

# ÍNDICE

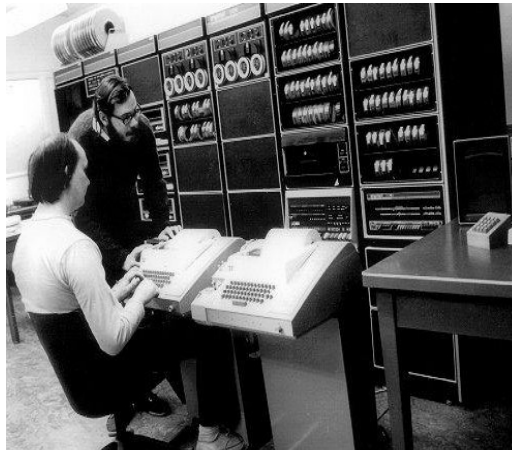
Historia de linux.....	2
1. Unix.....	2
2. Proyecto GNU – Ricard Stallman.....	3
3. Linus Torvalds – Nucleo de Linux.....	3
Distribuciones de Linux.....	5
Instalación de linux.....	7
1. Instalar la Guest Additions en lubuntu 18,04.....	7
2. Compartir carpetas en lubuntu 18,04.....	7
Instalación de Software en ubuntu.....	8
1. Desde el centro de software de Ubuntu.....	8
2. Gestor de paquetes Synaptic.....	9
3. Comandos apt-get (apt) y aptitude en una terminal o consola.....	9
4. Paquetes con extensión .deb.....	10
5. Instalar a partir del código fuente (Compilar).....	11
6. Actualización del sistema.....	11
Primeros comandos en linux.....	12
a) Comando pwd.....	12
b) Comando cd.....	12
c) Comando touch.....	13
d) Comando ls.....	13
e) Comando mkdir.....	13
f) Comando mv.....	14
g) Comando cp.....	14
h) Comando rm.....	14
i) Otros comandos.....	14
j) Iniciación a shell scripts.....	15

# Historia de linux

## 1. Unix

---

En 1970 , dos investigadores de los Laboratorios Telefónicos Bell (BTL) llamados Dennis Ritchie y Ken Thompson desarrollaron un sistema operativo Unix, este sistema operativo fué creado con lenguaje ensamblador.



A BTL le era imposible entrar en el negocio de los sistemas operativos por tanto no podían patentarlo. La presión de los investigadores en obtener una copia de Unix motivó a los ejecutivos de BTL a licenciar su uso como una herramienta de investigación. La licencia de Unix era muy barata para las universidades y bastante cara para la industria.

Muchas empresa y universidades empezaron a trabajar con Unix y se crearon muchas versiones de unix.

El profesor Andrew Tanenbaun creo un mini sistema operativo a partir de Unix llamado **Minix** para enseñarle a sus alumnos acerca del diseño de sistemas operativos. Sin embargo, debido al enfoque puramente educacional de Minix, Tanenbaum no permitía que este fuera modificado demasiado ya que esto complicaba el sistema y no permitía que sus estudiantes lo entendieran en un semestre.

## 2. Proyecto GNU - Ricard Stallman

---

En 1983 Richard Stallman entonces en el Instituto Tecnológico de Massachussets (MIT ) decidió dedicarse a la construcción de lo que denominó software libre e inició el **Proyecto GNU**, con el propósito de crear un sistema operativo similar y compatible con UNIX.



Dos años más tarde, 1985, creó la Fundación del Software Libre (FSF) y **desarrolló la GNU General Public License (GPL)**, para difundir el software libremente. De este modo el software de GNU fue desarrollado muy rápidamente, y por muchas personas.

A corto plazo, una multiplicidad de programas fueron desarrollados, de modo que a principios de los años 1990 había casi bastante software disponible para crear un sistema operativo completo. **Sin embargo, un núcleo todavía faltaba**

## 3. Linus Torvalds - Nucleo de Linux

---

En 1991 Linus Torvalds un estudiante finlandés de 21 años desarrolló un núcleo compatible con Unix y lo denominó Linux.

