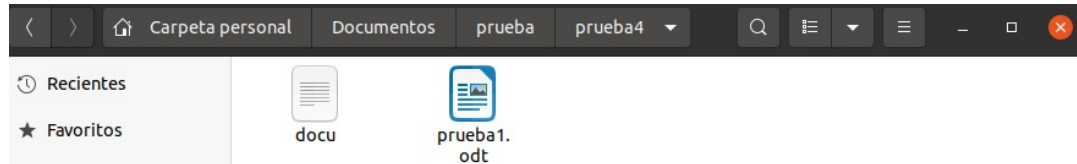
```
alex@alex-VirtualBox: ~/Documentos/prueba  
alex@alex-VirtualBox:~/Documentos/prueba$ ls prueba4  
archivodelliurex.odt  
alex@alex-VirtualBox:~/Documentos/prueba$ rm prueba4  
rm: no se puede borrar 'prueba4': Es un directorio  
alex@alex-VirtualBox:~/Documentos/prueba$ rmdir prueba4  
rmdir: fallo al borrar 'prueba4': El directorio no está vacío  
alex@alex-VirtualBox:~/Documentos/prueba$ rm -r prueba4  
alex@alex-VirtualBox:~/Documentos/prueba$
```

# Gestión de Archivos. Copia de archivos

## **cp [-irR] lista\_archivos\_origen destino**

- i: solicita confirmación antes de sobrescribir un archivo con el mismo nombre en el destino.
- r o R: copia de forma recursiva en directorios

Tenemos estos archivos dentro de prueba4, copiamos docu en prueba3 que está también dentro de la carpeta prueba



```
alex@alex-VirtualBox:~/Documentos/prueba$ cp prueba4/docu prueba3/
alex@alex-VirtualBox:~/Documentos/prueba$ cd prueba3
alex@alex-VirtualBox:~/Documentos/prueba/prueba3$ ls
docu
alex@alex-VirtualBox:~/Documentos/prueba/prueba3$ cd .
alex@alex-VirtualBox:~/Documentos/prueba/prueba3$ cd ..
alex@alex-VirtualBox:~/Documentos/prueba$ cd prueba4
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ ls
docu  prueba1.odt
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$
```

# Gestión de Archivos. Mover (renombrar) archivos

## **mv [-iu] lista\_archivos\_origen destino**

Sus opciones más utilizadas son:

- i: se emplea para asegurarse que en el destino no exista un archivo con el mismo nombre, ya que se sobrescribiría.
- u: solo mueve archivos o directorios con el mismo nombre entre origen y destino si estos son más actuales en el origen.

```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ mv prueba1.odt ../prueba3
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ cd ..
alex@alex-VirtualBox:~/Documentos/prueba$ cd prueba3
alex@alex-VirtualBox:~/Documentos/prueba/prueba3$ ls
docu  prueba1.odt
alex@alex-VirtualBox:~/Documentos/prueba/prueba3$
```

# Gestión de Archivos. Impresión de archivos

Hay muchos comandos para inspeccionar el contenido de un archivo. Para poder interpretarlos deben de estar en formato ASCII. Los comandos más frecuentes son:

**cat [lista\_archivos] [ {>|>>|<|<<}**

**archivo]**

Concatena la lista de archivos, mostrándolos por pantalla. Su salida se puede redireccionar mediante:

- > sobrescribe si existe el archivo o, en caso contrario, lo crea.
- >> añade el contenido de un fichero existente o, en caso contrario, lo crea

```
alex@alex-VirtualBox:~/Documentos/prueba$ cat > texto1
uno dos tres
cuatro cinco seis
alex@alex-VirtualBox:~/Documentos/prueba$ cat texto1
uno dos tres
cuatro cinco seis
```

```
alex@alex-VirtualBox:~/Documentos/prueba$ cat > texto2
siete ocho nueve
diez once doce
alex@alex-VirtualBox:~/Documentos/prueba$ cat texto1 texto2 > texto12
alex@alex-VirtualBox:~/Documentos/prueba$ cat texto12
uno dos tres
cuatro cinco seis
siete ocho nueve
diez once doce
```

# Gestión de Archivos.Impresión de archivos

Otros comandos muy utilizados son:

**more [lista\_archivos]:**

**less [lista\_archivos]**

Permiten visualizar el contenido por páginas y se puede navegar de unas a otras. less es más potente, ideal para archivos grandes.

