

# Sistemas Informáticos

---

UD 0. Ubuntu 20.04 LTS. Instalación y  
comandos básicos



# ÍNDICE

1.	Instalación de Ubuntu 20.04 LTS.....	3
1.1.	Motivos por los que instalar esta distribucion .....	3
1.2.	Requisitos de hardware de Ubuntu 20.04 Desktop.....	3
1.3.	Datos de la máquina virtual a crear.....	3
1.4.	Proceso de instalacion de Ubuntu Desktop.....	3
2.	Sistema de archivos en Linux .....	4
2.1.	Características generales de los sistemas de ficheros de Linux .....	4
2.2.	Directorios importantes de Linux .....	5
2.3.	Tipos de archivos .....	6
2.4.	Ficheros y directorios ocultos.....	7
2.5.	Operaciones con ficheros y directorios en modo gráfico.....	7
3.	Interfaz textual: shell .....	9
4.	Aplicaciones .....	11
4.1.	Agregar o quitar programas .....	11

# 1. INSTALACIÓN DE UBUNTU 20.04 LTS

---

## 1.1. MOTIVOS POR LOS QUE INSTALAR ESTA DISTRIBUCION

Actualmente, Ubuntu, como ya se comentó anteriormente, es una de las distribuciones más descargadas y además, es una de las que más derivadas o “distribuciones hijas” está desarrollando.

Además, el motivo de utilizar esta versión en concreto, es decir, la **20.04 LTS** es porque la 20.04 es la **última versión LTS**. Las versiones LTS (Long Term Support), se liberan cada dos años y reciben **soporte durante cinco años** en los sistemas de servidor, mientras que en los de escritorio son 3.

Página de descarga de las ISOs de Ubuntu: <http://releases.ubuntu.com/20.04/>

## 1.2. REQUISITOS DE HARDWARE DE UBUNTU 20.04 DESKTOP

Según <https://help.ubuntu.com/community/Installation/SystemRequirements/> los **requisitos mínimos “recomendados” para la edición “desktop”**, teniendo en cuenta los efectos de escritorio, son:

- Procesador 2 GHz.
- Memoria RAM de 4 GB.
- Disco Duro de 25 GB (swap incluida).
- Tarjeta gráfica y monitor con resolución de 1024x768. Además de lector de CD, puerto USB o tarjeta de red.

## 1.3. DATOS DE LA MÁQUINA VIRTUAL A CREAR

Datos de la máquina virtual a crear:

- **RAM:** 1,5-2GB
- **Disco duro:** 20GB

## 1.4. PROCESO DE INSTALACION DE UBUNTU DESKTOP

Puedes ver el proceso de creación de una MV e instalación de Ubuntu 20.04 en: <https://youtu.be/0JI7hRH0HTs>

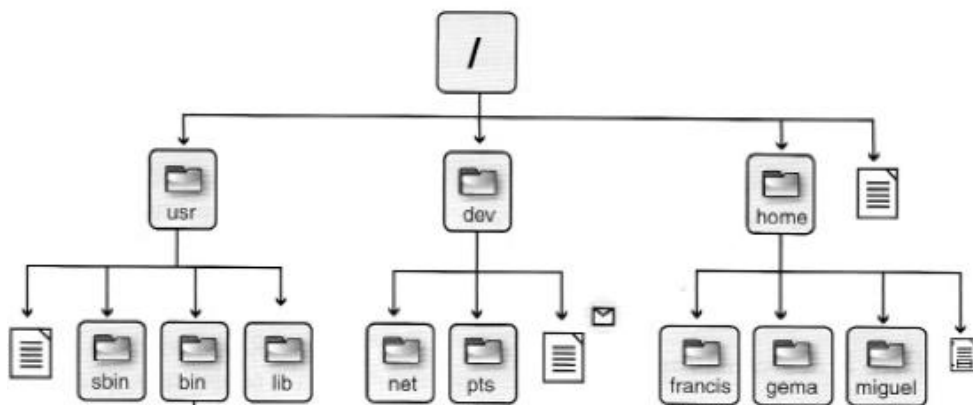
## 2. SISTEMA DE ARCHIVOS EN LINUX

### 2.1. CARACTERÍSTICAS GENERALES DE LOS SISTEMAS DE FICHEROS DE LINUX

Un **sistema de ficheros** es el modo en que el sistema operativo organiza los ficheros en el disco duro, gestionándolo de manera que los datos estén de forma estructurada y sin errores.

