

**UNIDAD DIDACTICA 3**

**GESTIÓN DEL SOFTWARE**

# 1. INTRODUCCIÓN

---

- GNU/Linux es código libre y ello permite que podamos descargar el software de Internet libremente.
- La mayoría de este software se encuentra en múltiples repositorios repartidos por todo Internet.
- Tipos de software existentes:
  - Universe , software mantenido por la comunidad.
  - Multiverse , software comercial o privativo de terceras empresas o fabricantes.
- La gestión del software conlleva:
  - Instalación/desinstalación. Creación de dependencias
  - Certificar la integridad (md5, cambios, etc)
  - Actualizaciones de seguridad

# 1. MÉTODOS DE GESTIÓN DEL SOFTWARE

---

## Métodos tradicionales/clásicos

- Paquetes .rpm (standard de RedHat).

Ejemplo: `rpm -ivh gcc-3.0-i386.rpm`

- Paquetes .deb de Debian.

Ejemplo: `dpkg -i gcc-3.0-i386.deb`

- Ficheros .tar.gz cuyo contenido normalmente es el código fuente, ficheros de ayuda y algún script que facilita la compilación e instalación.

Ejemplo: `./configure | make install`

## Métodos más modernos

Aplicaciones que facilitan la gestión, algunas visuales:  
apt, aptitude, synaptic, yum, etc.

## 2. DPKG DEBIAN PACKAGE MANAGER

- Instalar un paquete:

`dpkg -i <.deb filename>`

`dpkg -i avg71flm_r28-1_i386.deb`

- Instalar todos los paquetes de forma recursiva desde un directorio

`dpkg -R`

`dpkg -R /usr/local/src`

- Desempaquetar el paquete, pero no configurarlo `dpkg --unpack package_file`

`dpkg --unpack avg71flm_r28-1_i386.deb`

- Listar contenidos de un paquete deb: `dpkg -c filename`

## 2. DPKG DEBIAN PACKAGE MANAGER

- Reconfigurar un paquete desempaquetado

`dpkg --configure package`

`dpkg --configure avg71flm_r28-1_i386.deb`

- Borrar un paquete instalado excepto los ficheros de configuración

`dpkg -r`

`dpkg -r avg71flm_r28-1_i386.deb`

- Borrar un paquete instalado incluyendo los ficheros de configuración

`dpkg -P`

`dpkg -P avg71flm`

## 2. DPKG DEBIAN PACKAGE MANAGER

- Muestra información sobre un paquete:

`dpkg -I filename [control-file]`

- Lista todos los paquetes instalados:

`dpkg -l`

- Reporta el estado de un paquete específico

`dpkg -s package-name` ➡ `dpkg -s ssh`

- Listar los ficheros instalados en tu sistema desde un paquete `dpkg -L package-Name` ➡ `dpkg -L apache2`

- Busca por el nombre en los paquetes instalados

`dpkg -S filename-search-pattern`

`dpkg -S /sbin/ifconfig`

### 3. PAQUETES RPM

---

Opciones principales:

- i (install) instalar un paquete
- e (erase) desinstalar un paquete
- u (upgrade) actualizar un paquete
- q (query) obtener información
- V (verify) verificar
- v (verbose) proporciona información del proceso
- h (help) genera una barra de estado durante el proceso.

### 3. PAQUETES RPM

---

- INSTALAR UN PAQUETE ➦ `rpm -ivh paquete.rpm`

Opciones adicionales:

- test: simulación de la instalación de los paquetes.
- replacepks: reemplaza paquetes si ya están instalados.
- replacefiles: reemplaza archivos si ya están instalados.
- nodeps: no chequea las dependencias con otros paquetes.

Ejemplo ➦ `rpm -ivh --test foo_1.0_1_i386.rpm`

Simulación de la instalación del paquete, no instalará nada, pero informará de todas las incidencias de la instalación)

- BORRAR UN PAQUETE ➦ `rpm -ev paquete.rpm`

Opciones adicionales:

- test: simulación la desinstalación de los paquetes.
- nodeps: no chequea las dependencias con otros paquetes.



### 3. PAQUETES RPM

---

- ACTUALIZAR UN PAQUETE ➦ `rpm -uvh paquete.rpm`

Opciones adicionales:

---test: simulación de la actualización de los paquetes.

---replacepks: reemplaza paquetes si ya están instalados.