Para obtener el principio o final de una archivo podemos usar:

**head [-n] lista\_ficheros**

**tail [-n] lista\_ficheros**

Se emplean para visualizar las n primeras (head) o últimas (tail) líneas de uno o varios ficheros de texto. (por defecto 10 líneas, si no indicamos nada)



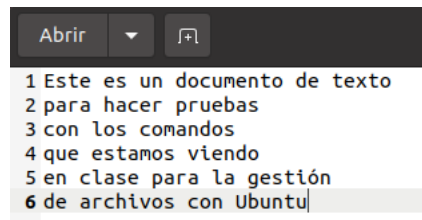
# Gestión de Archivos. Cuentéo de un fichero

Con wc podemos mostrar el número de líneas, palabras y número de bytes de un fichero.

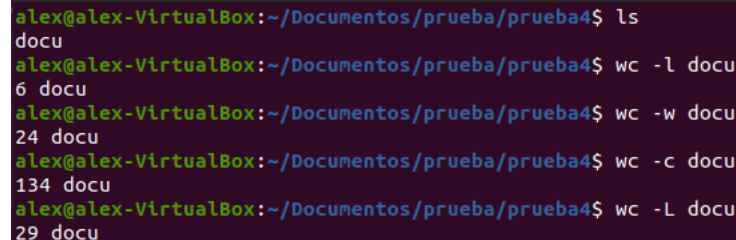
## **wc [-lwcL] lista de ficheros**

Podemos seleccionar algunos contadores de forma aislada o conjunta:

- l: número de líneas.
- w: número de palabras
- c: número de bytes
- L: longitud de la línea más larga.



```
Abrir ▼ [icon]  
1 Este es un documento de texto  
2 para hacer pruebas  
3 con los comandos  
4 que estamos viendo  
5 en clase para la gestión  
6 de archivos con Ubuntu
```



```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ ls  
docu  
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ wc -l docu  
6 docu  
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ wc -w docu  
24 docu  
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ wc -c docu  
134 docu  
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ wc -L docu  
29 docu
```

# Gestión de Archivos. Ordenación de un fichero.

La orden `sort` permite mostrar de forma ordenada las líneas de un fichero. Sigue el orden ASCII donde las minúsculas van antes que las mayúsculas.

**`sort [-fnru] [-t <delimitador>] [-k <num_campo>] lista_archivos`**

Las opciones más importantes son:

- `f`: ignoras las mayúsculas y las minúsculas
- `r`: invierte el orden
- `n`: ordena numéricamente en lugar de alfabéticamente
- `u`: elimina entradas repetidas
- `t` delimitador: indica el delimitador o separador de campos.
- `k`: indica el número de campo por el que se va a realizar la ordenación. Por defecto, toma los espacios, tabuladores, y carácter de fin de línea como delimitadores de campo.

```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ sort -f -t: -k2 docu2
María:Adell:34
Julia:Ferrer:24
Antonio:Font:13
Pedro:Martínez:19
JuanJo:Monfort:47
Jose:Renau:45
Adela:Suárez:34
```

docu2

---

```
1 Pedro:Martínez:19
2 Adela:Suárez:34
3 Jose:Renau:45
4 JuanJo:Monfort:47
5 Antonio:Font:13
6 María:Adell:34
7 Julia:Ferrer:24S
```

# Gestión de Archivos.Entradas y salidas estándar. Redirecciones.

Los comandos y los procesos que se generan de ellos reciben la información mediante un flujo de entrada de información y los resultados son enviados mediante un flujo de salida. El SO asigna automáticamente a cada comando entradas y salidas estándar asociadas a los flujos de entrada y salida, respectivamente.

- Entrada estándar: `stdin` (número o descriptor 0) Nutre al comando de información para su ejecución. (normalmente asociada al teclado)
- Salida estándar: `stdout` (número o descriptor 1) transmite el resultado (asociada con la pantalla)
- Salida de errores estándar: `stderr` (número o descriptor 2) si durante la ejecución del comando se produce un error o un aviso se envía a esta salida.

