6. Gestión de paquetes y actualizaciones.

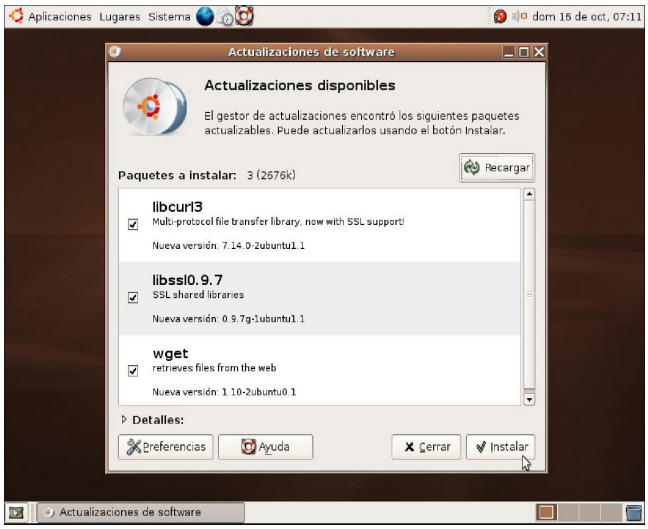
6.1. Gestión de actualizaciones de Ubuntu.

Ubuntu incorpora un gestor de actualizaciones que facilitan el mantenimiento de la distribución al usuario final.

Cada día el sistema comprueba si existen nuevas actualizaciones (sólo si hay conexión a Internet disponible). En caso de que haya nuevo software a actualizar, el sistema nos avisará con una señal en el área de notificación.

Para actualizar el sistema hacemos clic en el aviso del área de notificación o también podemos ir a "Sistema -> Administración -> Gestor de actualizaciones".

En caso de que no haya ninguna actualización disponible, simplemente obtendremos una ventana con un listado vacío. En caso contrario, se nos mostrarán todas las actualizaciones disponibles:



Siempre es recomendable tener el sistema actualizado para evitar problemas de seguridad. Para realizar la actualización, pulsaremos sobre el botón "Instalar".

A continuación se descargará de Internet el nuevo software para instalarse y configurarse de forma automática.

Finalmente se nos mostrará un aviso indicando que la actualización ha finalizado.

En el botón "Preferencias" podemos modificar los repositorios de Software desde donde Ubuntu adquiere las aplicaciones. Entraremos en este tema con mayor profundidad en los apartados siguientes.

6.2. El sistema de gestión de paquetes DEB.

Ubuntu esta basada en la distribución Debian, y por tanto hereda la forma en la que se gestionan las aplicaciones instaladas.

En términos generales, en GNU/Linux llamamos paquete a una aplicación, librería o componente que puede ser instalado en un sistema. A su vez, es frecuente ver que las diferentes distribuciones tiene su propio sistema de gestión de paquetes. Lo más conocidos son los **RPM** (originarios de RedHat) y los **DEB** (originarios de Debian). Ubuntu utiliza estos últimos.

Cuando se quiere instalar un paquete determinado, es posible que este dependa de terceros paquetes. Por ejemplo, si queremos instalar una aplicación gráfica para navegar por Internet que utiliza la librería gráfica GTK, es necesario que tengamos instalado en nuestro sistema la librería GTK para poder instalar el navegador.

Desde hace unos años, Debian implementó un sistema de gestión de paquetes DEB llamado APT. Este sistema resuelve automáticamente las **dependencias** que tiene una aplicación, de forma que si queremos instalar el navegador comentado en el ejemplo anterior, APT se encarga de bajar e instalar tanto el navegador como la librería GTK como todo aquello que sea necesario. Todo sin intervención del usuario y por tanto facilitando en gran medida la instalación de programas.

Ubuntu también utiliza el sistema APT, por tanto vamos a poder disfrutar de todas estas comodidades.

APT puede funcionar cogiendo los paquetes desde un CD, pero lo habitual es utilizar APT conjuntamente con Internet. En Internet podemos encontrar lugares con recopilaciones de paquetes para nuestra distribución, esos lugares son denominados **repositorios**.