---replacefiles: reemplaza archivos si ya están instalados.

---nodeps: no chequea las dependencias con otros paquetes.

Ejemplo ➦ `rpm -uvh -test foo_1.0_1_i386.rpm`

- VERIFICAR UN PAQUETE ➦ `rpm -V paquete.rpm`

Verifica que todo el paquete "nombredelpaquete" este correctamente instalado.

`rpm -Vf archivo` ➦ verifica que el archivo "archivo" este correcto.

### 3. PAQUETES RPM

---

- CONSULTAS

`rpm -q nombredelpaquete` ➡ información del paquete.

`rpm -qa` ➡ lista todos los paquetes instalados actualmente

`rpm -qf archivo` ➡ indica el nombre del paquete al que pertenece el archivo.

`rpm -qp paquete.rpm` ➡ proporciona el nombre del paquete contenido en el archivo paquete.rpm

`rpm -qc nombredelpaquete` ➡ muestra todos los archivos de configuración del paquete especificado

`rpm -qcf archivo` ➡ al añadir la opción f muestra todos los archivos de configuración del paquete que instalo el archivo "archivo".

`rpm -qcp paquete.rpm` ➡ al añadir la opción p muestra todos los archivos de configuración del paquete contenido en paquete.rpm

### 3. PAQUETES RPM

---

`rpm -ql nombrepaquete / rpm -qlf archivo / rpm -qlp paquete.rpm`

➡ muestra todos los archivos del paquete

`rpm -qd nombrepaquete / rpm -qdf archivo /rpm -qdp paquete.rpm`

➡ muestra todos los archivos de documentación del paquete.

`rpm -qi nombrepaquete /rpm -qif archivo /rpm -qip paquete.rpm`

➡ información sobre el paquete.

`rpm -qR nombrepaquete / rpm -qRf archivo / rpm -qRp paquete.rpm`

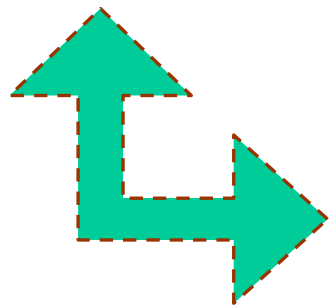
➡ lista las dependencias del paquete (otros paquetes que son necesarios para instalar el programa contenido dentro de nuestro paquete)

## 4. ARCHIVOS COMPRIMIDOS EN GNU/LINUX

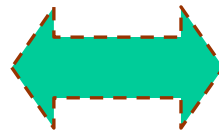
---

Formato archivo comprimido en LINUX: "nombre.tar.gz".  
Para la compresión de archivos se utilizan 2 algoritmos:

- gzip/gunzip: Uno de los primeros formatos de compresión que apareció, continúa usándose actualmente.
- bzip2/bunzip2: Algoritmo de compresión más eficaz pero más lento que gzip. En los equipos actuales las diferencias son mínimas.



Sólo pueden  
comprimir un  
único archivo.



Para poder comprimir  
varios archivos es  
necesario juntarlos en  
uno de solo mediante el  
proceso tar.

## 4. ARCHIVOS COMPRIMIDOS EN GNU/LINUX

---

### Compresión de un conjunto de archivos

Juntar todos los archivos deseados en uno y comprimir este archivo con alguno de los dos métodos disponibles gzip o bzip2.

### Descomprimir un archivo tar.gz

Es necesario primero descomprimirlo, para obtener así el archivo tar. A continuación será necesario obtener todos los archivos incluidos en el archivo tar.

En la actualidad estos dos procesos pueden realizarse mediante una única orden y paso.

## 4. ORDEN TAR

---

La orden tar permite manejar los archivos tar.gz con una única orden. Sus opciones las siguientes:

Función básica (sólo es posible indicar una opción en cada ejecución)

- -x: Extrae los archivos de un tar.gz específico
- -c: Crea un archivo
- -r: Añade archivos dentro de un tar.gz específico
- -t: Lista los archivos de un tar.gz específico

Opciones suplementarias (es posible especificar más de una en cada ejecución)

- -f: Permite especificar el archivo donde tar.gz (para comprimir o descomprimir)
- -v: Verbose: Indica lo que hace el programa mientras se ejecuta.
- -z: Compresión con algoritmo gzip/gunzip2
- -j: Compresión con algoritmo bzip2/bunzip2