# Gestión de Archivos. Entradas y salidas estándar. Redirecciones de la salida estándar.

Podemos usar `>` para volcar la salida de una orden sobre un fichero: **orden > fichero**

Ejemplo: `sort fichero1 > fichero2`

Si el fichero no existe se crea uno nuevo, si existe sobrescribe su contenido. Es equivalente a usar `1>` (1 hace referencia al descriptor del fichero)

Podemos usar `>>` para no sobrescribir si no añadir el contenido al final. Ejemplo: `sort fichero1 >> fichero2`

```
1 Pedro:Martínez:19
2 Adela:Suárez:34
3 Jose:Renau:45
4 JuanJo:Monfort:47
5 Antonio:Font:13
6 María:Adell:34
7 Julia:Ferrer:245
```

```
docu
1 Este es un documento de texto
2 para hacer pruebas
3 con los comandos
4 que estamos viendo
5 en clase para la gestión
6 de archivos con Ubuntu
```

```
1 Este es un documento de texto
2 para hacer pruebas
3 con los comandos
4 que estamos viendo
5 en clase para la gestión
6 de archivos con Ubuntu
7 Adela:Suárez:34
8 Antonio:Font:13
9 Jose:Renau:45
10 JuanJo:Monfort:47
11 Julia:Ferrer:245
12 María:Adell:34
13 Pedro:Martínez:19
```

```
1 Adela:Suárez:34
2 Antonio:Font:13
3 con los comandos
4 de archivos con Ubuntu
5 en clase para la gestión
6 Este es un documento de texto
7 Jose:Renau:45
8 JuanJo:Monfort:47
9 Julia:Ferrer:245
10 María:Adell:34
11 para hacer pruebas
12 Pedro:Martínez:19
13 que estamos viendo
```

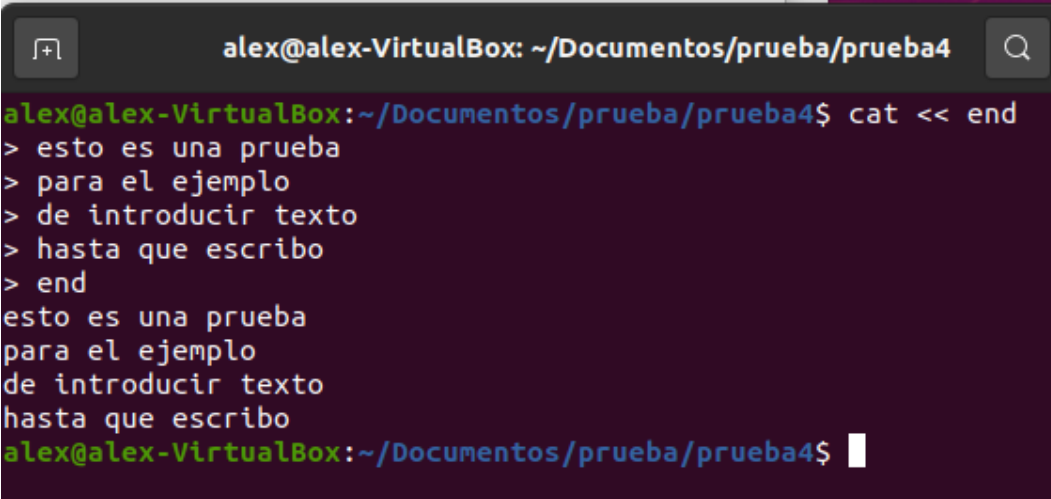
```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ sort docu2 >> docu
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ sort docu > docu3
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$
```

# Gestión de Archivos. Entradas y salidas estándar. Redirecciones de la entrada estándar.

Podemos redireccionar un fichero como entrada de una orden en lugar de la entrada estándar con : **orden < fichero**

Una redirección particular consiste en introducir texto hasta introducir un delimitador:

**< < delimitador**



```
alex@alex-VirtualBox: ~/Documentos/prueba/prueba4
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ cat << end
> esto es una prueba
> para el ejemplo
> de introducir texto
> hasta que escribo
> end
esto es una prueba
para el ejemplo
de introducir texto
hasta que escribo
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$
```

