

# Gestión de Software



GNU/LINUX



# Ubuntu GNU/Linux

Maikel Nieves Matamoros  
Joven Club de Computación y Electrónica  
2008

## Introducción



La gestión de software en cualquier Sistema Operativo es una tarea administrativa de vital importancia, permite actualizar el sistema en cuanto a lo que necesitamos tener instalado en la PC, permite al usuario tener sus aplicaciones preferidas listas para usar en el momento que las necesite, así como desinstalarla cuando no les sean de utilidad, logrando ahorrar espacio en disco duro. En Ubuntu GNU/Linux este proceso puede realizarse de manera sencilla, así como con rasgos avanzados, pero siempre enfocado a la simpleza del proceso y la comodidad del usuario. A continuación vamos a centrarnos en la **gestión de software** y en las herramientas disponibles para la **instalación/desinstalación de programas**.

## Índice

No.	Temática	Página
1-	<a href="#"><u>Paquetes y programas</u></a>	.....4
2-	<a href="#"><u>Las dependencias</u></a>	.....7
3-	<a href="#"><u>El sistema de repositorios</u></a>	.....8
4-	<a href="#"><u>Estructura de los repositorios de Ubuntu</u></a>	.....10
5-	<a href="#"><u>Añadir y quitar programas gráficamente</u></a>	.....11
6-	<a href="#"><u>Añadir y quitar programas en línea de comandos</u></a>	.....13
7-	<a href="#"><u>Búsqueda y descarga de aplicaciones para Ubuntu</u></a>	.....17

## 1- Paquetes y programas

Los **paquetes** son los distintos componentes de software, y los **programas** son, normalmente, un conjunto de paquetes. Habitualmente los proyectos software suelen tener algún sitio en **internet** donde alojar los diferentes ficheros para que cualquier usuario pueda obtenerlos fácilmente. Estos ficheros pueden ser, por ejemplo, el código fuente del programa **.tar.gz**, o un paquete con extensión **.deb** para Ubuntu.



Si se descarga el **código fuente** de un programa, para instalarlo en el sistema **hay que compilarlo** primero. Compilar, visto de forma muy sencilla, es el proceso de convertir un fichero de texto que contiene órdenes en algún lenguaje de programación en un **archivo ejecutable** por la máquina. Estos ficheros suelen venir organizados y empaquetados en un archivo comprimido con extensión **.tar.gz**, aunque no tiene por qué ser necesariamente ese tipo de compresión.

Los paquetes de código fuente generados con las herramientas estándar se construyen e instalan siempre de la misma forma.

Primero se configuran los detalles de compilación para la PC en cuestión, después se compila para generar el ejecutable y por último se instala en el sistema. Estas acciones se realizan en orden, teniendo instalado el compilador apropiado para ese programa, con los tres comandos:

- `./configure`
- `make`
- `sudo make install`

Normalmente todos los paquetes tienen un fichero de texto llamado README o INSTALL en el que se detallan las **instrucciones para instalar el programa** y las dependencias que hay que cumplir. También se puede ejecutar el comando `./configure --help` para obtener más información sobre la configuración de la compilación.



Otra forma en la que podemos encontrar un paquete es **con extensión .deb**, lo cual significa que es un paquete específicamente preparado para ser instalado en una distribución basada en **Debian**. Con este tipo de paquetes y mediante los gestores del propio sistema,

se hace mucho más rápida y fácil la instalación y desinstalación de software.

Del mismo modo, otras distribuciones de GNU/Linux también tienen paquetes con una estructura específica preparados para ser instalados fácilmente, como los **.rpm** de **Fedora, Red Hat**, etc. Son lo mismo que los **DEB** para Debian/Ubuntu y también tienen sus instaladores automáticos. Incluso existen algunas herramientas como **Alien** para convertir paquetes de un tipo a otro, por ejemplo para pasar un paquete RPM a un DEB.

## 2- Las dependencias

Las dependencias de un programa son aquellos paquetes de librerías o de otros programas que **se necesita tener previamente instalados** en el sistema para su correcto funcionamiento.