#### Directorio raíz “/” :

El sistema de ficheros de Linux tiene una **estructura jerárquica**, donde el **directorio principal (directorio raíz)** es el directorio **/**, del que cuelga toda la estructura del sistema. Por tanto, en **Linux** existe **un único árbol de directorios** y no uno por partición o disco (C:, D:...) como sucede en Windows. Es decir, en Linux todos los archivos y directorios “cuelgan” este directorio raíz **/**. Por tanto, cuando se conecta por ejemplo, un pendrive, este estará colgando del directorio raíz **/** o de uno de sus directorios, no tendrá una unidad independiente o letra como pasaba en Windows.



#### Archivos de usuarios (/home):

- El directorio **home** es el directorio que el sistema crea para cada usuario que ha sido dado de alta. En él cada usuario puede guardar sus archivos y directorios propios.
- Los **directorios personales de los usuarios** se encuentran bajo el directorio **/home**. De este modo el directorio home del usuario mar sería **/home/mar**
- En cambio, el **directorio personal del usuario root** o superusuario, es decir, el directorio personal del administrador del sistema, no se encuentra en **/home**, sino que se encuentra en **/root**

#### Nombre de los ficheros y carpetas:

Respecto al nombre de ficheros y carpetas, debe tener entre 1 y 255 caracteres y se puede usar **cualquier carácter, excepto el carácter /** (que se usa para el directorio raíz). De todas maneras no es recomendable usar los siguientes caracteres, por tener significado especial en Linux:

`\ ^ ~ ! # ? & ( ) ' " ` ; . $ = < > @ { } * + -`

#### Distinción de mayúsculas y minúsculas:

Linux **distingue entre mayúsculas y minúsculas** en todo, es decir, en nombres de carpetas, ficheros, comandos, programas o cualquier elemento. Eso significa que podremos crear en el mismo directorio el fichero “ejercicios.doc” y el fichero “Ejercicios.doc”, porque serán ficheros diferentes.

## Rutas absolutas y relativas:

Existen dos maneras de indicar la ruta a seguir a través del árbol de directorios de Linux hasta alcanzar un fichero o directorio:

- **Ruta absoluta:** Si indicamos la ruta desde el directorio raíz del árbol, construimos una ruta absoluta. Comenzamos indicando el directorio raíz "/" y, a continuación, añadimos los sucesivos directorios hasta alcanzar el destino separándolos mediante la barra /.  
Por ejemplo: /home/mar/doc
- **Ruta relativa:** Si indicamos la ruta a seguir desde el directorio actual (que se representa por un punto .), construimos una ruta relativa.  
Por ejemplo: si estamos en el directorio home de mar, la ruta relativa a al directorio doc sería: ./doc

## Directorio actual (.), directorio padre (..) y directorio personal (~):

Existen una serie de trucos que facilitan el acceso a ficheros y directorios:

- .. (dos puntos seguidos) → representa el **directorio padre** del directorio actual
- . (un punto) → representa el **directorio actual**
- ~ → representa nuestro **directorio home** (se escribe pulsando Alt Gr + 4)

Puedes usar estos "trucos" a la hora de moverte por el árbol de directorios de Linux.