# Gestión de Archivos. Entradas y salidas estándar. Redirecciones de la salida de error estándar

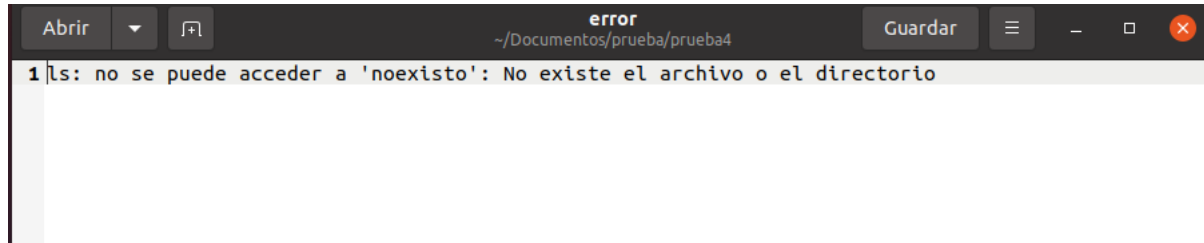
Durante la ejecución de un comando, tanto de salida estándar como de salida de error estándar pueden tener o no actividad. Al ser flujos diferentes, la salida de error estándar también se puede redireccionar empleando el operador 2>

## orden 2> fichero

Y para añadir contenido al fichero: 2 > >

Ejemplo: probamos a un ls sobre un archivo que no exista y redireccionado al fichero error:

```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ ls noexist 2> error
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$
```



# Gestión de Archivos. Entradas y salidas estándar. Otras opciones

el operador tubería o pipe “|” para concatenar salidas estándar con entradas estándar entre órdenes **ordenA | ordenB**

# Gestión de Archivos. Procesamiento de textos

Existen dos comandos ampliamente utilizados para extraer información sobre textos generados por otros comandos, ficheros o cadenas de caracteres en general. Son **cut** y **grep**

El comando **cut** se emplea para obtener información a partir de la división de un fichero o cadena de caracteres en columnas. Estas columnas se pueden establecer por caracteres o por campos delimitados por un delimitador de campo.

```
1 Pedro:Martínez:19
2 Adela:Suárez:34
3 Jose:Renau:45
4 JuanJo:Monfort:47
5 Antonio:Font:13
6 María:Adell:34
7 Julia:Ferrer:24
```

**cut -c <lista\_de\_caracteres> | -f <lista\_de\_columnas> [<-d delimitador>] fichero\_texto**

- -c <lista\_de\_caracteres>: corta por caracteres especificados
- -f <lista\_de\_columnas> [<-d delimitador>]: corta por campos establecidos por lista\_columnas. Por defecto, los delimitadores son espacio, tabuladores, espacio fin de línea, o el que especifique -d . Incompatible con -c

```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ cut -f3 -d: docu2
19
34
45
47
13
34
245
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$
```



# Gestión de Archivos. Procesamiento de textos

El comando grep localiza un patrón en uno o varios ficheros, mostrando las líneas donde se encuentra.

**grep [-nvlicw] patrón fichero\_texto [fichero\_texto ...]**

Las opciones más empleadas son:

- l: solo muestra los ficheros que contienen el patrón especificado.
- i: elimina la distinción entre mayúsculas y minúsculas
- c: muestra el número de líneas totales que cumplen con el patrón para cada fichero
- w: localiza el patrón como palabra y no como parte de una cadena de texto
- n: imprime el número de línea del patrón localizado
- v: busca líneas que no contengan el patrón especificado

Este comando permite emplear expresiones regulares básicas (patrones que definen un conjunto de cadenas de texto).

# Gestión de Archivos. Procesamiento de textos

Expresiones regulares más utilizadas:

Símbolo	Significado	Ejemplos
.	Cualquier carácter excepto el carácter fin de línea	Cas.
*	Cero o más repeticiones del carácter que le precede	C*
[Lista]	Coincide con uno de los caracteres en la lista	[aBgh]
	Se puede negar la coincidencia de un patrón	[^aBgh]
	Se pueden indicar rangos de caracteres, si se incluyen guiones y estos caracteres se especifican de mayor a menor	[0-9] [^0-9]
^	Comienzo de línea	^C
\$	Fin de línea	a\$

Los símbolos que tienen un significado especial se pueden emplear si se anteceden con el carácter “\” en los patrones. Ej: “\\*” o “\^”

Ejemplo grep. Líneas que contienen 34 mostrando el número de línea

docu2

```
1 Pedro:Martínez:19
2 Adela:Suárez:34
3 Jose:Renau:45
4 JuanJo:Monfort:47
5 Antonio:Font:13
6 María:Adell:34
7 Julia:Ferrer:24
```

```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ grep -n "34" docu2
2:Adela:Suárez:34
6:María:Adell:34
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$
```

contienen j seguido de cualquier letra mostrando el número de línea

```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ grep -n "J." docu2
3:Jose:Renau:45
4:JuanJo:Monfort:47
7:Julia:Ferrer:24
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$
```