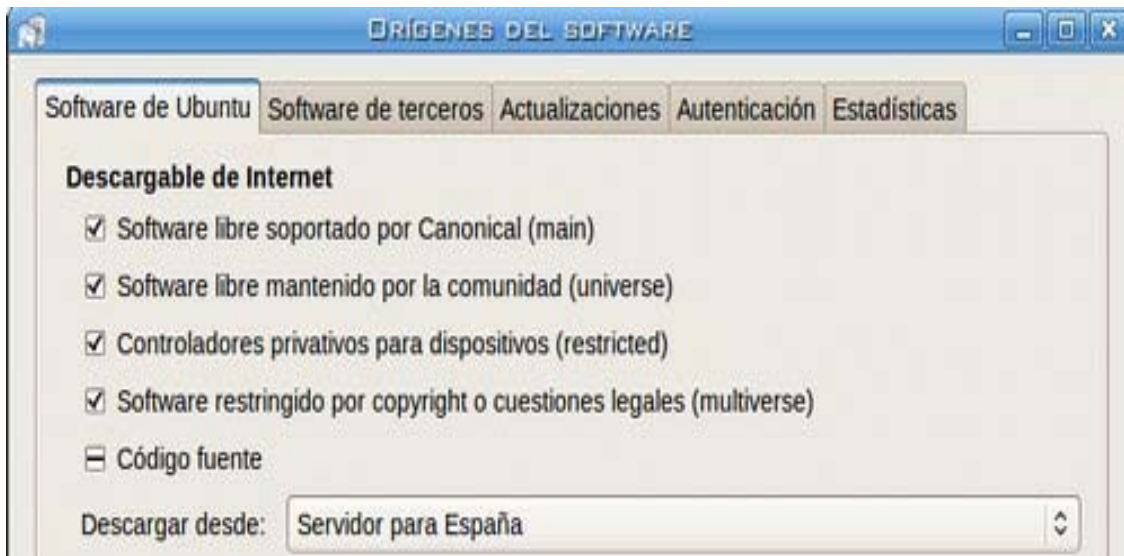
Una parte importante de la utilidad de las herramientas actuales de gestión de paquetes en sistemas GNU/Linux es su capacidad para resolver las dependencias. Lógicamente, en este sentido puede suponer un problema para el usuario el querer instalar un programa a partir del código fuente, compilándolo uno mismo. En caso de no tener instaladas previamente todas las dependencias necesarias, al intentar construirlo dará fallos por esa razón y el usuario deberá instalar primero cada una de ellas por separado para proceder con el programa principal.

### 3- El sistema de repositorios

Principalmente la instalación de programas en Ubuntu, al igual que muchas distribuciones de GNU/Linux, se basa en un **sistema de repositorios**. Los paquetes correspondientes a los diferentes programas y librerías se encuentran en un servidor. A su vez, los usuarios tienen alguna aplicación en el sistema que les permite conectarse a ese servidor para elegir el programa que quieren **descargar e instalar**, así como realizar **actualizaciones a nuevas versiones**.

El fichero de configuración `/etc/apt/sources.list` contiene una lista con todos los repositorios de software a los que se accede. **Se pueden añadir y quitar** editando directamente el fichero o usando alguna herramienta de configuración como la de *Sistema -> Administración -> Orígenes del software*, en la que vienen preconfiguradas algunas fuentes para activar y desactivar fácilmente, además de poder añadir más líneas de repositorios a nuestro gusto.





La mayoría de los usuarios prefieren activar todas las opciones de **Orígenes del software** para poder acceder a más aplicaciones y a sus últimas versiones.

## 4- Estructura de los repositorios de Ubuntu

Se divide en 4 componentes:

- **Main:** software libre, con actualizaciones de seguridad y soporte técnico oficial de Ubuntu.
- **Universe:** sin soporte oficial pero con actualizaciones cuando son ofrecidas a la comunidad, aplicaciones de todo tipo y con más variedad de licencias. Muchos programas de la comunidad libre preparados para el sistema concreto. Algunos proyectos son movidos a *Main* si consiguen el soporte de desarrolladores que estén dispuestos a garantizar los requisitos de este repositorio.
- **Multiverse:** licencias que no tienen por qué ser libres, es responsabilidad del usuario el cumplirlas y no es posible que estén soportadas o corregidas oficialmente.
- **Restricted:** software importante y muy usado, que no es libre pero es “soportado” por Ubuntu, en la medida de lo posible y de lo que permita la licencia. Algunos *drivers* de tarjetas gráficas están en este componente.