## 2.2. DIRECTORIOS IMPORTANTES DE LINUX

- **/bin y /usr/bin:** contienen la mayoría de ficheros ejecutables y comandos comunes de Linux que pueden usar todos los usuarios.
- **/boot:** contiene los ficheros necesarios para el arranque del sistema (ficheros del gestor de arranque LILO o GRUB)
- **/dev:** almacena "ficheros especiales" que representan los dispositivos de E/S. En realidad no son "ficheros" propiamente dichos, sino que es la forma en que Linux implementa los controladores de dispositivos.
- **/etc:** contiene los ficheros de configuración del sistema. Este directorio es MUY importante. Evidentemente, solo un usuario con privilegios de administrador puede modificarlos.
- **/home:** donde se almacenan el "directorio home" o, dicho de otro modo, la "carpeta personal", de cada uno de los usuarios. Es como el "documents and settings" de Windows.
- **/lib y /usr/lib:** contienen librerías compartidas del sistema (librerías necesarias para que se ejecuten los programas que tenemos en /bin y /sbin)
- **/lost+found:** encontramos información y los procesos que se ejecutaban antes de una caída del sistema
- **/media:** cuando se monta automáticamente un CD\_ROM, o pendrive o disco duro externo, se crea aquí su subdirectorio. Por ejemplo: /media/usbdisk
- **/mnt:** es el directorio por defecto para realizar el montaje de otros dispositivos de almacenamiento
- **/opt:** directorio opcional donde se puede instalar aplicaciones (las que no se instalan automáticamente), además de en /usr. En algunas distribuciones Linux no existe este directorio.
- **/proc:** contiene archivos que reciben o envían información al kernel (núcleo). Por ejemplo: el fichero partitions contiene información de las particiones del disco y filesystem contiene los sistemas de archivos del sistema.
- **/root:** es el directorio home (personal) del usuario root

- **/sbin y /usr/sbin** : contienen comandos y ejecutables de tareas de administración que en su mayoría solo puede usar, evidentemente, el usuario root
- **/tmp**: almacena ficheros temporales
- **/usr**: almacena las aplicaciones de uso general para todos los usuarios, por lo que si hay muchos paquetes instalados, puede ocupar mucho espacio.
- **/var**: contiene información variable, como colas de impresión, de envío, de recepción de correos, archivos de registro y de eventos del sistema...

### 2.3. TIPOS DE ARCHIVOS

En Linux existen estos tipos de archivos:

- **Archivos regulares.** Contienen la información con la que trabaja cada usuario. Son los archivos ordinarios de datos.
- **Directorios.** Son archivos especiales que contienen referencias a otros archivos o directorios.

- **Enlaces físicos o duros (hard links).** No es específicamente una clase de archivo sino un segundo nombre que se le da a un archivo.

Cada archivo en el sistema de archivos de Linux tiene su propio *i-nodo*. Un i-nodo guarda los atributos del archivo y su posición en el disco. Si necesitamos referirnos a un archivo usando dos nombres de archivo distintos, podemos crear un enlace duro. El enlace duro tendrá el mismo i-nodo que el archivo original y guarda el mismo aspecto y comportamiento que el original. Con cada enlace duro creado, se incrementa la *cuenta de enlaces o referencia*. El conteo del número de enlaces duros que tiene un fichero se realiza, como ya hemos visto, en el inodo correspondiente a los datos del fichero. Cuando se borra un enlace duro, se decrementa esta cuenta de enlaces. Hasta que la cuenta de enlaces no alcanza cero, el archivo permanece en el disco.

- **Enlaces simbólicos o blandos (soft links).** También se utilizan para asignar un segundo nombre a un archivo. La diferencia con los enlaces duros es que los simbólicos solamente hacen referencia al nombre del archivo original, mientras que los duros hacen referencia al inodo en el que están situados los datos del archivo original. Sería como los “accesos directos” de Windows.

De esta manera, si tenemos un enlace simbólico y borramos el archivo original perderemos los datos, mientras que si tenemos un enlace duro los datos no se borrarán hasta que se hayan borrado todos y cada uno de los enlaces duros que existen hacia esos datos en el sistema de ficheros. Además, se mostraría un error al intentar acceder al archivo borrado a través del enlace simbólico.

- **Archivos especiales.** Suelen representar dispositivos físicos, como unidades de almacenamiento, impresoras, terminales, etc. En Linux, todo dispositivo físico que se conecte al ordenador está asociado a un archivo. Pueden ser de dos tipos: de bloques o de caracteres. Un disco duro es un dispositivo de bloque y una impresora es un dispositivo de carácter.
- Otros tipos de archivo son **las tuberías (pipes) o los sockets**.