# Gestión de almacenamiento

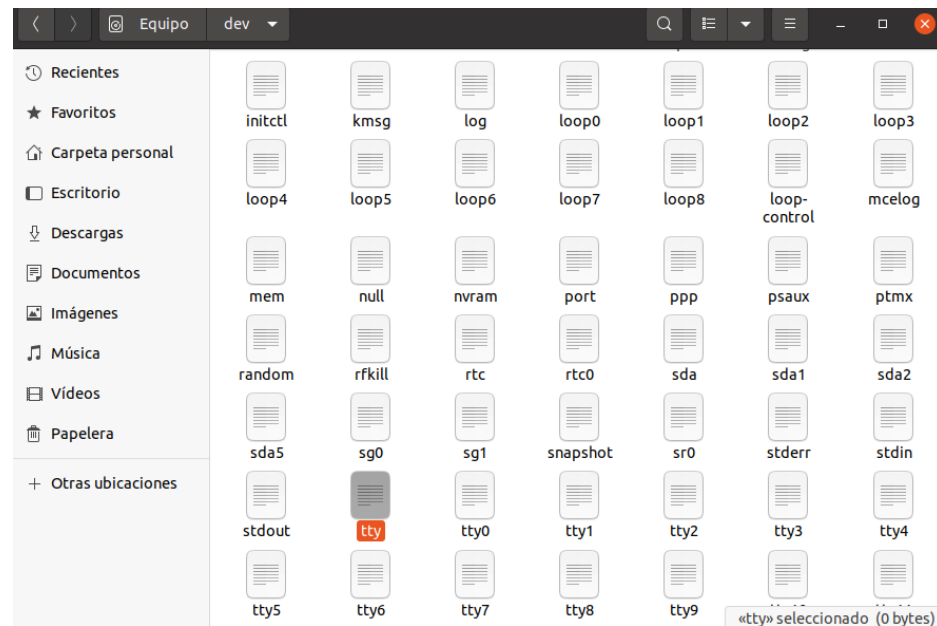
Dispositivos de almacenamiento se administran desde /dev (como el resto de dispositivos físicos).

Nos interesan estos archivos:

- `/dev/hd*`: interfaz para unidades de disco duro IDE
- `/dev/sd*`: interfaz para discos SCSI, SATA y unidades USB
- `/dev/tty*`: consolas o terminales físicos (CTRL+ALT+F1..F6 para cambiar entre ellas +F7 volver a entorno gráfico)
- `/dev/ttyS*`: puertos serie.
- `/dev/sr*` y `/dev/scd*`: interfaz para unidades CD o DVD

`/dev/<id_dispositivo><letra_orden><numero_partición>`

`/dev/sda1`



# Gestión de almacenamiento. Montaje y desmontaje

## **mount [-avwrt] [tipo][dispositivo][punto\_de\_montaje]**

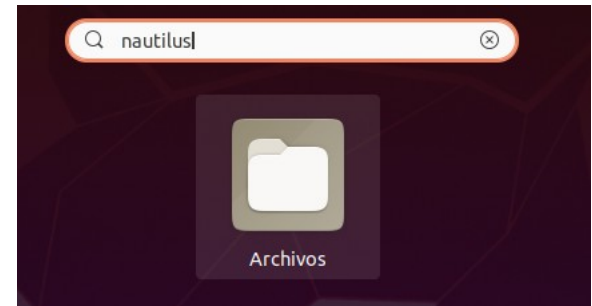
- -a: monta los sistemas de archivos presentes en /etc/fstab (si no está el parámetro noauto)
- -v: muestra información del proceso de montaje.
- -w: monta el sistema de archivos con permisos de solo escritura
- -r: monta el sistema de archivos con permisos de solo lectura
- -t<tipo>: indica el tipo de sistema de archivos para montar. Este puede ser entre otros ext, ext2, ext3, ext4, ntfs, nfs, smbfs, msdos ...

