

Repositorios y paquetes

***Arquitectura y Sistemas Operativos.
Tecnatura Superior en Programación.
UTN-FRA***

Autores: *Prof. Martín Isusi Seff*

Revisores: *Prof. Marcos Pablo Russo*

Versión: 1



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).

¿Qué es un repositorio?

Un repositorio es una "bodega" donde se encuentran los paquetes a instalar en cualquier distribución. Por lo general, los repositorios son servidores ftp o http (aunque también pueden ser locales) en donde se encuentran todos los paquetes disponibles para una distribución. Se habla de "paquetes" en vez de "programas", porque un paquete no necesariamente contiene un programa. Un paquete puede contener imágenes, librerías, código fuente, documentación, traducciones y desde luego programas. Cada distribución tiene sus propios repositorios y su forma de clasificarlos, y básicamente funcionan muy parecido. Generalmente, existe un programa en consola o en modo gráfico que se encarga de descargar y de instalar los paquetes, todo de modo automático.

Tipos de paquetes

Existen, por lo general, tres tipos de paquetes: estable, testing, inestable.

- Paquetes estable o estables: Consta de software estable y bien probado, y cambia sólo al incorporar correcciones importantes de seguridad.
- Paquetes en prueba o testing: Esta área contiene paquetes que se espera que sean parte de la próxima distribución estable. No goza de actualizaciones del equipo de seguridad en el mismo momento en que salen.
- Inestables o unstable: Contiene los paquetes más recientes. Una vez que un paquete ha cumplido las exigencias de estabilidad y calidad de empaquetado, será incluido en testing. Unstable tampoco está soportada por el equipo de seguridad.

Gestor de paquetes APT

APT (Advanced Packaging Tool) es el gestor de paquetes que se utiliza en distribuciones derivadas de Debian GNU/Linux: Crunchbang, Trisquel Ubuntu, Lubuntu, Linux Mint, etc... Es un sistema muy estable y muy fácil de usar a la vez, aquí se mostrarán los comandos principales para tener el control de la biblioteca **apt** de forma básica.

1) Actualizar la base de datos de paquetes.

```
# apt-get update
```

o

```
# apt update
```

Estos comandos leen el contenido del directorio **/etc/apt/sources.list** y **/etc/apt/sources.list.d/***, y actualizan la base de datos que se encuentra en el directorio **/var/lib/apt/lists/**.

2) Validar el estado de los paquetes

```
# apt-get check
```

Este comando chequea que la información que se ingresó en **sources.list** es correcta.

3) Actualizar paquetes

apt-get upgrade

Este comando actualiza los paquetes, es decir, verifica la versión de los paquetes instalados, y si hay nuevas versiones o dependencias, las actualiza. Previamente se debe ejecutar el comando **apt-get update**.

4) Actualizar la distribución

apt-get dist-upgrade

Esto nos permite actualizar la distribución, es decir una nueva actualización contiene nuevos paquetes como también software propio de la distribución que estamos utilizando.

5) Instalación de un paquete

apt-get install mc

El comando **apt-get install** instalará el paquete especificado. En el ejemplo, el paquete **mc** (Norton Commander)

6) Reinstalar un paquete

apt-get --reinstall mc

Esto es, si ya está instalado el paquete, y por alguna razón tiene problemas, este comando lo reinstalará.

7) Resolver dependencias

apt-get -f install

Este comando resuelve las dependencias de un paquete que hayamos instalado mediante el comando **dpkg**, es decir bajamos un paquete como puede ser **virtualbox*.deb** y desde la maquina lo instalamos. Esto necesita dependencias, y la forma de resolverlo es mediante este comando.

8) Remover paquetes

apt-get remove mc

Esto lo que hace es solo remover el paquete, si queremos remover los archivos de configuración también, ejecutamos el siguiente comando:

apt-get --purge remove mc

9) Limpiar los paquetes bajados

apt-get clean

Este comando borra el contenido del directorio **/var/cache/apt/archives/** que contiene todos los paquetes que bajamos de los repositorios. Antes de borrarlo, podemos copiarlo para que se instale en otro equipo.

10) Limpiar paquetes no usados

apt-get autoclean

Hay paquetes que se vuelven viejos y no se usan debido, por ejemplo, actualizaciones. En estos casos, este comando ayuda a removerlos.

11) Eliminar librerías innecesarias

apt-get autoremove

Cuando instalamos un paquete posiblemente se instalen dependencias junto con él. Si después borramos ese programa, quedan dependencias inservibles. Estas dependencias pueden borrarse con el anterior comando.

12) Buscar paquetes

apt-cache search paquete

Este comando busca paquetes en la base de datos que tenemos en el directorio **/var/lib/apt/lists/**.

13) Obtener sobre un paquete instalado la versión y el candidato

apt-cache policy paquetes

Con este comando obtenemos información sobre el paquete que tenemos instalado y cuál es el candidato a ser instalado.

Ej:

apt-cache policy mc

mc:

Instalados: 3:4.8.18-1

Candidato: 3:4.8.18-1

Tabla de versión:

***** 3:4.8.18-1 500**

500 http://ar.archive.ubuntu.com/ubuntu zesty/universe amd64

Packages

100 /var/lib/dpkg/status

14) Obtener información de un paquete instalado

apt-cache showpkg paquetes

La información que se muestra, se extrae de la base de datos de paquetes que se encuentra en el directorio **/var/lib/apt/lists/**.

15) Instalar un paquete que tengamos en nuestra pc (un archivo .deb)

dpkg -i virtualbox-5.1_5.1.26-117224~Ubuntu~zesty_amd64.deb

Instala directamente el paquete que bajamos. A diferencia del comando **apt-get install paquete**, al usar el comando **dpkg -i**, no baja las dependencias necesarias. Por dicha razón, tenemos que usar el comando **apt-get -f install**.

16) Listar los paquetes instalados

dpkg -l | more

Con este comando podemos ver todos los paquetes que tenemos instalados en nuestro equipo tanto sea por el comando **apt-get install** como también **dpkg -i**.

17) Ver contenido de los paquetes

dpkg -L mc | more

Esto nos muestra todo el contenido del paquete que fue instalado, toda su estructura de archivos.

18) Borrar un paquete

dpkg -r mc

Borrar un paquete instalado mediante **dpkg -i**.

19) Buscar a qué paquete pertenece un comando.

which ls**# dpkg -S /bin/ls**

20) Configurar un paquete

dpkg --configure paquete

21) Reconfigurar todo

dpkg --configure -a

Esto nos permite reconfigurar los paquetes que quedaron sin configurar.