Tipos de archivos (ls -l)	
-	Archivo ordinario
d	Directorio
l	Enlace simbólico
b / c	Archivo especial (c= de caracteres y b=de bloques)
p	Tubería
s	Socket

## 2.4. FICHEROS Y DIRECTORIOS OCULTOS

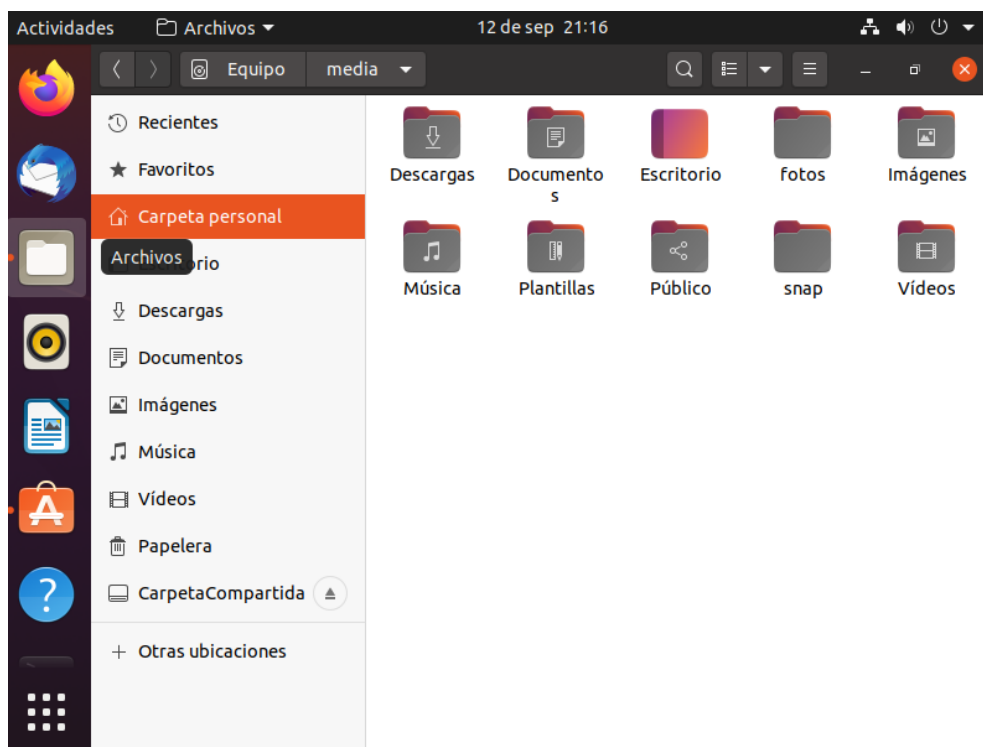
Los ficheros ``ocultos'' en Linux son **aquellos cuyo nombre comienza con el carácter punto (.)**

Al igual que en los sistemas de ficheros de Microsoft, en todos los directorios existen dos ficheros ocultos especiales: `.` (directorio actual) y `..` (directorio padre).

## 2.5. OPERACIONES CON FICHEROS Y DIRECTORIOS EN MODO GRÁFICO

Sobre un fichero o directorio podemos realizar distintas operaciones, como, por ejemplo: crearlos, entrar o salir de estos, leer su contenido, borrarlos...

Para acceder a nuestros archivos y poder trabajar con ellos, haremos clic en el lanzador “Archivos”.



A continuación, se describe cómo realizar algunas de estas habituales operaciones:

#### Cómo **seleccionar más de 1 archivo/carpeta**:

- Si están juntos:
  - Pinchar con el ratón en el primero y luego pinchar en el último manteniendo pulsada la tecla SHIFT. Así seleccionamos todos los que estén en medio también.
  - Pinchando con el ratón y dibujando un cuadrado que los envuelva.
- Si están separados:
  - Pinchar con el ratón cada archivo, pulsando además la tecla Ctrl del teclado a la vez (con la otra mano)

#### Cómo **copiar, cortar, pegar** archivos/carpetas:

- Teclado:
  - COPIAR: Ctrl + C
  - CORTAR: Ctrl + X
  - PEGAR: Ctrl + V
- Botón derecho del ratón → cortar | copiar | pegar
- Menú Editar → cortar | copiar | pegar