También hay muchos proyectos y páginas Web que mantienen sus propios repositorios, que pueden añadirse fácilmente como “**repositorios de terceros**” en los Orígenes de software del sistema.

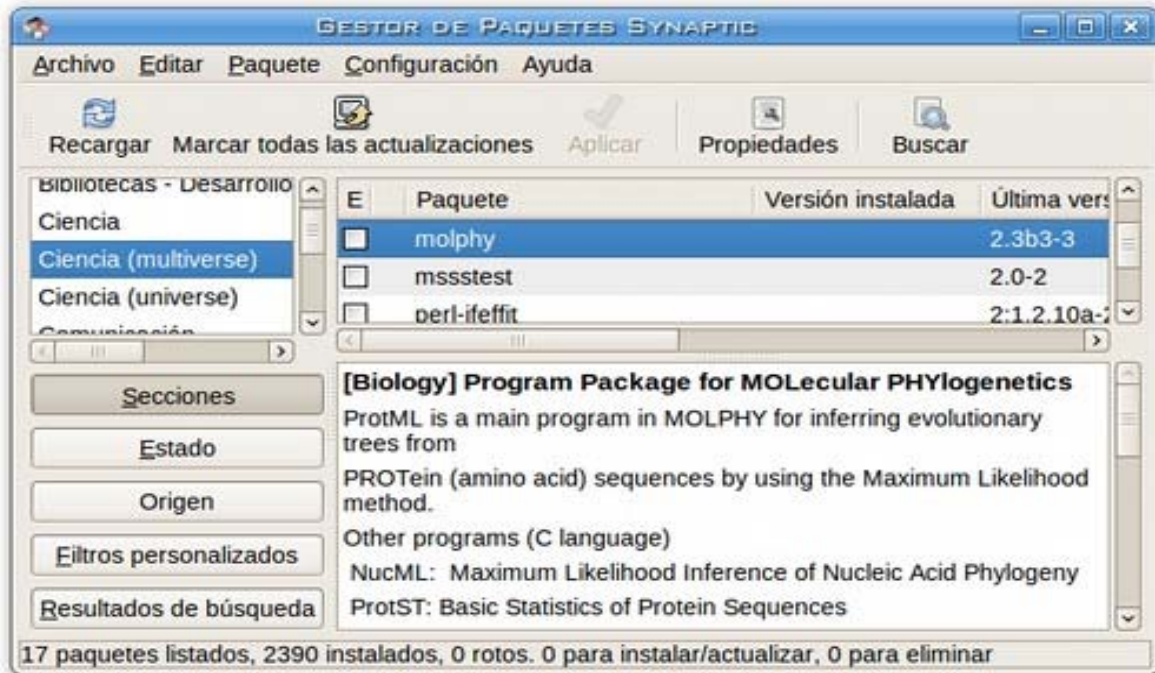
## 5- Añadir y quitar programas gráficamente

El **gestor de software** más sencillo e intuitivo de Ubuntu está en la opción del menú principal **Añadir/Eliminar...** Muestra una ventana con las diferentes categorías de software y permite buscar al vuelo entre los programas del tipo que tú escojas para que muestre: todas las aplicaciones, sólo las libres o sólo las soportadas oficialmente... Si haces **doble click** sobre un paquete DEB desde el navegador de archivos se abrirá una ventana de **Gdebi** para resolver las dependencias, guiar al usuario e instalar el paquete automáticamente a golpe de ratón.



**Synaptic** y **Adept** son dos opciones **más avanzadas para instalar y desinstalar programas gráficamente**, ya que no sólo muestran los programas completos sino todos los paquetes disponibles en los repositorios. Siguen teniendo una interfaz visual bastante agradable y son muy intuitivos. El primero de ellos utiliza las librerías gráficas GTK

y el segundo usa QT, diseñados para funcionar en los entornos **Gnome** y **KDE**, respectivamente.