Todo comenzó como un proyecto fin de carrera: se trataba de programar un núcleo para sistema operativo inspirado en Minix, un pequeño Unix desarrollado por el profesor Andrew Tanenbaum.

Cuando linus termino su proyecto lanzó el siguiente mensaje a la red:

"Estoy haciendo un sistema operativo libre (es sólo un hobby, no será grande ni profesional como GNU) pero funciona en 386(486) AT clones, lo he estado cocinando desde abril y está quedando listo. Quisiera alguna retroalimentación de las cosas que a ustedes les gustan y no les gustan de MINIX, ..."



Este sencillo mensaje de Torvalds en un grupo de noticias motivó una avalancha de colaboradores y los frutos no se harían esperar. En enero de 1992 se publicó la versión 0.02, y poco tiempo después, en marzo de 1994 se liberó la versión 1.0.0, ya lista para sistemas en producción. A partir de esta versión al combinar Linux con el sistema no completo GNU resultó un sistema operativo libre completo cuyo nombre correcto es **GNU/Linux**.

**Linus Torvalds, llenó el hueco final que el sistema operativo GNU exigía..**

Aplicaciones usuario	Aplicaciones usuario	Aplicaciones usuario	Aplicaciones usuario	Aplicaciones usuario	Aplicaciones usuario	PROGRAMAS DE USUARIO
Shell (intérprete de comandos): Modo texto / GNOME & Unity Aplicaciones Base GNU/Linux						PROGRAMAS DEL SISTEMA
Planificación CPU RR por prioridades: De -19 a +19 siendo los menores más prioritarios	Gestión de memoria Paginación y Memoria virtual (swapping en partición propia)	Sistema de ficheros Estructura en árbol Bloques dos tamaños	Gestión de E/S Dispositivos bloque (Discos) Dispositivos caracteres (impresoras)			KERNEL
HARDWARE						

## Distribuciones de Linux

Linux es un sistema de libre distribución por lo que se puede encontrar todos los ficheros y programas necesarios para su funcionamiento en multitud de servidores conectados a Internet.

La tarea de reunir todos los ficheros y programas necesarios, así como instalarlos en tu sistema y configurarlo, puede ser una tarea bastante complicada. Por esto mismo, nacieron las llamadas distribuciones de Linux, empresas y organizaciones que se dedican a hacer el trabajo "sucio" para nuestro beneficio y comodidad.



Una distribución no es otra cosa, que una recopilación de programas y ficheros, organizados y preparados para su instalación.

Estas distribuciones se pueden obtener a través de Internet, o comprando los CDs de las mismas, los cuales contendrán todo lo necesario para instalar un sistema Linux bastante completo y en la mayoría de los casos un programa de instalación que nos ayudara en la tarea de una primera instalación. Casi todos los principales distribuidores de Linux, ofrecen la posibilidad de bajarse sus distribuciones.

Al ser un software con licencia GPL, puedes coger cualquier distribución y modificarla siempre que distribuyas la nueva con licencia GPL.

Las principales distribuciones son:

- Debian – De esta derivan distribuciones tan importantes como Ubuntu, Mint,
- Red Hat - Creada por la compañía RedHat ofreció soporte gratuito hasta la versión 9 momento en que decidió concentrar sus esfuerzos en el desarrollo de la versión corporativa RedHat Enterprise Linux y delegó la versión libre a Fedora Core, un proyecto abierto e independiente de Red Hat. CentOS también deriva de Red Hat.
- Open Suse - Es una de las principales distribuciones GNU/Linux existentes a nivel mundial, nacida en Alemania

[https://es.wikipedia.org/wiki/Distribuci%C3%B3n\\_Linux#/media/Archivo:Linux\\_Distribution\\_Timeline.svg](https://es.wikipedia.org/wiki/Distribuci%C3%B3n_Linux#/media/Archivo:Linux_Distribution_Timeline.svg)



## Instalación de linux

En la primera evaluación se vio en detalle la instalación de lubuntu.