Vamos a poder instalar/desinstalar el software contenido en esos repositorios con gran facilidad. Es posible instalar software no disponible en los repositorios, pero suele ser más complicado.

Ubuntu dispone de diferentes repositorios con paquetes:

- Repositorio oficial de paquetes de la distribución inicial para cada versión.
- Actualizaciones de seguridad (Security): nuevas versiones de paquetes que tenían fallos de seguridad y han sido corregidos.
- Actualizaciones (Updates): nuevas versiones de paquetes que tenían fallos importantes (no de seguridad) y han sido corregidos.

APT siempre selecciona por defecto la versión más reciente de los paquetes, por tanto siempre cogerá la versión actualizada.

Dentro de cada uno de estos repositorios, existen 4 componentes diferentes:

- "**main**": Aplicaciones libres con soporte por parte de la empresa Canonical. Son actualizadas rápidamente en caso de encontrarse fallos de seguridad. Se puede encontrar el software más utilizado en entornos de escritorio y servidores.

- "**restricted**": Aplicaciones con licencias propietarias soportadas por la empresa Canonical. Dispone de actualizaciones de seguridad sujetas a los creadores originales del software, ya que este no es libre.

- "**universe**": Más de 13.000 aplicaciones, no están soportadas directamente por la empresa Canonical pero si por la comunidad del Software Libre. Por defecto no esta activado, pero es

posible hacerlo fácilmente utilizando la herramienta gráfica de gestión de paquetes Synaptic.

- "**multiverse**" Aplicaciones que no queda claro si son legales (depende de la legislación de cada país) y no pueden ser distribuidas sin problemas. Por ejemplo, el reproductor de vídeo mplayer y sus codecs privativos (divx...). Por defecto no esta activado, pero es posible hacerlo fácilmente utilizando la herramienta gráfica de gestión de paquetes Synaptic.

Como hemos visto, por defecto sólo vienen activados los componentes "main" y "restricted". Es muy probable que nos resulten de gran utilidad activar "universe" y "multiverse", así dispondremos de una mayor cantidad de aplicaciones disponibles para instalar.

Para poder activar esos componentes podemos hacerlo a través de "Sistema -> Administración -> Gestor de actualizaciones Ubuntu", pulsando el botón "Preferencias". O bien a través de Synaptic ("Sistema -> Administración -> Gestor de paquetes Synaptic".

Iremos a "Configuración -> Repositorios". Aquí se nos mostrará un listado con los repositorios activos y sus componentes.

Por defecto se incorpora el CD de instalación como un repositorio más. Como lo habitual es trabajar con Internet, podemos eliminar esa entrada y así no se nos solicitará el CD cada vez que queramos instalar algo. Simplemente lo descargará de Internet.

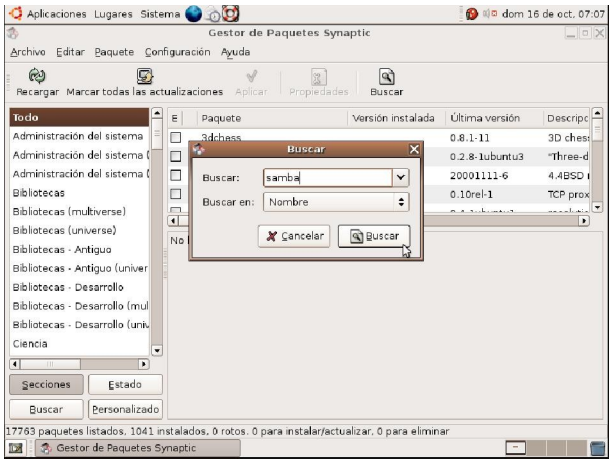
Nuestro sistema mantiene un listado de las aplicaciones que tenemos disponibles para instalar. Dado que hemos cambiado los repositorios, el listado también debe cambiar y por tanto debemos indicar que actualice dicho listado. Por tanto debemos pulsar "Si".

El listado de aplicaciones se irá actualizando automáticamente cada día para comprobar si existen nuevas versiones de aplicaciones.