Los dos programas que vienen de serie en Ubuntu. En cualquier caso, se pueden instalar ejecutando:

- `sudo apt-get install synaptic`
- `sudo apt-get install adept`

## 6- Añadir y quitar programas en línea de comandos

**APT** son las siglas de *Advanced Packaging Tool*. Es una librería de funciones usadas de forma poderosísima: el usuario elige un programa y la herramienta se conecta a los repositorios, descarga todos los paquetes necesarios (incluyendo todas sus dependencias) y los instala en el orden correcto. Incluso se puede actualizar de una versión a otra del sistema operativo con un solo comando. Hay varias herramientas que usan dichas librerías, las más conocidas son probablemente apt-get y apt-cache.

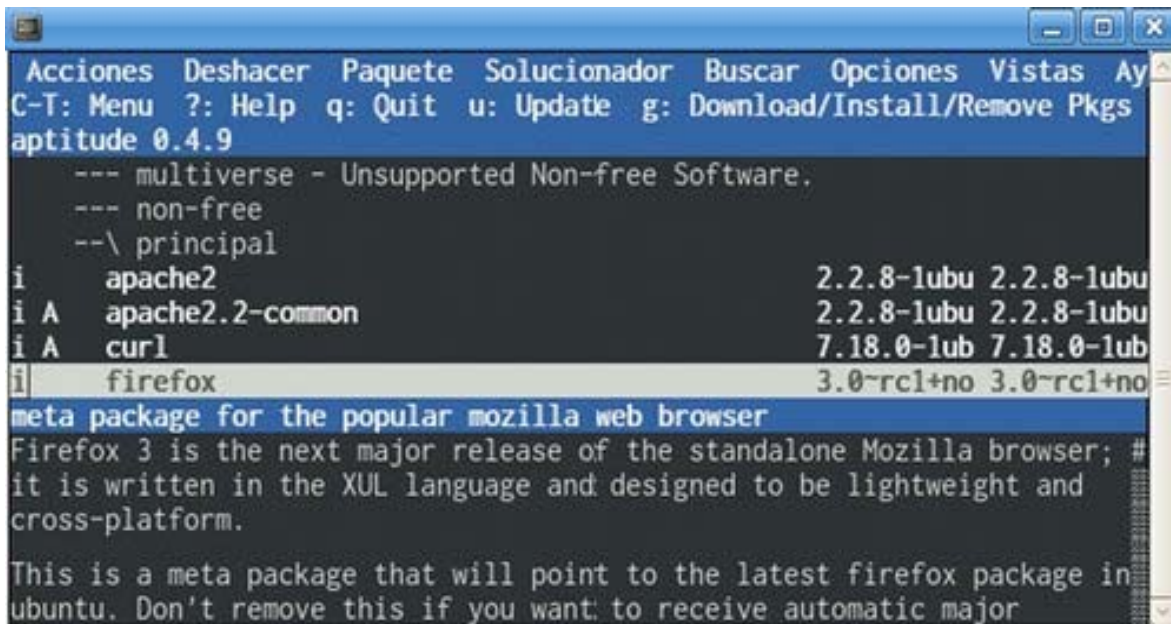
```
guille@:~$ apt-cache search music applet
music-applet - GNOME panel applet to control several music players
topshelf - current files applet for GNOME
guille@:~$ sudo apt-get install music-applet
sudo: unable to resolve host
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  music-applet
0 actualizados, 1 se instalarán, 0 para eliminar y 25 no actualizados.
Necesito descargar 80,1kB de archivos.
```

Algunos ejemplos de uso básico:

- apt-cache search music applet
- apt-cache show nombre\_paquete
- sudo apt-get install nombre\_paquete
- sudo apt-get remove --purge nombre\_paquete
- sudo apt-get update
- sudo apt-get upgrade

El primer comando busca en los repositorios y muestra una lista con los nombres exactos de los paquetes relacionados con las palabras “*music*” y “*applet*”. El segundo muestra toda la información de un paquete, el tercero lo instala, el siguiente lo elimina y los dos últimos son, respectivamente, para actualizar la lista de paquetes de los repositorios y para actualizar todos aquellos que tengan una versión posterior a la que hay instalada en el sistema.