## 4. EJEMPLOS TAR

---

`tar -cvf ejemplo.tar /home/*` ➦ Junta todos los archivos del directorio /home SIN ningún tipo de compresión

`tar -xvf ejemplo.tar` ➦ Extrae todos los archivos

`tar -czvf ejemplo2.tar.gz /home/*` ➦ Junta todos los archivos del directorio /home CON compresión gzip

`tar -xzvf ejemplo2.tar.gz` ➦ Extrae todos los archivos del archivo ejemplo2.tar.gz anterior

`tar -cjvf ejemplo3.tar.gz /home/*` ➦ Juntar todos los archivos del directorio /home CON compresión bzip2

`tar -xjvf ejemplo3.tar.gz` ➦ Obtiene todos los archivos del archivo ejemplo3.tar.gz anterior

## 5. COMANDO APT (ADVANCED PACKAGING TOOL)

---

- ⌘ Es un sistema de gestión de paquetes creado por el proyecto Debian. APT simplifica en gran medida la instalación y eliminación de programas en los sistemas GNU/Linux.
- ⌘ Instala y elimina paquetes del sistema, y permite actualizar las listas de paquetes o el propio software.
- ⌘ No existe un programa apt en sí mismo, sino que APT es una librería de funciones C++ que se emplea por varios programas de línea de comandos para distribuir paquetes.
- ⌘ Está formado de 16 herramientas hermanas diferentes. En especial, apt-get y apt-cache.



## 5. COMANDO APT (ADVANCED PACKAGING TOOL)

---

**Sudo apt-get update** ➦ Descarga nuevas listas de paquetes

**Sudo apt-get upgrade** ➦ Actualiza el sistema sin borrar nada.

**Sudo apt-get install** ➦ Instala el paquete o los paquetes que se indica separados por espacios.

**Sudo apt-get remove** ➦ Desinstala el paquete o los paquetes junto con sus ficheros de configuración que se indica separados por espacios.

**Sudo apt-get autoremove** ➦ Elimina automáticamente todos los paquetes que ya no se usan

**Sudo apt-get purge** ➦ Elimina y purga los paquetes del sistema.

**Sudo apt-get dist-upgrade** ➦ Actualiza los paquetes de la distribución a la ultima versión .

**Sudo apt-cache search** ➦ Busca un paquete que contenga ese nombre o descripción.

## 5. COMANDO APT (ADVANCED PACKAGING TOOL)

---

**Sudo apt-get clean** ➦ Borra los paquetes descargados.

**Sudo apt-get autoclean** ➦ Borra (no desinstala) los paquetes descargados que sean redundantes (se quedará con la última versión).

**Sudo apt-get check** ➦ Verifica que no haya dependencias incumplidas.

**Sudo apt-get help** ➦ Para tener más opciones del comando apt-get

Opciones adicionales de apt (se añaden al final):

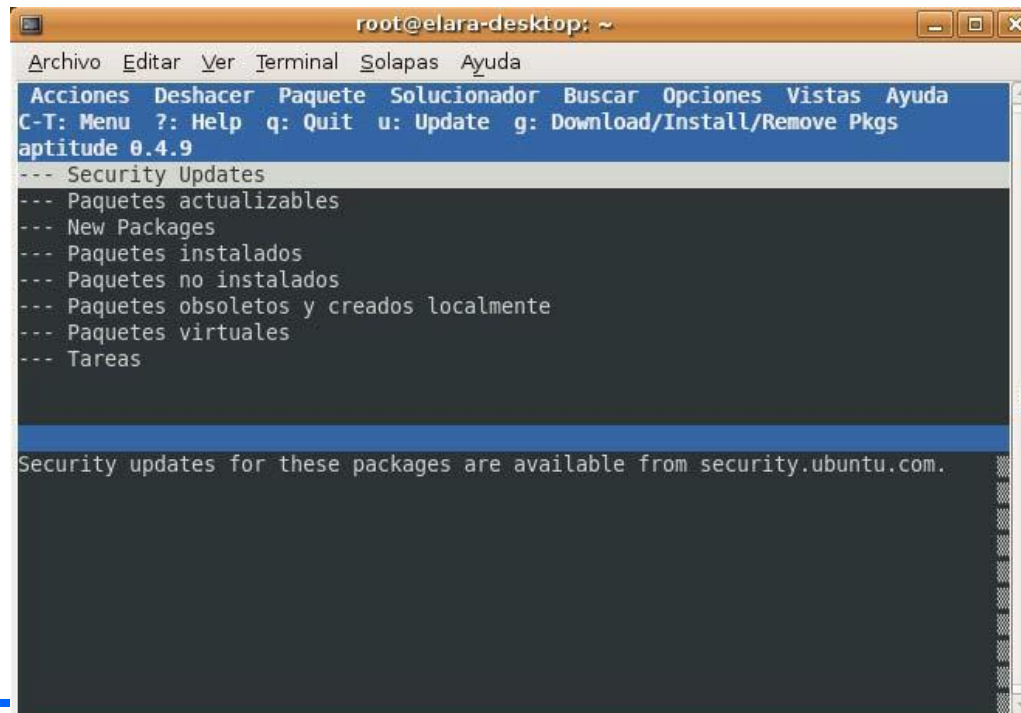
--download-only: solo descargara los paquetes, no los instalará.