### 1. Instalar la Guest Additions en lubuntu 18,04

---

Lo primero que tenemos que hacer es insertar el CD de las Guest Additions – En Dispositivos→Insertar cd de las Guest Additions.

Antes de ejecutar las Guest necesitamos instalar los siguientes programas:

```
sudo apt-get install gcc make perl
```

A Continuación en línea de comando se tiene que ejecutar el archivo VBoxLinuxAdditions.run que está en el cd.

```
sistemas@sistemas-VirtualBox:~$ cd /media/sistemas/VBox_GAs_6.1.14/
sistemas@sistemas-VirtualBox:/media/sistemas/VBox_GAs_6.1.14$ dir
AUTORUN.INF      VBoxDarwinAdditions.pkg
autorun.sh       VBoxDarwinAdditionsUninstall.tool
cert             VBoxLinuxAdditions.run
NT3x            VBoxSolarisAdditions.pkg
OS2             VBoxWindowsAdditions-amd64.exe
runasroot.sh     VBoxWindowsAdditions.exe
TRANS.TBL       VBoxWindowsAdditions-x86.exe
sistemas@sistemas-VirtualBox:/media/sistemas/VBox_GAs_6.1.14$ sudo ./VBoxLinuxAdditions.run
[sudo] contraseña para sistemas:
Verifying archive integrity... All good.
Uncompressing VirtualBox 6.1.14 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Removing installed version 6.1.14 of VirtualBox Guest Additions...
update-initramfs: Generating /boot/initrd.img-4.15.0-20-generic
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions: /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 4.15.0-20-generic.
update-initramfs: Generating /boot/initrd.img-4.15.0-20-generic
VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted
sistemas@sistemas-VirtualBox:/media/sistemas/VBox_GAs_6.1.14$
```

### 2. Compartir carpetas en lubuntu 18,04

---

Una vez instaladas las Guest Additions, podemos compartir carpetas con el sistema real. La forma de hacerlo es igual que en Windows pero antes de compartirlas debemos introducir al usuario con el que estamos trabajando en el grupo vboxsf con el siguiente comando:

```
sudo addgroup sisteamas vboxsf
```

## Instalación de Software en ubuntu

La instalación de programas en Linux es muy diferente a Windows, en Linux no hay que descargar aplicaciones, y después activarlo.

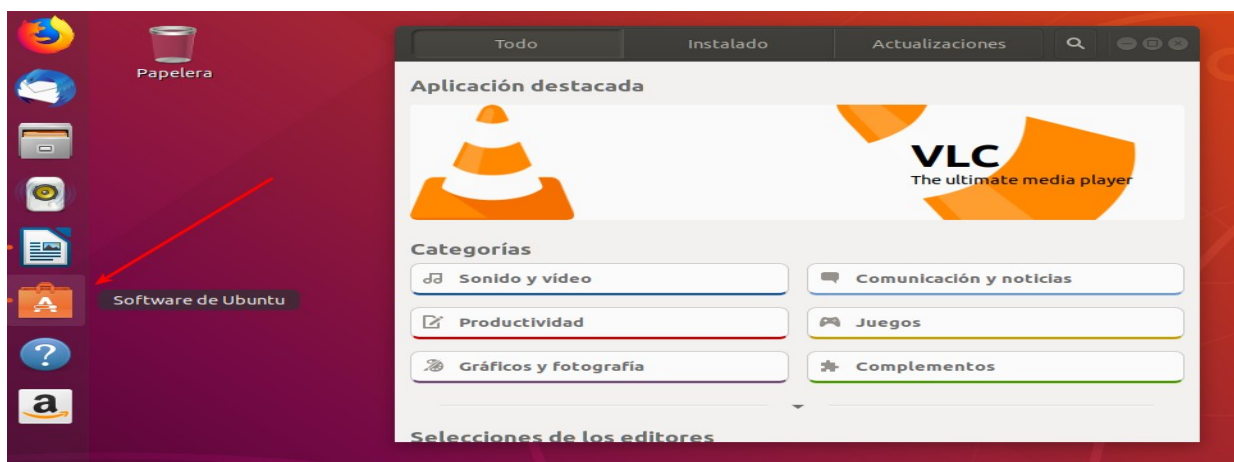
En Linux existen los repositorios que son como un depósito o un sitio centralizado donde se almacena y actualiza todo el software disponible. Sólo tenemos que seleccionar la aplicación y el sistema se encarga de descargar e instalar.