6.3. El gestor de paquetes Synaptic.

Para gestionar los paquetes (instalarlos, eliminarlos,...) desde el entorno gráfico Ubuntu GNU/Linux cuenta con el gestor de paquetes Synaptic. Para ponerlo en marcha vamos a "*Sistema → Administración → Gestión de paquetes Synaptic*".

Veamos sus características.



En la parte inferior izquierda tenemos varios botones que modifican el contenido del lateral izquierdo:

- Secciones: Muestra diferentes secciones que catalogan el software disponible.
- Estado: Cataloga los paquetes por instalado, no instalado, nuevo, etc...
- Buscar: Muestra los resultados de anteriores búsquedas que hayamos realizado.
- Personalizado: Diferentes formas de catalogar los paquetes que puede ser personalizado.

Todos nos pueden servir para navegar por la inmensa cantidad de paquetes disponibles.

En caso de que queramos buscar un paquete determinado podemos pulsar, en la barra de herramientas, el botón "*Buscar*".

Es posible buscar por diferentes criterios, aunque habitualmente se buscará por nombre.

En la parte central superior aparecerán todos los paquetes que su nombre contenga la palabra buscada. Si en la fila del paquete buscado podemos observar el logo de Ubuntu, eso significa que ese paquete tiene soporte directo por parte de la empresa Canonical (es del repositorio/componente "main").

Si hacemos clic con el botón izquierdo sobre el paquete para seleccionarlo, aparecerá su descripción en la parte central inferior.

Es posible obtener más información pulsando sobre el paquete en cuestión con el botón derecho y seleccionando "*Propiedades*".

Desde esa nueva ventana podremos información general del paquete:

- Paquetes de los que depende.
- Ficheros que lo componen (muy útil cuando instalamos un programa y no sabemos como se llama el ejecutable, basta ir ahí y buscar los archivos ubicados en los directorios "bin"), etc...

Si volvemos a hacer clic con el botón derecho sobre el paquete, vemos que es posible llevar a cabo varias acciones:

- Marcar para instalación.
- Marcar para reinstalación.
- Marcar para eliminación: Desinstala programa, pero no borra los archivos de configuración.
- Marcar para eliminación completa: Desinstala el programa borrando también los archivos de configuración.

Si deseásemos instalar dicho paquete, simplemente tendríamos que seleccionar la opción deseada y pulsar sobre el botón "*Aplicar*" de la barra de herramientas. Esto abre una ventana que nos recuerda que es lo que va a ser instalado/desinstalado. Pulsamos "*Aplicar*" para llevar a cabo las acciones.

Desde Synaptic también es posible realizar actualizaciones del sistema. Habitualmente se harán utilizando el gestor de actualizaciones Ubuntu, pero si deseásemos hacerlo usando Synaptic también es posible.

Primero habría que asegurarse que el listado de aplicaciones esta actualizado pulsando "*Recargar*". A continuación pulsaremos "*Marcar todas las actualizaciones*" para que compruebe todos los paquetes instalados en busca de actualizaciones, si encuentra aplicaciones más recientes las marca

para actualizar y en caso de que sea necesario la instalación de nuevos paquetes o la eliminación de paquetes instalados, también se marcan.

A continuación simplemente tendremos que pulsar "*Aplicar*".

Ahora ya sabemos instalar/desinstalar aplicaciones en Ubuntu. Como se ha podido comprobar, el sistema es extremadamente sencillo. Incluso más rápido que con otros sistemas operativos.

Un problema típico del nuevo usuario de sistemas GNU/Linux es que desconoce por completo las aplicaciones libres disponibles, que podrían ser de utilidad. Para encontrar estas aplicaciones se puede usar diferentes medios:

- Buscar entre las categorías de "Sistema -> Administración -> Añadir programas".
- Buscar en Synaptic por descripción o secciones.
- Buscar en la página web GnomeFiles http://www.gnomefiles.org/
- Buscar en la página web Freshmeat http://freshmeat.net
- Buscar en la página Alternativas libres http://alts.homelinux.net/

Hay ocasiones en las que el software que buscamos no se encuentra en el repositorio. Entonces se hace necesaria una instalación manual, cosa que por desgracia no suele ser sencilla y varia dependiendo de la aplicación a instalar. La mejor recomendación es leerse las instrucciones de instalación de cada programa. En este manual veremos algún ejemplo de ese tipo.