#### Cómo **seleccionar todo**:

- Ctrl + A
- Menú Editar → Seleccionar todo

#### Cómo **crear carpetas**:

- Botón derecho del Ratón → crear una carpeta
- Menú Archivo → Crear una carpeta

#### Cómo **cambiar el nombre a un archivo/carpeta**:

- Botón derecho del Ratón > Renombrar
- Menú Editar → renombrar
- Tecla F2

#### Cómo **compartir carpetas a través de la red**:

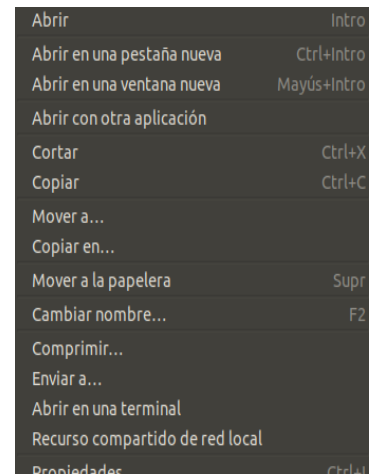
- Botón derecho → opciones de compartición
- Menú Editar → opciones de compartición

#### Cómo **eliminar un archivo/carpeta**:

- Botón derecho del ratón → mover a la papelera
- Menú Editar → mover a la papelera

Habrás observado que el manejo de archivos y carpetas diferente que Windows: fácilmente, desde el botón derecho del ratón puedes:

- Abrir, cortar y copiar
- Abrir con otra aplicación
- Crear un enlace
- Cambiar nombre
- Mover a la papelera
- Comprimir: se utiliza para hacer una copia comprimida de todo el contenido de la carpeta seleccionada
- Recurso compartido de red local: se usa para compartir la carpeta en la red
- Enviar a: se usa para enviar directamente por correo electrónico, o a un DVD
- Propiedades: muestra las propiedades del archivo o carpeta seleccionada.
  - Pestaña básica: vemos las características de un archivo, como el tipo, contenido, fecha de modificación, además nos permite modificar su nombre
  - Permisos: podemos modificar los privilegios o permisos que los usuarios tienen sobre esta carpeta o archivo
  - Compartir: se utiliza para hacer que la carpeta esté a disposición de otros usuarios de la red





### 3. INTERFAZ TEXTUAL: SHELL

---

En Linux la interfaz de línea de comandos se conoce como **Shell**. La Shell Bash es la predeterminada en la mayoría de los sistemas Linux, como Ubuntu. No obstante, aparte de la Shell Bash, existen otras como sh, csh, ksh...

En Ubuntu se accede a la Shell a través del denominado **Terminal**. El terminal consiste en una pantalla en la que podemos introducir comandos.

Podemos acceder a un Shell de varias formas:

- Menú Aplicaciones del Dock → Terminal
- Pulsando simultáneamente las teclas Ctrl+Alt+T

En general, todas las órdenes de UNIX/Linux son programas que están almacenados en el sistema de ficheros. Su sintaxis es similar a la del entorno comando de Windows y es la siguiente:

**orden [-opciones] [argumentos]**

Antes de empezar a trabajar con los comandos de la Shell, ten en cuenta los siguientes **consejos**:

- Para trabajar más rápida y cómodamente con el Shell, puedes utilizar los siguientes **atajos**:
  - o Puedes **completar** los comandos usando la tecla **tabulador**
  - o Puedes **reescribir** comandos ya utilizados usando las **teclas** ↑ y ↓
- Puedes **detener** la ejecución de un comando pulsando **Ctrl+c**
- Mediante el comando **sudo** (que deberás escribir antes del comando) puedes **ejecutar comandos** que requieran **privilegios de root**
  - o El fichero **/etc/sudoers** contiene qué **usuarios o grupos** pueden **usar el comando sudo**
- Mediante la orden **su**, puedes **pasar a ser otro usuario**. Por ejemplo:
  - o **sudo su** → te conviertes en el usuario root
  - o **sudo su mar** → te conviertes en el usuario mar

El **prompt** del Shell bash muestra dos **formas** distintas según el tipo de usuario que lo utilice:

- Si se trata de un usuario **estándar** → finaliza con el carácter **\$**
- Si se trata de la cuenta de superusuario o **root** del sistema → finaliza con el carácter **#**

#### 3.1.1. Comandos

A continuación, se indica un resumen de los comandos básicos de Linux que necesitarás para empezar a trabajar en los módulos de Programación y Bases de datos, entre otros:

COMANDO	USO	QUÉ HACE	EJEMPLO
<b>Comandos de manejo de ficheros y directorios</b>			
pwd	pwd	Muestra la ruta completa del directorio actual	pwd
ls	ls [directorio] ls -l ls -a ls -r ls -i	Ver el contenido de un directorio	ls SOM muestra todos los ficheros y directorios del directorio SOM ls -l lista los archivos con su información detallada ls -a lista todos los archivos, incluidos los archivos ocultos ls -r SOM lista el contenido del directorio SOM y de todos sus subdirectorios (de forma recursiva) ls -i muestra el nodo-i
cd	cd [directorio]	Cambiar de directorio	cd /home/mar/SOM cambia al directorio SOM usando una ruta absoluta cd ./SOM cambia al directorio SOM usando una ruta relativa cd .. cambia al directorio padre cd ~ cambia al directorio home del usuario
mkdir	mkdir directorio mkdir -p	Crear un directorio	mkdir SOM crea un directorio llamado SOM mkdir -p SOM/PRACTICAS crea todos los directorios que conforman la ruta que le damos, es decir, crea el directorio SOM y dentro de éste, el directorio PRACTICAS
rm	rm fichero(s) rm -r directorio(s)	Eliminar un archivo	rm ./SOM/s* borra todos los ficheros del directorio SOM que comienzan por el carácter s (borra solo los ficheros, no subdirectorios)
		Eliminar un directorio	rm -r ./SOM/SUBDIR1 borra el directorio SUBDIR1 completo (con sus ficheros y directorios)
cp	cp fichero(s) directorio cp -r fichero(s) directorio(s) directorio	Copiar un archivo	cp ejercicio1.doc SOM copia el archivo ejercicio1.doc al directorio SOM cp ej* ./SOM copia todos los ficheros del directorio actual que empiezan por "ej" al directorio SOM cp ej? ./SOM copia todos los ficheros de tres caracteres del directorio actual que empiezan por "ej" al directorio SOM
		Copiar un archivo o directorio	cp -r ./SOM/practica1 ./SOM/PRACTICAS copia la carpeta practica1 dentro de la carpeta PRACTICAS
mv	mv fichero(s) directorio mv fichero1 fichero2	Cambiar el nombre o mover un archivo o directorio	mv ej1.txt pr1.txt ./tema1 mueve los ficheros ej1.txt y pr1.txt al directorio tema1 mv ej1.txt ej2.txt cambia el nombre del fichero ej1.txt a ej2.txt

Puedes ver un vídeo con ejemplos de algunos de estos comandos básicos en: <https://youtu.be/8e-UBTlu6X8>

## 4. APLICACIONES

### 4.1. AGREGAR O QUITAR PROGRAMAS

El directorio **/usr** de Linux aloja el **software instalado** en el sistema. Concretamente:

- **/usr/bin** contiene ejecutables
- **/usr/sbin** contiene ejecutables en modo root
- **/usr/lib** contiene librerías
- **/usr/share** contiene documentación de los paquetes

Un **paquete** es un conjunto de ficheros que se utilizan para instalar una aplicación. Los paquetes que contienen código máquina y no código fuente, se les llama paquetes binarios. Las distribuciones GNU/Linux poseen un sistema de administración de paquetes que permite encontrar, descargar e instalar programas.

Existen tres **tipos principales de paquetes** o métodos de distribución:

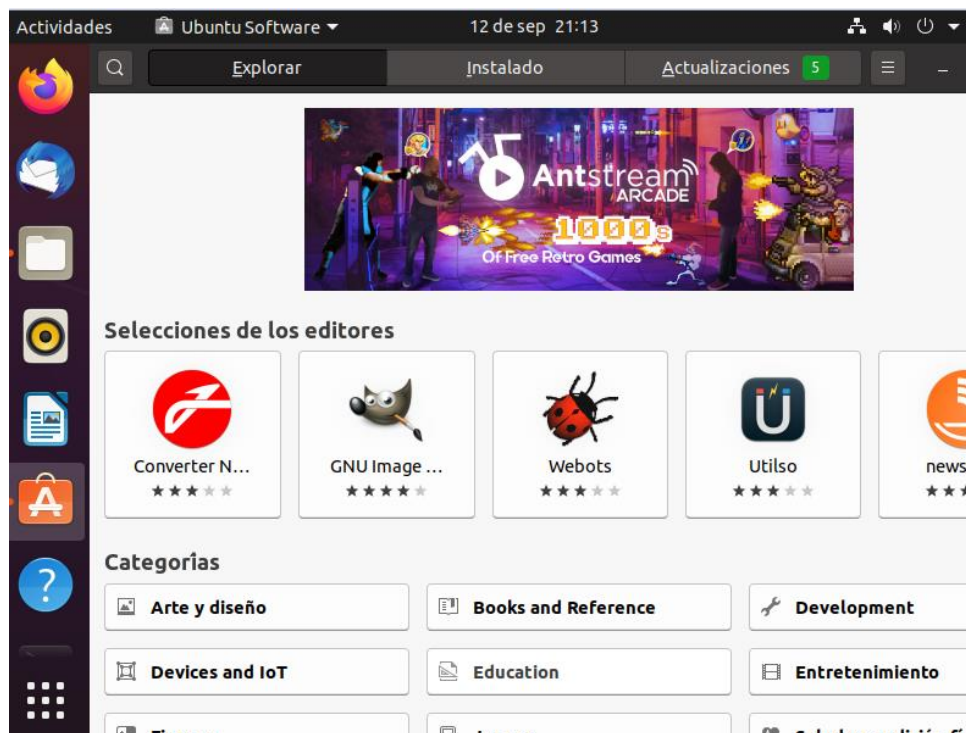
- **Deb** o paquetes Debian
- **Rpm** o paquetes RedHat
- **Tarball**, que es una colección de ficheros agrupados mediante la utilidad tar y comprimidos mediante gzip.

Los deb y rpm han sido adoptados por la mayoría de distribuciones.

Respecto a la gestión del software de Linux, ésta se puede realizar a través de la interfaz gráfica o mediante comandos.

#### 4.1.1. Modo gráfico: Ubuntu Software

Para gestionar paquetes software en Ubuntu mediante interfaz gráfica podemos usar **Ubuntu Software**:



En esta aplicación tienes todo ordenado por categorías, te da información sobre las aplicaciones, sugerencias, plugins o paquetes complementarios, comentarios de otros usuarios... Resulta muy sencillo de utilizar para usuarios con poca experiencia.

#### 4.1.2. Modo comando: apt, aptitude, dpkg...

Para gestionar paquetes software en Ubuntu mediante comando podemos usar:

- **Instalación de paquetes deb:**

- **Comando apt:**

- **apt-get update:** actualiza la lista de paquetes disponibles para instalar desde los repositorios del fichero sources.list
- **apt-cache search búsqueda:** si no conocemos el nombre exacto del paquete a instalar
- **apt-get install paquete:** instala paquetes por la red
- **apt-get install --reinstall paquete:** reinstala un paquete
- **apt-get remove paquete:** desinstala paquetes
- **apt-get upgrade:** instala las nuevas versiones de los paquetes ya instalados
- **apt-get purge paquete:** elimina el paquete y sus archivos de configuración

- **Aptitude** es una versión mejorada de apt y se usa con las mismas opciones que este.

- **Comando dpkg:**

- **dpkg -i paquete:** instala el paquete
- **dpkg -r paquete:** desinstala un paquete
- **dpkg --purge paquete:** desinstala un paquete y sus archivos de configuración

- **Instalación de paquetes tgz:**

Los paquetes tgz contienen el código fuente del programa con los archivos necesarios para compilar e instalar el programa. Estos paquetes están un tipo de fichero .tar.gz o tar.bz2.

En primer lugar, mediante los **comandos tar** (para des/empaquetar) y **gzip** o **bzip** (para des/comprimir) desempaquetamos y descomprimos. A continuación, el paquete se instala desde el directorio donde están los fuentes del programa con los siguientes comandos:

- **./configure**
- **make**
- **make install**

- **Instalación de paquetes rpm:**

Para los paquetes rpm en modo gráfico tenemos las utilidades yast y yum

En modo texto, utilizamos el comando rpm (rpm [-opciones] [paquetes])

- **rpm -i:** instala un paquete
- **rpm -u:** actualiza los paquetes ya instalados
- **rpm -e:** desinstala el paquete

Ojo! Estos métodos de instalación de aplicaciones **no puedes ejecutarlos de forma simultánea** (tampoco apt-get desde la terminal), ya que la primera aplicación que se ejecuta toma el control de ciertos archivos y los bloquea para que ninguna otra aplicación trastoque tu instalación. Esto último también se aplica al proceso de actualización automática.