**Por defecto Ubuntu monta automáticamente un sistema de archivos (al detectar Pendrive, CD, etc) y gestionados por Nautilus**

Para desmontar (dejar de usar o extraer un dispositivo con sistema de archivos):

**umount <dispositivo>**

**umount <punto\_de\_montaje>**





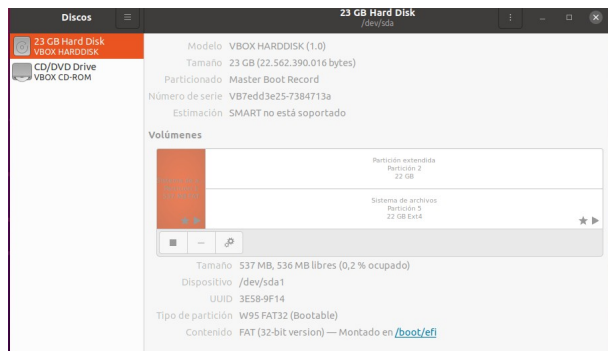
# Gestión de almacenamiento. Particionar

Herramientas: **fdisk**, **gdisk**, **parted** (estas tres con interfaz de texto) y **GParted**. Menos gdisk todas trabajan con MBR y GPT (gdisk solo GPT)

Ubuntu con escritorio GNOME incorpora **GNOME-disks**

Esta permite:

- × Formatear
- × Editar la partición
- × Redimensionarla
- × Comprobar y reparar el sistema de archivos
- × Crear y restaurar una imagen
- × Probar el rendimiento de la partición



**GParted. No viene por defecto.**

Esta permite:

- × Crear nuevas particiones en espacio no particionado
- × Eliminar o crear nuevas particiones
- × Redimensionar o mover particiones
- × Copiar y pegar particiones en otros discos
- × Formatear
- × Asignar un nombre
- × Y más...



# Gestión de almacenamiento. Formatear

## **mkfs [-t tipo\_sistema\_archivos] [opciones] dispositivo**

- t tipo\_sistema\_archivos: ext2, ext3, vfat (msdos), ntfs...
- Opciones: se pueden especificar muchas opciones como etiquetas, formato rápido, tamaño de clúster...

Para formatear hay que desmontar la partición primero.

ejemplo

```
mkfs -t ntfs /dev/sdc1
```

# Gestión de almacenamiento. Chequeo

**fsck [-A][-V][-t tipo\_sistema\_archivos] [-a][-r] [sistema\_de\_archivos]**

- A: Chequea todos los sistemas de archivos establecidos en fstab. (arranque)
- V (verbose): detalla las acciones realizadas por fsck
- T tipo\_sistema\_archivos: ext2, ext3, ext4, fat, ntfs
- a: repara sin pedir confirmación
- r: pide confirmación antes de reparar los daños

Para chequear sistemas ext podemos usar también **e2fsck**



# Administración de RAID

## 1. Creación de RAID.

**Mdadm - -create /dev/mdX - -level=Y - -raid-devices=Z dispositivos**

Donde:

- **create /dev/mdX** indica la creación del multidispositivo, siendo X un número.
- **level=Y** es el nivel de RAID para aplicar, pudiendo ser Y:
  - linear para RAID lineal.
  - raid0, 0 o stripe para RAID0
  - mirror, raid1 o 1 para RAID1
  - raid4 o 4 para RAID4
  - raid5 o 5 para RAID5
  - raid6 o 6 para RAID6
  - raid10 o 10 para RAID10
- **raid-devices=Z dispositivos**, donde Z indica el número de dispositivos asociados al RAID y cada uno de ellos separados por espacios (/dev/sdX /dev/sdY ...)

# Administración de RAID

2. Establecer un disco como defectuoso de un RAID:

**mdadm /dev/mdX - -fail /dev/sdY**

3. Eliminar un disco de un RAID:

**mdadm /dev/mdX - -remove /dev/sdY**

4. Añadir un disco a un RAID:

**mdadm /dev/mdX - -add /dev/sdY**

5. Comprobar el estado de todos los multidispositivos:

**cat /proc/mdstat**

6. Obtener información de todos los multidispositivos:

**mdadm - - detail - - scan**

# Administración de RAID

7. Obtener información de un multidispositivo:

**mdadm --detail /dev/mdX** y **mdadm --detail /dev/mdX --scan**

8. Examinar el estado de un dispositivo asociado a un RAID:

**mdadm --examine /dev/mdX**

9. Detener un RAID:

**mdadm --stop /dev/mdX**

10. Eliminar el superbloque de un dispositivo sobrescribiendo ceros:

**Mdadm --zero-superblock /dev/mdY**

# Administración de RAID. Eliminar

- 1.Desmontar el dispositivo si está en uso.
- 2.Detener el multidispositivo RAID:
  1. `sudo mdadm -- stop /dev/md0`
- 3.Borrar el superbloque de cada dispositivo que constituía el RAID:
  - 1.`sudo mdadm -- zero-superblock /dev/sde1`
  - 2.`sudo mdadm -- zero-superblock /dev/sdd1`
  - 3.`sudo mdadm -- zero-superblock /dev/sdc1`
- 4.En caso de que estuviera asociado al arranque del sistema, actualizar `/etc/fstab` eliminando la línea asociada y actualizar `initramfs`