install -f: tratará de arreglar los paquetes rotos.

--assume-yes: asume respuesta afirmativa a todas las preguntas simples.

## 6. COMANDO APTITUDE

- Herramienta mejor que apt para instalar, eliminar, actualizar, y administrar los paquetes.
- Resuelve el problema de las dependencias huérfanas.
- Usa una sola herramienta con muchas funciones.



Se puede utilizar  
en modo comando o  
en modo gráfico,  
ejecutando  
aptitude en un  
terminal

## 6. COMANDO APTITUDE

---

**aptitude** ➦ Muestra una interfaz para buscar, navegar, instalar, actualizar y gestionar paquetes.

**aptitude install** ➦ Instala el paquete o los paquetes que se indica separados por espacios, junto con las dependencias necesarias.

**aptitude remove** ➦ Elimina los paquetes que se indican separados por espacios junto con las dependencias que queden huérfanas.

**aptitude purge** ➦ Elimina los paquetes que se indican separados por espacios (y dependencias huérfanas junto con los ficheros de configuración,

**aptitude search** ➦ Busca paquetes en las listas de paquetes locales de apt un paquete que contenga ese nombre o descripción.

**aptitude update** ➦ Actualiza las listas de paquetes disponibles.

**aptitude upgrade** ➦ Actualiza el sistema sin borrar nada.

**aptitude clean** ➦ Elimina los ficheros que se descargaron para instalar software en tu sistema.

---

## 6. COMANDO APTITUDE

---

**aptitude dist-upgrade** ➦ Actualiza la distribución del sistema, borrando e instalando lo que sea necesario.

**aptitude show** ➦ Muestra la información disponible sobre un paquete.

**aptitude clean** ➦ Borra los paquetes descargados.

**aptitude autoclean** ➦ Borra (no desinstala) los paquetes descargados que sean redundantes (se quedará con la última versión).

**aptitude hold** ➦ Fuerza a que un paquete permanezca en su versión actual, y no se actualice.

Opciones adicionales de aptitude:

-s: simula las acciones pero no las realizará

-d: sólo descargara los paquetes, no los instalará.

-y: supone respuesta afirmativa a todas las preguntas simples.

-f: trata de arreglar paquetes rotos de forma agresiva.

## 6. APTITUDE VS APT-GET

---

- 'apt-get remove' elimina los paquetes indicados, pero deja sus dependencias, aunque éstas queden huérfanas (ningún paquete de tu sistema las necesita).
- aptitude recuerda las dependencias que se han aplicado en la instalación de un paquete, y borrará el programa junto con todas sus dependencias huérfanas.
- Si se instala con apt-get o con entorno gráfico Synaptic, la desinstalación borrará sólo el paquete especificado, pero no las dependencias.
- Apt-get autoremove ➡ Puede encargarse de las dependencias huérfanas pero sólo para versiones > 6.10)

## 7. INSTALACIÓN SOFTWARE FORMA GRÁFICA

---

Existen dos aplicaciones:

### 1) Centro de software en Ubuntu.

Una sencilla aplicación con la que se puede añadir o quitar paquetes, del S.O. de una manera muy sencilla.