Existen varias maneras de instalar software en Ubuntu

- ◆ Desde el centro de software de ubuntu
- ◆ Gestor de paquetes Synaptic
- ◆ Comandos apt-get (apt) y aptitude en una terminal o consola
- ◆ Paquetes con extensión .deb
- ◆ Instalar a partir del código fuente (Compilar)

### 1. Desde el centro de software de Ubuntu

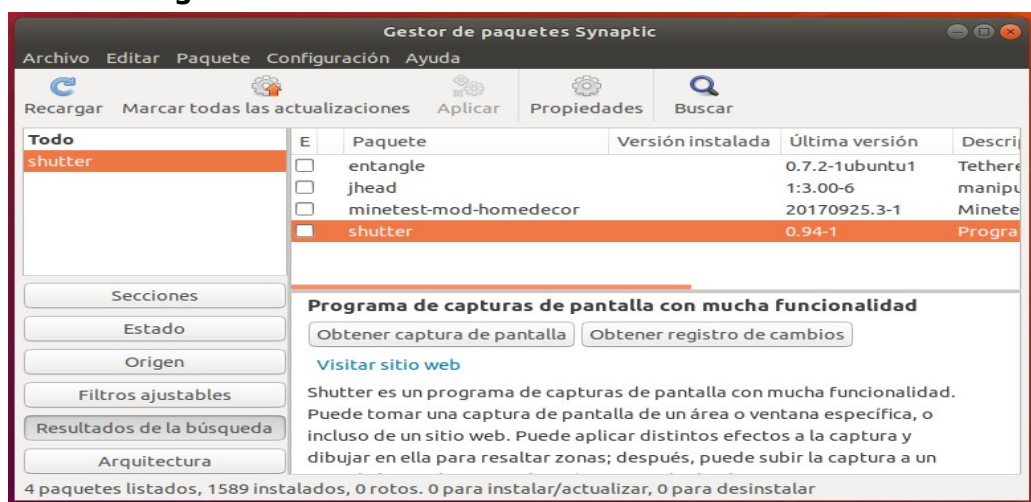
Es el más simple e intuitivo de todos y según van saliendo las nuevas versiones de Ubuntu, se le van implementando más funcionalidades. Es una auténtica tienda de software, con más de mil aplicaciones a tu disposición.





## 2. Gestor de paquetes Synaptic

Synaptic es un avanzado sistema para instalar o eliminar aplicaciones de tu sistema. El entorno es gráfico y muy potente. Con Synaptic tienes el control completo de los paquetes (aplicaciones) instalados en tu sistema, sus dependencias y desde aquí puedes seleccionar las diferentes versiones de los paquetes a instalar según las necesidades.



## 3. Comandos apt-get (apt) y aptitude en una terminal o consola

Existen dos comandos para instalar programas en modo texto: con apt-get (apt) o con aptitude.

Ambos programas son muy similares, pero nos vamos a centrar en el apt-get.

Como el comando apt instala software en nuestro equipo se tiene que ejecutar con sudo

\* Instalar paquetes:

```
sudo apt-get install nombre_paquete
```

\* Instalar varios paquetes a la vez dejando un espacio en blanco entre ellos ej.

```
sudo apt-get install gnome-do pidgin amsn
```

\* **Desinstalar paquetes:**

```
sudo apt-get remove nombre_paquete
```

\* **Desinstalar paquetes (incluyendo archivos de configuración):**

```
sudo apt-get remove --purge nombre_paquete
```

\* **Actualizar la lista de paquetes disponibles:**

```
sudo apt-get update
```

\* **Actualizar el sistema con las actualizaciones de paquetes disponibles:**

```
sudo apt-get upgrade
```

\* **Obtener una lista de opciones del comando:**

```
sudo apt-get help
```

#### **4. Paquetes con extensión .deb**

---

Otra forma de instalar aplicaciones en el sistema es por medio de los paquetes ya preparados para ser instalados y con extensión .deb.