# Búsqueda de información

**find [ruta] [criterio] [acción]**

Los criterios de búsqueda pueden ser:

- A) Por nombre de fichero
- B) Nivel de profundidad en subdirectorios
- C) Tiempos de acceso, modificación y cambio
- D) Comparación de ficheros
- E) Tamaños
- F) Tipo de fichero
- G) Permisos (**lo veremos tras ver permisos**)

Realiza búsqueda sobre la ruta dada.  
Si no se la damos sobre el directorio actual  
Resultado y errores por salida estándar  
Podemos aplicar criterios  
Realizar acciones sobre los resultados

# Búsqueda de información

**Por nombre de fichero** tenemos las siguientes opciones:

- ✓ name patrón: permite buscar ficheros con patrones.
- ✓ iname patrón: lo mismo pero sin distinguir mayúsculas y minúsculas

```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ ls
docu docu2 docu3 error
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ find -name 'do*'
./docu2
./docu
./docu3
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$
```

**Por nivel de profundidad de subdirectorios** tenemos las siguientes opciones:

- ✓ maxdepth n: ¿Hasta qué subdirectorios buscamos?
- ✓ mindepth n: desde qué nivel empezamos a buscar?

```
alex@alex-VirtualBox:~/Documentos/prueba/prueba4$ cd ..
alex@alex-VirtualBox:~/Documentos/prueba$ find -name 'do*'
./prueba4/docu2
./prueba4/docu
./prueba4/docu3
./prueba3/docu
alex@alex-VirtualBox:~/Documentos/prueba$ cd ..
alex@alex-VirtualBox:~/Documentos$ find -name 'do*'
./prueba/prueba4/docu2
./prueba/prueba4/docu
./prueba/prueba4/docu3
./prueba/prueba3/docu
alex@alex-VirtualBox:~/Documentos$ find -maxdepth 1 -name 'do*'
alex@alex-VirtualBox:~/Documentos$ find -maxdepth 2 -name 'do*'
alex@alex-VirtualBox:~/Documentos$ find -maxdepth 3 -name 'do*'
./prueba/prueba4/docu2
./prueba/prueba4/docu
./prueba/prueba4/docu3
./prueba/prueba3/docu
alex@alex-VirtualBox:~/Documentos$
```

# Búsqueda de información

## Por **Tiempos de acceso, modificación y cambio:**

- ✓ (c) última modificación del i-nodo
- ✓ (m) última modificación de su contenido
- ✓ (a) último acceso a su contenido
- ✓ Se puede especificar el tiempo en minutos o días
- ✓ Tenemos pues estas opciones: cmin, mmin, amin, ctime, mtime, atime el valor puede ser +n (mayor que), -n (menor que) o n (igual a)

```
alex@alex-VirtualBox:~/Documentos/prueba$ find -amin -5  
./texto1  
./texto1.txt  
alex@alex-VirtualBox:~/Documentos/prueba$
```

## Por **comparación de ficheros:**

- ✓ newer fichero: ficheros que se modificaron más recientemente que fichero
- ✓ aneuer fichero: accedidos más recientemente que este fichero
- ✓ cnewer fichero: el estado del i-nodo se modificó más recientemente que el fichero modificado

# Búsqueda de información

Por **Tamaños con -size[+-]n[bckMG]**:

- ✓ +n mayor que n, -n menor que n y n igual (n numérico)
- ✓ b bloques de 512 bytes
- ✓ c bytes
- ✓ k kilobytes
- ✓ M megabytes
- ✓ G gigabytes

Por **tipo fichero** tenemos las opción -type con las siguientes opciones:

- l enlace simbólico
- d directorio
- f fichero regular
- b dispositivo de tipo bloque
- c dispositivo de tipo carácter

```
alex@alex-VirtualBox:~/Documentos/prueba$ find -size +50c
./prueba2
./prueba1
./texto12
./prueba4
./prueba4/error
./prueba4/docu2
./prueba4/docu
./prueba4/docu3
./prueba3
./prueba3/prueba1.odt
```