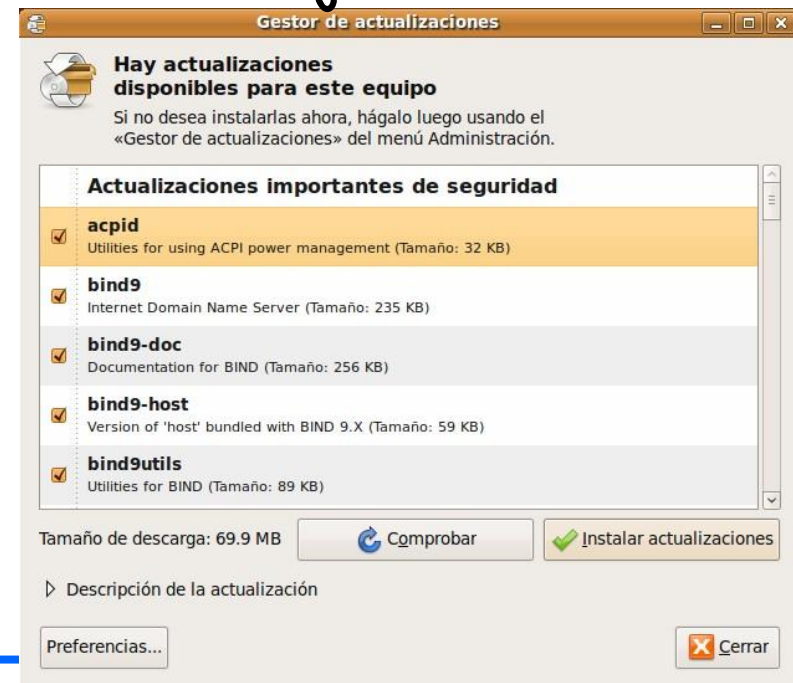
### 2) En Modo gráfico. Aplicación Synaptic Sistema/Administración/Gestor de paquetes Synaptic

Se tiene más control sobre los programas que se instalan en el sistema. Por el momento, Synaptic utiliza apt-get.

## 8. ACTUALIZACIÓN DEL SERVIDOR

- ☞ Igual que ocurre con Windows, cada cierto tiempo los desarrolladores de Ubuntu publican actualizaciones de seguridad y nuevas funcionalidades para el sistema Ubuntu
- ☞ Cuando las actualizaciones están probadas y pueden instalarse, el S.O. nos avisa con un mensaje

Pulsando en el icono desde donde parte el mensaje, se nos abrirá el *Gestor de actualizaciones*, que nos permitirá actualizar el software con las nuevas funcionalidades de forma sencilla.





## 8. ACTUALIZACIÓN DEL SERVIDOR

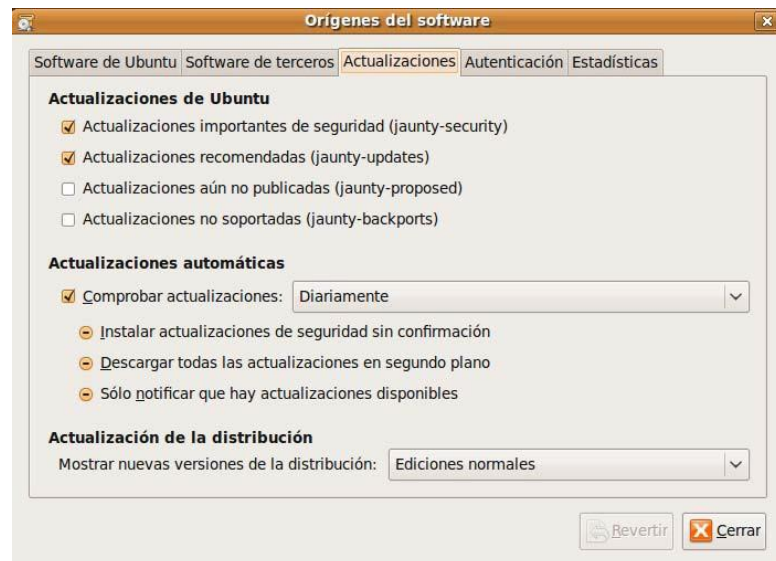
---

☞ Podemos ejecutar directamente el gestor:

**Gestor de actualizaciones**

☞ Para configurar la frecuencia con la que el Gestor de actualizaciones comprobará las nuevas versiones de software y parches de seguridad:

**Gestor de paquetes Synaptic/  
Configuración/Repositorios/Pestaña Actualizaciones**



Ambos gestores son procesos que se accede como root. Por eso se nos solicitará la contraseña del súper usuario para poder realizar esta operación

## **8. ACTUALIZACIÓN DEL SERVIDOR**

---