**Aptitude** es una interfaz gráfica para APT, sigue ejecutándose en el terminal pero consigue mucha más interactividad y ser una herramienta mucho más flexible. Tiene muchas diferencias con APT, aunque las más visibles de cara al usuario son la interfaz mediante *ncurses*, la mejora de búsqueda y muestra de los resultados y la desinstalación automática de programas.



```
Acciones  Deshacer  Paquete  Solucionador  Buscar  Opciones  Vistas  Ay
C-T: Menu  ?: Help  q: Quit  u: Update  g: Download/Install/Remove Pkgs
aptitude 0.4.9
--- multiverse - Unsupported Non-free Software.
--- non-free
--\ principal
i   apache2                2.2.8-1ubu 2.2.8-1ubu
i A  apache2.2-common      2.2.8-1ubu 2.2.8-1ubu
i A  curl                   7.18.0-1ub 7.18.0-1ub
i |  firefox                3.0-rc1+no 3.0-rc1+no
meta package for the popular mozilla web browser
Firefox 3 is the next major release of the standalone Mozilla browser; #
it is written in the XUL language and designed to be lightweight and
cross-platform.
This is a meta package that will point to the latest firefox package in
ubuntu. Don't remove this if you want to receive automatic major
```

Al ejecutar el comando **aptitude** el programa muestra un **menú principal** por el que el usuario puede “navegar” casi del mismo modo que con los gestores de archivos con interfaz gráfica, pero desde dentro de la terminal.

Algunos ejemplos de uso básico de Aptitude, análogos a los de arriba de APT:

- aptitude search music applet
- aptitude search nombre\_paquete
- sudo aptitude install nombre\_paquete
- sudo aptitude remove nombre\_paquete
- sudo aptitude update
- sudo aptitude upgrade

**Aptitude** viene de serie en el sistema, pero si por lo que sea no lo tienes puedes instalarlo ejecutando el comando:

- sudo apt-get install aptitude.

Hay quienes prefieren tanto de APT como Aptitude, cada uno defiende y valora aspectos en los que uno es mejor que el otro, pero al final se trata de usar el que cada uno prefiera o considere más cómodo.

**Dpkg** es *la llave de la puerta* en cuanto a la gestión de paquetes de software en sistemas basados en Debian, como es el caso de Ubuntu. Instala, desinstala y muestra información sobre los paquetes que se

especifican, pero no es capaz de acceder a los repositorios ni de resolver dependencias. Es la **herramienta de bajo nivel** que usan los gestores de más alto nivel, como los descritos anteriormente.

```
guille@:~$ dpkg -l | grep firefox
ii firefox                 3.0~rc1+nobinonly-0ubuntu0.8.04.
ii firefox-3.0            3.0~rc1+nobinonly-0ubuntu0.8.04.
ii firefox-3.0-gnome-support 3.0~rc1+nobinonly-0ubuntu0.8.04.
ii firefox-gnome-support  3.0~rc1+nobinonly-0ubuntu0.8.04.
guille@:~$
```

Algunos ejemplos de uso de Dpkg para instalar, desinstalar y mostrar la lista de paquetes instalados en el sistema:

- `sudo dpkg -i paquete.deb`
- `sudo dpkg -r nombre_paquete`
- `sudo dpkg -l`



## **7- Búsqueda y descarga de aplicaciones para Ubuntu**

Hemos comentado cuáles son las herramientas principales de gestión de software en Ubuntu, con ellas se puede directamente buscar en los repositorios e instalar cualquier programa que encuentre.

También viene bien saber que Ubuntu mantiene una página Web de búsqueda de paquetes (<http://packages.ubuntu.com>) en la que se pueden realizar búsquedas avanzadas según nombre, descripción, versión de Ubuntu o repositorio.

Además, están empezando a funcionar los primeros portales de descarga de software con una sección especial dedicada a Ubuntu, como son UpToDown.com (<http://www.uptodown.com/ubuntu/>) y GetDeb.net (<http://getdeb.net/>). En ellos puedes encontrar programas empaquetados en formato DEB que quizá no estén en los sitios oficiales.