Para instalar estos paquetes sólo tienes que hacer doble click sobre el fichero en el navegador Nautilus y automáticamente se lanzará el "Centro de Software" o la aplicación gdebi (si la tenemos instalada), que se ocupará de instalar el paquete y buscar las dependencias de otros paquetes que pudiera necesitar para su correcta instalación.

Si lo preferimos, también se pueden instalar mediante la línea de comandos, mediante el comando dpkg:

```
sudo dpkg -i nombre_del_paquete.deb
```

El mismo comando también se puede usar para desinstalar el paquete:

```
sudo dpkg -r nombre_del_paquete
```

Si un programa está en los repositorios que tenemos configurados, se puede descargar dicho paquete con el siguiente comando:

```
sudo apt-get install --download-only nombre-del-paquete
```

En la web <https://pkgs.org/> se encuentran recopilados numerosos paquetes para diferentes versiones de sistemas operativos.

## 5. Instalar a partir del código fuente (Compilar)

---

A veces se encuentran aplicaciones que no proporcionan paquetes de instalación, y hay que compilar a partir del código fuente.

## 6. Actualización del sistema

---

Para actualizar el sistema en ubuntu hay que ejecutar dos comando:

```
sudo apt-get update
```

Con este comando actualizaremos los repositorios de nuestro sistema. Los repositorios son almacenamientos en la red que contienen una gran cantidad de programas que se pueden instalar. Lo que se está haciendo cuanto hacemos `sudo apt-get update` es actualizar la lista de programas y sus versiones que hay en dichos repositorios.

La dirección de estos almacenamientos están configurados en el fichero:

`/etc/apt/sources.list`

```
deb http://security.ubuntu.com/ubuntu bionic-security main restricted  
# deb-src http://security.ubuntu.com/ubuntu bionic-security main restricted  
deb http://security.ubuntu.com/ubuntu bionic-security universe  
# deb-src http://security.ubuntu.com/ubuntu bionic-security universe  
deb http://security.ubuntu.com/ubuntu bionic-security multiverse
```

```
# deb-src http://security.ubuntu.com/ubuntu bionic-security multiverse
```

Una vez actualizados los repositorios, procederemos a actualizar los archivos disponibles con:

```
sudo apt-get upgrade
```

Este comando compara las versiones instaladas en el sistema con las nuevas versiones disponibles en los repositorios, para descargarlas y actualizarlas.

Se pueden instalar y configurar las actualizaciones con interface gráfica usando el programa Gestor de Actualizaciones.

## **Primeros comandos en linux**

Para poder utilizar bien linux tenemos que dominar los comandos en línea de comando, existen multitud de comandos con multitud de opciones que iremos viendo a lo largo del curso. Algunos de estos son:

### **a) Comando pwd**

Te indica el directorio donde estas. No se pasa ningún parámetro

```
pwd
```

### **b) Comando cd**

Nos permite cambiarnos de un directorio a otro. Se puede usar con ruta absoluta y con ruta relativa

```
cd /home/rubenb/Escritorio — RUTA ABSOLUTA- Indicamos todo el camino completo desde el /
```

```
cd ./Escritorio – RUTA RELATIVA – Indicamos el camino a partir de donde estamos
```

*. - Un punto indica el directorio donde estamos*

*.. -Dos puntos indica el directorio anterior de donde estamos*

### c) Comando touch

---

Permite crear ficheros vacios. Se pasa como argumento el camino y nombre donde se quiere crear el fichero. Se puede usar ruta relativa y ruta absoluta

```
#pwd
```

```
/home/rubenb
```

```
#touch ./Escritorio/fichero1.txt – Crea un fichero en el Escritorio con ruta relativa
```

```
#touch /home/rubenb/Escritorio/fichero1.txt – Lo mismo con ruta absoluta
```

Para crear ficheros de texto también se puede hacer con el operador > o >>

```
echo "contenido" > fichero.txt
```

### d) Comando ls

---

Permite visualizar el contenido de un directorio.