6.4. Gestión de paquetes desde la consola.

Las aplicaciones para manipular el sistema de paquetes de Debian GNU/Linux son, básicamente, de dos tipos: los programas apt (*Advanced Packaging Tool*) y los dpkg (*Debian package*). El conjunto de aplicaciones apt sirven para configurar de dónde conseguimos los paquetes, cuáles son los que queremos y resuelven dependencias y conflictos con otros. Los programas dpkg sirven para instalar los paquetes, configurarlos, saber cuáles tenemos instalados, etc. Hay otras aplicaciones, como dselect o aptitude, que sirven para manipular los programas apt y dpkg proporcionando, en un solo entorno, herramientas interactivas para la manipulación de los mismos.

Los programas que en última instancia se encargan de instalar las aplicaciones son los `dpkg`. Estos programas descomprimen el fichero "`.deb`" e instalan el programa. Las aplicaciones `apt` nos ayudan a localizar las últimas versiones de los programas que necesitamos, copian en el disco los ficheros de las fuentes de donde las hayan extraído (FTP, CD-ROM, etc.) y comprueban dependencias y conflictos de los nuevos paquetes para que se puedan instalar correctamente. Las principales aplicaciones `apt` son las siguientes:

- `apt-config`: sirve para configurar algunas de las opciones de `apt` (la arquitectura de nuestro sistema, directorio donde se guardan los archivos, etc.).
- `apt-setup`: aplicación para configurar las fuentes de los paquetes (de dónde los obtenemos).
- `apt-cache`: gestión de la caché de paquetes (directorio donde se guardan los archivos "`.deb`"

antes de ser instalados).

- `apt-cdrom`: aplicación para gestionar CD-ROM que contengan paquetes.
- `apt-get`: actualización, instalación o descarga de los paquetes.

Toda la configuración de `apt` está en el directorio `/etc/apt/`.
En el fichero `/etc/apt/sources.list` es donde se guarda la configuración de las fuentes de los paquetes. Con todas estas fuentes se genera un listado de paquetes disponibles, que podemos consultar e instalar siempre que nos interese. Generalmente, el formato de este archivo sigue la siguiente sintaxis:

```
deb http://site.http.org/debian distribución sección1 sección2 sección3

deb-src http://site.http.org/debian distribución sección1 sección2 sección3
```

El primer campo de cada línea indica el tipo de archivo al que nos referimos: binarios (deb) o código fuente (deb-src). Seguidamente encontramos la referencia de la fuente de los paquetes, que puede ser un CD-ROM, una dirección de Internet, etc. El campo de distribución indica a `apt` qué versión de Debian GNU/Linux estamos utilizando. Este campo es importante porque cada versión de la distribución tiene sus propios paquetes. En los últimos campos podemos especificar qué tipo de paquetes queremos utilizar.

Si cambiásemos este fichero de forma manual, podríamos utilizar el comando "`apt-get update`" para actualizar todos los paquetes disponibles en el sistema. Para insertar los paquetes de un CD-ROM en el listado de paquetes disponibles, podríamos utilizar "`apt-CD-ROM add`", con lo cual se exploraría el CD insertado y se actualizaría el listado de paquetes del sistema. Si algunas de las fuentes contuvieran paquetes iguales, al instalarlo la misma aplicación `apt` detectaría cual es el más reciente o el que su descarga implica menos tiempo y lo bajaría de la fuente correspondiente. Con el programa `netselect`, además, podríamos configurar más ampliamente todo este sistema de descarga.

Otra opción muy interesante que nos proporciona la mayoría de distribuciones es la de la actualización de paquetes en los que se ha descubierto algún tipo de vulnerabilidad o fallo en su funcionamiento. Con Debian, tan sólo tenemos que añadir la siguiente línea en el archivo `/etc/apt/sources.list`:

```
deb http://security.debian.org/ stable/updates main contrib non-free
```

A medida que se van detectando paquetes críticos, se van poniendo en esta fuente, de forma que con sólo ejecutar "apt-get update" se avisa de las nuevas actualizaciones que debemos realizar en el sistema y se reinstalan los paquetes necesarios.

Tabla resumen de comandos apt	
apt-get update	Actualización del listado de aplicaciones disponibles
apt-get upgrade	Actualización de los paquetes del sistema sin dependencias
apt-get dist-upgrade	Actualización de los paquetes del sistema y las dependencias
apt-get install [aplicación]	Instalación de una aplicación
apt-get remove [aplicación]	Eliminación de una aplicación
apt-get remove --purge [aplicación]	Eliminación de una aplicación y sus archivos de configuración
apt-cache search [aplicación]	Buscar una aplicación

Por otro lado existe la herramienta "**aptitude**" que cumple la misma funcionalidad que apt-get pero con dos características más:

- Si se ejecuta sin parámetros muestra una interfaz textual con menús.
- Cuando instalamos una aplicación que tiene dependencias, estas las marca como "Automáticas" de forma que si borramos la aplicación que habíamos instalado, también borra las dependencias "Automáticas" excepto las que sean usadas por otros paquetes. Esta estupenda funcionalidad nos ayuda a mantener el sistema limpio y libre de paquetes no útiles.

Por desgracia ni apt-get ni synaptic funcionan igual, y ese es uno de los motivos por los que hay usuarios que prefieren usar únicamente "aptitude".

6.5. Gestión de paquetes que no se encuentran en los repositorios.

Existe la posibilidad de obtener paquetes en formato DEB fuera de los repositorios o desde repositorios que no tengamos listados en nuestro sources-list. Para gestionar estos paquetes tenemos el comando **dpkg**.

El comando **dpkg** es una herramienta que permite instalar, compilar, eliminar o gestionar los paquetes Debian, manteniendo la base de datos de paquetes.

Para listar todos los paquetes disponibles le podemos pasar el parámetro "`-l`", con lo cual se mostrará una lista completa de los paquetes y su estado de instalación (instalados, instalados pero no configurados, etc.).

Si quisiéramos ver toda la información de un determinado paquete, podríamos utilizar el parámetro "`-p`" seguido del nombre del paquete, con lo cual se muestran todas las dependencias, conflictos con otros paquetes, versión, descripción, etc.

Para instalar nuevos paquetes podemos utilizar el parámetro "`-i`" seguido del nombre del archivo.

Si nos da problemas de dependencias, podemos ignorarlas con "`--ignoredepends=X`", donde la "`X`" indica la dependencia, aunque debemos vigilar mucho cómo utilizamos este parámetro porque al ignorar dependencias es posible que el programa instalado no funcione correctamente.

Si sólo quisiéramos descomprimir el archivo "`.deb`"para ver qué contiene, también podríamos utilizar "`-x`".

Para eliminar los paquetes, debemos pasar "`-r`" seguido del nombre del paquete, que lo elimina del sistema, pero guardando sus archivos de configuración (con "`-P`" se elimina todo).

Otro parámetro muy interesante es el de "`--force-things X`" (donde la "`X`" es una de las siguientes opciones), que nos puede ayudar en alguno de los casos que mostramos a continuación:

- "`auto-select`": selecciona automáticamente los paquetes que se deben instalar o desinstalar con el nuevo paquete que elegimos.
- "`downgrade`": instala el paquete aunque haya versiones más nuevas del mismo.
- "`remove-essential`": aunque el paquete esté considerado como esencial en el sistema, lo elimina.
- "`depends`": no tiene en cuenta las dependencias, las considera como alertas.
- "`depends-version`": no tiene en cuenta dependencias de versión de los paquetes.
- "`conflicts`": instala el paquete, aunque entre en conflicto con algún otro del sistema.