Algunas opciones:

```
#ls -l – muestra la salida en formato largo.
```

```
#ls -R – muestra recursivamente todos los directorios
```

```
#ls -a – Muestra archivos ocultos(archivos que empiezan por .)
```

```
#ls *.txt – Muestra todos los archivos que terminan con .txt
```

```
#ls f*.txt – Muestra todos los archivos que empiezan por f y terminan por .txt
```

### e) Comando mkdir

---

El comando mkdir se utiliza para crear directorios.

```
$ mkdir bin
```

```
$ mkdir -p docs/linuxdocs/howtos/pdf – se crean los directorios intermedios si es necesario
```

#### **f) Comando mv**

---

El comando mv mueve un fichero hacia otro, o hacia un directorio.

Este permite a su vez renombrar ficheros o directorios.

Algunas opciones:

```
$ mv mail.cf mail.cf.old – Cambia de nombre el fichero mail.cf por mail.cf.old
```

```
$ mv -i *.txt /tmp – Mueve todos los archivos que terminan en .txt al directorio /tmp, por cada fichero que mueve te pregunta
```

#### **g) Comando cp**

---

Permite copiar un fichero o directorio en otro, o varios ficheros en un directorio.

```
# cp /etc/passwd . – Copia el fichero passwd del directorio /etc al directorio . (donde estoy)
```

```
# cp /usr/bin/*sh tmp – Copia todos los ficheros que terminan en sh del directorio usr/bin al directorio tmp
```

```
#cp -R Escritorio ./Documentos – Copia el directorio Escritorio al directorio Documentos
```

#### **h) Comando rm**

---

Se utiliza para borrar ficheros

```
# rm prueba – Borra el fichero prueba
```

```
# rm -i bin/* – Borra todos los ficheros del directorio bin (la -i hace que pregunte si quieres borrar)
```

```
# rm -rf temp/ – Borra el directorio temp y todos sus subdirectorios ( -r borrado recursivo, y -f fuerza el borrado sin preguntar
```

#### **i) Otros comandos**

---

```
#clear - Borra la pantalla
```

```
#echo – Muestra en pantalla el mensaje indicado
```



*#who am i – indica el usuario con el que estas conectado.*

*#sudo – sirve para ejecutar como root un comando.*

*#cat – muestra el contenido de un archivo*

*#lsb\_release -a – Indica la versión de sistema operativos*

*#uname -a – Indica la versión de sistema operativo*

## j) Iniciación a shell scripts

---

Un shell script es un archivo con comandos de una shell(en nuestro caso bash) que se ejecutan secuencialmente, se utilizan para automatizar tareas.

Todo script Bash comienza con la cabecera:

```
#!/bin/bash
```

Por tanto si edito un archivo, lo guardo como miprimerscript.sh y dentro escribo:

```
#!/bin/bash
```

```
echo "Hola mundo"
```

Y para ejecutarlo debemos de darle permisos

```
chmod +x miprimerscript.sh
```

Y para ejecutarlo

```
./miprimerscript.sh
```

El resultado seria, mostrar por pantalla: Hola mundo.

Vamos a crea un escript que se mueva con el comando cd a un directorio y después mueva un fichero de ese directorio a otro.

```
#!/bin/bash
```

```
echo "Este es mi segundo scripts"
```

```
cd /home/rubenb/PracticaLinux/PrimeraParte/Sección1
```

```
mv fichero1.txt /home/rubenb/PracticaLinux/PrimeraParte/Sección2
```