En el sistema también podemos encontrar otra serie de aplicaciones relacionadas con el comando **dkpg** que pueden resultarnos muy útiles:

- `dpkg-divert`: nos sirve para manipular el lugar de instalación de algunos de los paquetes instalados en el sistema. Muy útil para evitar algunos problemas de dependencias.
- `dpkg-reconfigure`: con un mismo paquete deb muchas veces se incluye algún mecanismo para configurar algunas de las opciones de la aplicación de forma interactiva. Con esta aplicación podemos volver a configurar el paquete que le indiquemos con los mismos mecanismos utilizados en su instalación.
- `dpkg-scanpackages`: este programa sirve para escanear un determinado directorio del sistema que contenga archivos "`.deb`" para que se genere un archivo de índice. Con este archivo de índice podemos incluir el directorio como una fuente más de `apt`. Muy útil cuando bajamos programas no oficiales de la distribución.

- `dpkg-scansource`: aplicación con las mismas funcionalidades que la anterior pero para paquetes de código fuente.

- `dpkg-split`: programa para dividir y unir un paquete en varios archivos diferentes.

Nautilus desde Gnome también tiene la capacidad de manejar desde el entorno gráfico paquetes DEB.

6.6. Compilación e instalación de programas desde su código fuente.

En la administración de cualquier equipo es muy probable que en algunos casos nos encontremos que debemos utilizar algún programa que nuestra distribución no tiene o que necesitemos la última versión de un servidor de aplicaciones que todavía no está convenientemente empaquetado, etc. En estos casos, siempre podemos descargarnos el código fuente del programa y compilarlo manualmente. Es importante comprender la diferencia que existe entre compilar un programa para conseguir su ejecutable que descargar directamente el binario. Cuando compilamos un programa, éste utiliza las librerías disponibles en el sistema, mientras que si lo descargamos directamente, lo más probable es que no funcione adecuadamente porque intentará utilizar alguna librería que no será exactamente igual a la que tengamos instalada en el sistema. Por ello, lo más recomendable, cuando necesitemos instalar un nuevo programa del que no disponemos del paquete correspondiente, es compilarlo de nuevo.

Al bajar las fuentes de una aplicación, nos encontraremos con un fichero empaquetado y comprimido con `tar` y `gzip` o similares. Es usual añadir un fichero llamado `README` en el cual se explica paso a paso todas las acciones necesarias para compilar correctamente el programa. Aunque es recomendable leerlo, en la mayoría de casos, el proceso de instalación siempre es el mismo.

Lo primero que debemos hacer para compilar el nuevo programa es descomprimirlo y desempaquetarlo. Una vez hecho esto, dispondremos del código fuente estructurado en varios directorios. En su raíz, podemos encontrar (o no) un fichero llamado `Makefile`. Este archivo indica al compilador qué librerías se utilizan, cómo se deben compilar los archivos de código, etc. Si tenemos este `Makefi-le,` ya podemos compilar el programa ejecutando `make`. No hace falta que le pasemos ningún parámetro porque por defecto ya busca el fichero de `Makefile` y ejecuta las acciones que se especifican en él. Si el proceso no ha dado ningún error, ya podremos mover el ejecutable generado para ponerlo en alguno de los directorios del `PATH` configurado; de este modo, siempre que lo queramos ejecutar no tendremos que escribir su ruta completa. Muchos `Makefile` proporcionan, asimismo, instrucciones para que

podamos ahorrarnos este último paso. Generalmente, ejecutando "`make install`" el mismo programa se encarga de situar adecuadamente los binarios y, si existieran, los archivos de documentación. Finalmente, si no nos interesara guardar el código fuente del programa, ya podemos eliminar todo el contenido de los directorios creados.

Si el programa no incorpora el archivo de `Makefile`, generalmente se suele incluir algún *shell script* para generar automáticamente este fichero (habitualmente, *este script* se suele nombrar `configure`). Al ejecutar este *shell script* se comprobará que el sistema tenga instaladas todas las librerías necesarias para una buena compilación y, si faltara alguna, se daría un mensaje de aviso. Una vez ejecutado correctamente este *shell script,* ya dispondremos del `Makefile`, con lo que el proceso vuelve a ser el mismo que anteriormente

A continuación, a modo de resumen presentamos el diagrama del proceso de compilación e instalación de una aplicación partiendo de su código fuente: